
MODELAGEM E SIMULAÇÃO DE REDES DE COMUNICAÇÃO BASEADAS NO PROTOCOLO CAN - CONTROLLER AREA NETWORK

Eduardo P. Godoy*
epgodoy@yahoo.com.br

Rafael V. Sousa*
rafael.sousa@gmail.com

Ricardo Y. Inamasu†
ricardo@cnpdia.embrapa.br

Wellington C. Lopes*
lito_lopes@hotmail.com

Arthur J. V. Porto*
ajvporto@sc.usp.br

*Departamento de Engenharia Mecânica, Escola de Engenharia de São Carlos, USP
Av. Trabalhador São carlense 400 CEP 13566-590 São Carlos - SP

†EMBRAPA Instrumentação Agropecuária, Rua XV de Novembro 1452 CEP 13560-970 São Carlos - SP

RESUMO

Uma grande tendência na área industrial tem sido a aplicação de tecnologias de redes de comunicação, como o protocolo CAN (Controller Area Network), como solução em sistemas de controle distribuído. O maior desafio no desenvolvimento de sistemas distribuídos baseados em redes industriais é o efeito dos atrasos de comunicação da rede no desempenho do sistema de controle. Este fato tem norteado o desenvolvimento de modelos e ferramentas de simulação para calcular esses atrasos, analisar o desempenho e simular o comportamento temporal desses sistemas. Sob este contexto, apresenta-se o desenvolvimento de um modelo matemático e de um modelo com Redes de Petri Coloridas para simulação de sistemas de controle distribuído baseados no protocolo CAN. Os modelos desenvolvidos apresentaram grande flexibilidade e potencial de aplicação, permitindo a obtenção de diversos parâmetros relacionados ao desempenho da rede e do sistema distribuído. Além disso, um caso de estudo referente à simulação de um sistema distribuído com rede CAN para um robô móvel permitiu evidenciar que os modelos desenvolvidos podem ser utilizados para a análise de qualquer aplicação de sistemas de controle distribuído com

redes CAN.

PALAVRAS-CHAVE: Protocolo CAN, Redes de Petri Coloridas, Modelo Matemático, Atrasos de Comunicação

ABSTRACT

Modeling and Simulation of CAN-Based Communication Networks

A major trend in industrial systems is the application of communication networks technologies, such as CAN (Controller Area Network), in distributed control systems. A challenging problem in the development of distributed system based on communication networks is the network delay effects in the performance of the system. This fact has motivated the development of simulation tools and models to calculate these delays, to simulate the timing behavioral and evaluate the performance of these systems. This work presents the development of a mathematical model and a Colored Petri Nets model to the simulation of CAN-based distributed systems. The flexibility and potential obtained with the developed models allow determining and analyzing several performance parameters related to communication network and the distributed control. An example of a CAN-based distributed system applied to perform the control of a mobile robot demonstrates the functionalities of the models and its

Artigo submetido em 18/08/2009 (Id.: 01027)

Revisado em 06/11/2009, 04/02/2010

Aceito sob recomendação da Editora Associada Profa. Emília Villani

applicability to the analysis of any CAN application.

KEYWORDS: CAN protocol, Colored Petri Nets, Mathematical model, Network delays

1 INTRODUÇÃO

As arquiteturas tradicionais de comunicação e controle ponto-a-ponto, implementadas nos sistemas de manufatura industriais nos últimos tempos, são compostas por cabeamento conectando o computador ou dispositivo central de controle a cada sensor e atuador do sistema. Este tipo de controle tradicional e centralizado, no entanto não fornece os novos requisitos de desenvolvimento de sistemas de controle como modularidade, controle descentralizado, facilidade de diagnóstico e baixo custo (Yang, 2006).

Nos sistemas de manufatura atuais, a introdução de arquiteturas de redes baseadas em barramento ou *fieldbus* podem melhorar a eficiência, a flexibilidade e a confiabilidade do sistema, reduzindo o tempo e os custos de instalação e manutenção (Moyne e Tilbury, 2007). Essa tecnologia de redes industriais, com suas vantagens, foi rapidamente absorvida para satisfazer as necessidades de comunicação entre sistemas e equipamentos aplicados em automação e controle (Hou *et alli*, 2004).

Em sistemas de controle distribuído baseados em redes de comunicação, os dispositivos conectados têm que compartilhar da melhor maneira possível o meio disponível para a troca de informações e ainda cumprir com requisitos temporais de transmissão de dados. Como solução para este problema, diversos protocolos de comunicação têm sido pesquisados e desenvolvidos (Moyne e Tilbury, 2007) como o Profibus, Foundation Fieldbus, Modbus e CAN.

O CAN (Bosch, 2006) é um dos protocolos de comunicação mais aplicados em sistemas de controle distribuído, sendo utilizado em diversas áreas como robótica, automação da manufatura, controle de processos e eletrônica embarcada (Othman *et alli*, 2006) através de variados padrões como DeviceNet, CANOpen, J1939 e SDS (CIA, 2006). Entre os fatores que norteiam essa grande utilização do CAN estão seu baixo custo de desenvolvimento, vasta disponibilidade de dispositivos (microcontroladores, DSPs) com controladores CAN embarcados e características interessantes como acesso priorizado à rede de comunicação, robusto método de controle de erros e arbitragem não destrutiva que determinam sua grande aceitação no meio industrial e acadêmico.

Não importando o protocolo de comunicação utilizado, o desempenho e a estabilidade de um sistema distribuído podem ser afetados pela presença de atrasos de comunicação (*delay*) entre os sensores, atuadores e controladores do sistema (Bail-

lieul e Antsaklