

Supervisionamento de Coordenadas Espaciais de um Quadricóptero Baseado em Redes de Petri Colorida

Raif C. Gomes¹, George André Pereira Thé¹

¹Departamento de Engenharia de Teleinformática– Universidade Federal do Ceará
(UFC)

Caixa Postal 6007 – 60455-970 – Fortaleza – CE – Brasil

raif.carneiro@gmail.com, george.the@ufc.br

Abstract. *This paper aims to develop a CPN (Coloured Petri Net) able to supervise a quadrotor, UAV (Unmanned Aerial Vehicles) rotary-wing, characterized by a structure typically symmetrical and cross, with the use of the tool CPN Tools, 4.0.0. Thus, allowing the supervision of a quadrotor graphically, intuitively and low cost.*

Resumo. *Este trabalho tem como objetivo, desenvolver uma RPC (Rede de Petri Colorida) capaz de supervisionar um quadricóptero, VANT (Veículos Aéreo Não Tripulado) de asas rotativas, caracterizado por uma estrutura, tipicamente, simétrica e cruzada, com a utilização da ferramenta gratuita CPN Tools, versão 4.0.0. Possibilitando assim, o supervisionamento de um quadricóptero de forma gráfica, intuitiva e de baixo custo.*

1. Introdução

Cada vez mais, grandes empresas multinacionais como Google, Amazon e até o Facebook investem no uso de VANT's (Veículos Aéreos Não Tripulados) ou drones tanto para solucionar problemas de logística, quanto para levar o acesso à internet a áreas remotas.

Em breve milhares de drones poderão ser vistos nos céus realizando tarefas cotidianas como guiar um turista pela cidade ou até acompanhar crianças até a escola. Apesar da tendência de um aumento exponencial no uso de VANT's nos próximos anos, eles ainda possuem um valor elevado e somente podem ser operados em aplicações profissionais por pessoas qualificadas e/ou através de softwares dispendiosos.

Com base nisso, é imperativo o desenvolvimento de um modelo supervisor que possibilite o barateamento do custo operacional envolvido no controle dos equipamentos supracitados, já que por meio desse, pessoas com conhecimentos básicos em drones por exemplo, poderiam manusear um quadricóptero. É neste contexto que se propõe o uso de Redes de Petri Coloridas. Conforme [Murata 1989], as Redes de Petri e suas extensões são classes de modelos conceituais, as quais podem ser usadas na modelagem dos mais diversos tipos de sistemas computacionais.

Assim, este trabalho tem como objetivo, desenvolver uma RPC (Rede de Petri Colorida) capaz de supervisionar um quadricóptero ou quadrotor, VANT com asas rotativas, caracterizado por possuir quatro motores, com a utilização da ferramenta gratuita CPN Tools, versão 4.0.0 de forma gráfica, intuitiva e de baixo custo.

2. Metodologia

Para desenvolver a Rede de Petri Colorida proposta, foi realizado um levantamento bibliográfico, o qual culminou na escolha dos softwares empregados nesta pesquisa. Essa seleção, deu-se a partir da necessidade de se utilizar programas de modelagem já consolidados no meio científico, graças a confiabilidade na geração de gráficos e dados por eles.

Desta forma, foi escolhido o CPN Tools, versão 4.0.0, o qual é ferramenta livre voltada para modelagem através das Redes de Petri e suas extensões. Assim o modelo referido foi empregado para supervisionar um quadricóptero em condições ideais, que foi implementado e simulado com o uso do software de modelagem matemática Matlab, versão 2014.a, conforme [Gomes and Aquino 2013]. Para o cálculo das forças atuantes na estrutura do quadricóptero, utilizou-se a equação da força e a do momento angular, as quais podem ser observadas, respectivamente, por (1) e (2):

$$F = \frac{dp}{dt} \quad (1)$$

Onde:

dp - Derivada do momento linear.

dt - Derivada do tempo.

F - Força.

$$N = \frac{dl}{dt} \quad (2)$$

Onde:

dl - Derivada do momento angular.

N - Torque.

Diante disso, as expressões matemáticas anteriores foram decompostas para que o estudo da atitude e altitude do quadricóptero fossem realizados. Assim, obteve-se as equações (3) e (4), as quais estão em conformidade com os resultados encontrados por [Jirinec 2011].

$$F = m \frac{dV}{dt} + W e \times V \quad (3)$$

Onde:

m - Massa.

dV - Derivada da velocidade linear.

We - Velocidade angular.

V - Velocidade linear.

$$N = I \frac{dW}{dt} e + W e \times (V \cdot I) \quad (4)$$

Onde:

I - Matriz identidade.

Isto posto, na Figura 1, é exposto um protótipo de um drone usado como referência na implementação citada.



Figura 1: Protótipo de um quadricóptero.

Posteriormente, notou-se a necessidade de comunicar os dois modelos desenvolvidos, já que tais modelagens iriam interagir entre si. Deste modo, a partir de [Gallasch and Kristensen 2001], percebeu-se indispensável a presença de um servidor elaborado em linguagem de programação Java, capaz de mediar a conexão entre os modelos citados. Logo, foi utilizado o NetBeans, ambiente de desenvolvimento integrado gratuito, versão 8.0, para a implementação do *Middleware* referido.

Portanto, as informações provenientes do MatLab ou do CPN Tools foram recebidas pelo servidor que posteriormente, as encaminhava para a RPC ou para o drone, respectivamente. A forma de conexão entre os *softwares* supracitados ocorreu por meio de socket, o qual Segundo [Kurose 2010], é uma interligação entre um servidor e um cliente. Destaca-se que ocorreu a aplicação do TCP (*Transmission Control Protocol*) na camada de transporte de dados via *socket*, proporcionando assim, confiabilidade na entrega de todas as informações. Para tanto, a Figura 2 apresenta o diagrama que sintetiza o que foi descrito.



Figura 2: Diagrama representando a interligação entre softwares utilizados na pesquisa.

Na Figura 2, constata-se a existência de dois clientes(MatLab e CPN Tools) e

somente um servidor(Java), os quais possuem uma interligação do tipo *full duplex*, ou seja, podem realizar transmissões e recepções, simultaneamente, durante toda a conexão. Ademais, na Tabela 1, constata-se os tipos de dados enviados e recebidos por cada bloco da Figura 2.

Tabela 1: Tipos de dados enviados e recebidos por cada bloco da Figura 2.

	Matlab	Servidor(Java)	CPN Tools
Envia	[Ack]	[Ack] / [x,y,z]	[x,y,z]
Recebe	[x,y,z]	[Ack] / [x,y,z]	[Ack]

A partir da Tabela 1, nota-se a presença de dois tipos de pacotes diferentes, o [Ack], responsável pela sinalização de recebimento das coordenadas espaciais, e o [x,y,z], o qual carrega as informações relativas ao posicionamento do equipamento simulado. Desta forma, o CPN Tools gera os pacotes do tipo [x,y,z] e os envia ao servidor(Java), os quais são retransmitidos ao drone simulado no MatLab. Posteriormente, o MatLab transmite um pacote do tipo [Ack] ao CPN Tools por intermédio do *Middleware* utilizado, quando o veículo aéreo chega aos valores alvos, inicialmente, definidos.

Vale ressaltar que a arquitetura de comunicação desenvolvida facilitará a supervisão de qualquer quadricóptero real, já que o bloco MATLAB, no qual executa-se a simulação do drone, poderá ser, facilmente, substituído por um quadrotor físico, respeitando-se, apenas, os padrões do TCP para que haja a comunicação do servidor (Java) com o veículo aéreo referido.

No que tange ao supervisor desenvolvido neste trabalho, a RPC modelada apresentou duas subpáginas, as quais interagem entre si para gerar os *setpoints*, isto é, valores alvos de posicionamento a serem transmitidos ao quadricoptero simulado no MatLab. Desta maneira, a Figura 3, demonstra a página principal do modelo proposto criado no CPN Tools.

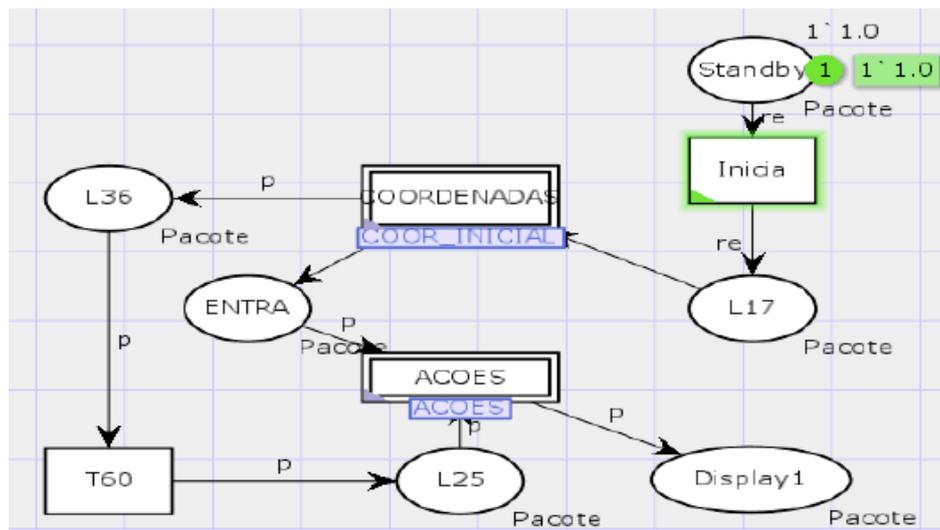


Figura 3: Comportamento geral da rede de Petri colorida desenvolvida.

Com base na Figura 3, nota-se a presença de quatro blocos principais, sendo o primeiro intitulado Inicia, responsável pelo estabelecimento da comunicação via *socket* entre o CPN Tools e o servidor desenvolvido. O segundo localiza-se ao lado esquerdo do bloco anterior, denominado COORDENADAS, que representa a primeira subpágina da RPC e é responsável pela geração de coordenadas aleatórias espaciais de posição e o

envio dessas ao MatLab para que sejam assumidas pelo quadrotor inicialmente.

Na parte inferior da figura em análise, nota-se o bloco nomeado por ACOES, o qual simboliza a segunda subpágina desenvolvida, atuando na criação de novas posições aleatórias do quadricóptero e, também, na transmissão de tais informações para o software MatLab. Enfim, o quarto e último bloco, designado Display1, armazena as posições alcançadas pelo drone durante toda a simulação, permitindo assim, o supervisionamento do equipamento controlado.

Com base nos antecedentes, pode-se inferir que na RPC, os retângulos sempre desempenham ações, os quais são chamados de transições e as elipses atuam como os estados do sistema modelado no tempo discreto, denominados lugares. Para um melhor detalhamento da modelagem criada no CPN Tools, na Figura 4 são expostos alguns componentes da subpágina chamada COORDENADAS.

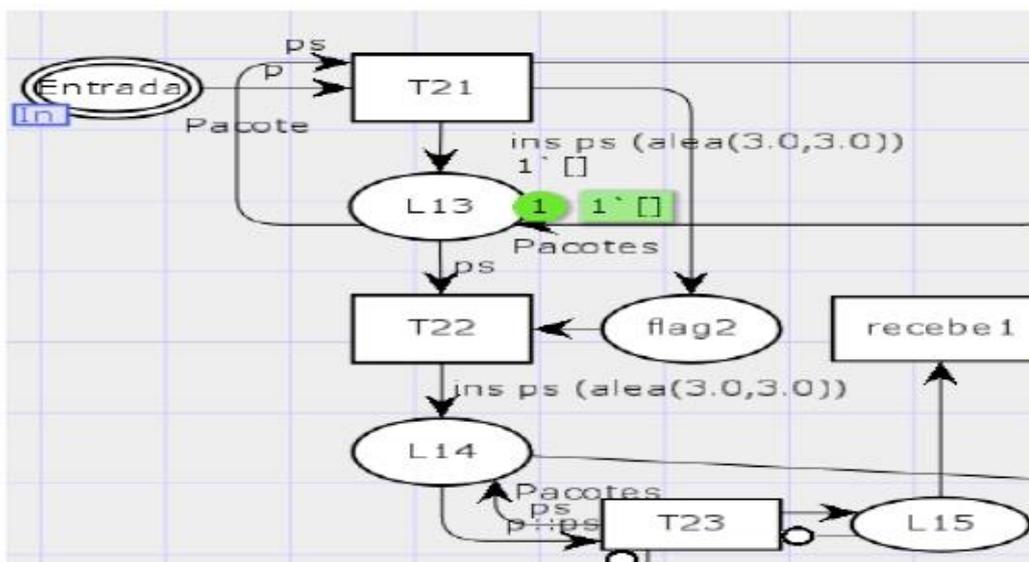


Figura 4: Comportamento geral da rede de Petri colorida desenvolvida.

A partir da Figura 4, o lugar designado Entrada, recebe uma sinalização por parte da página principal observada na Figura 3, informando que a conexão do CPN Tools com MatLab foi realizada. Após essa ação, as transições T21 e T22 conceberam as coordenadas horizontais x e y uma vez que a posição vertical inicial do quadricóptero sempre será zero. Já T23 faz envio dos dados para o quadricóptero. Em seguida, o MatLab envia uma confirmação da recepção das informações a RPC, essa recepção é representada pelo retângulo, recebe1.

É importante frisar, que os componentes da subpágina COORDENADAS, são semelhantes as encontradas na ACOES, existindo apenas um acréscimo de uma transição e de um lugar, os quais são responsáveis pela geração da coordenada vertical z.

3. Resultados e Discussão

Com base nos procedimentos técnicos metodológicos mencionados, foram iniciados os testes com os modelos desenvolvidos em conjunto com o Middleware, servidor em Java, com a finalidade de comprovar a funcionalidade do supervisor implementado.

Inicialmente, foram feitos 50 testes, os quais eram compostos por: 1) pela inicialização da comunicação entre a RPC modelada com o MatLab; 2) a geração de

coordenadas iniciais aleatórias; e 3) por fim novas localizações randômicas, com o intuito de englobar os possíveis *setpoints* envolvidos em uma teleoperação de um quadrotor real.

Após a verificação do perfeito funcionamento do sistema, realizou-se mais 100 ensaios, os quais proporcionaram a compreensão de como os *setpoints* gerados e enviados ao MatLab estavam relacionados com as resposta comportamentais do drone simulado. Tal fato está de acordo com os resultados de [Jensen 1997], pois segundo ele, as Redes de Petri permitem a identificação dos estados e ações de qualquer sistema modelado de modo claro, graças as representações gráficas disponíveis. Por meio da Figura 5, observa-se a página principal da RPC implementada, ao final de um ensaio.

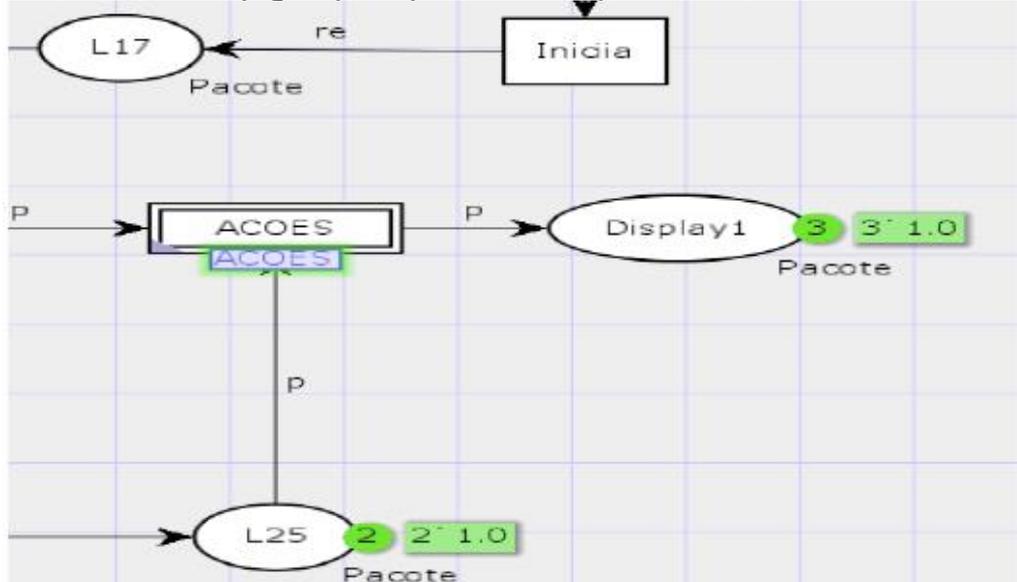


Figura 5: Página principal da RPC implementada ao final de um ensaio.

Na Figura 5, nota-se que ao fim do teste feito, o quadricóptero simulado assumiu as posições horizontais $x=1$ e $y=1$ e a coordenada vertical z como já citado, inicialmente, é igual a zero. Essas informações são explicitadas pelo lugar L25 na figura acima. Além do posicionamento inicial, pode-se constatar que novas coordenadas aleatórias de localização, $x=1$, $y=1$ e $z=1$, foram geradas a partir do supervisor e essas atingidas, posteriormente, pelo drone referido. Tais dados, são expostos no lugar, denominado Display1. Desta forma, a cada ensaio é possível efetuar o monitoramento do sistema controlado em tempo real. Para um melhor entendimento do que foi relatado, na Figura 6, expõem-se os *setpoints* sendo alcançados ao longo do tempo pelo quadrotor simulado no caso de posicionamento inicial, $x=1, y=1$ e $z=0$, configuração essa, anteriormente, abordada.

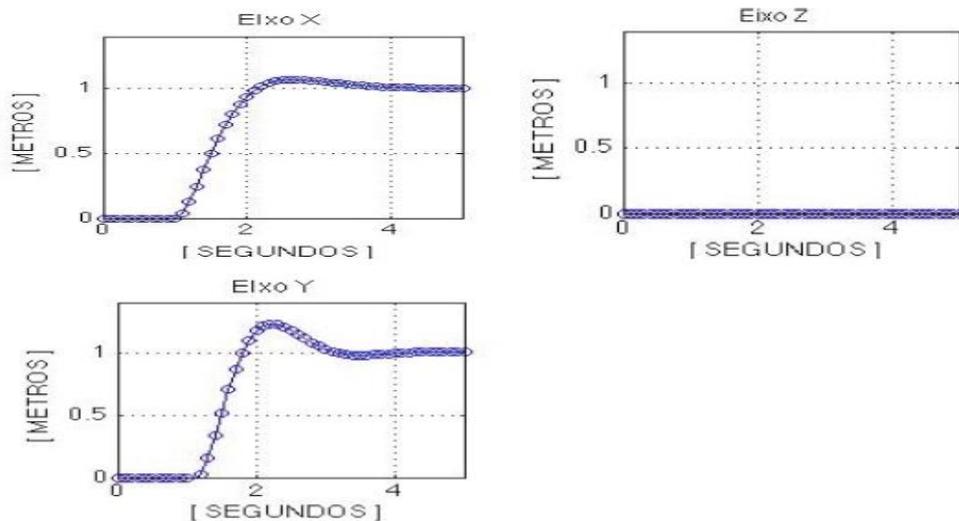


Figura 6: Setpoints sendo alcançados ao longo do tempo pelo quadricóptero simulado em sua configuração de posicionamento inicial.

De acordo com [Jensen 1997], as redes de Petri, também, possuem a capacidade de exibir as características dos Sistemas a Eventos Discretos. Isto posto, realizou-se uma análise da RPC modelada, através das ferramentas *built-in* encontradas no *software* CPN Tools. Dessa maneira, com a avaliação dos dados retornados a partir verificação citada, pode-se afirmar que não houve a presença de conflitos ou erros estruturais ou comportamental no modelo supervisor criado, garantindo o perfeito funcionamento da RPC implementada.

Os métodos aqui apresentados e discutidos estão em consonância aos usados por [Batista and Giovanni 2008], os quais propõem um sistema de navegação para robôs móveis terrestres por meio de Redes de Petri Coloridas. Entretanto, há diferenças entre as ações tomadas pelos equipamentos envolvidos e os níveis de sofisticação presentes nos sistemas de supervisionamento referidos.

4. Conclusão

Portanto, a partir da modelagem e simulação dos sistemas dinâmicos supracitados, percebeu-se uma operacionalidade menos complexa em relação aos métodos tradicionais, os quais apresentam uma extensa quantidade de variáveis numéricas usadas para o supervisionamento do drone. Tais parâmetros são contituídos por ângulos de guinada, arfagem e rolagem, velocidade de rotação de cada hélice, posicionamento espacial e torques presentes no equipamento supervisionado. Isto posto, pode-se inferir uma redução de custos referentes a mão-de-obra especializada, pois com utilização do modelo proposto, o quadricóptero não necessitará de pessoas, altamente, qualificadas para guiá-lo ou monitorá-lo.

Ademais, convém destacar que todos os procedimentos realizados em nível de simulação, poderão ser transferidos para uma aplicação prática, possuindo, teoricamente, os mesmos níveis de funcionalidade aqui apresentados e que no caso de uma RPC de supervisionamento mais sofisticada seja implementada, as ferramentas *built-in* do CPN Tools propiciarão a verificação de possíveis erros não previstos na modelagem.

Assim, qualquer quadricóptero será capaz de ser monitorado, remotamente, por meio do supervisor aqui apresentado, desde que tal equipamento siga as especificações de

conexão com o servidor desenvolvido nesta pesquisa.

5. Agradecimentos

Os autores agradecem a CAPES pelo suporte financeiro e a Fundação NUTEC, pelas apoio administrativo.

Referências

- TOC \f \n 1-9 Batista, Í. J. L. B. and Giovanni, C. (2008). Modelo de navegação para robôs móveis baseados em redes de petri coloridas. Master's thesis, Universidade Federal do Ceará, Fortaleza.
- Gallasch, G. and Kristensen, L. M. A. (2001). Comms/cpn: A communication infrastructure for external communication with design/cpn. In Third Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, pages 75–91. DAIMI.
- Gomes, R. C. and Aquino, F. J. A. (2013). Simulação de voo vertical de um quadricóptero usando software livre. Congresso Tecnológico InfoBrasil Ti — Telecom, pages 1–4.
- Jensen, K. (1997). Coulored petri net: Basic concepts. Springer, 2th edition.
- Jirinec, T. (2011). Stabilization and control of unmanned quadcopter. Master's thesis, Master of Science in Space Engineering-Czech Technical University, Prague.
- Kurose, J. F. e ROSS, K. (2010). Redes de Computadores e a Internet. Pearson, 5th edition.
- Murata, T. (1989). Petri nets: Properties, analysis and application. In Proceedings of the IEEE, pages 541–580. IEEE.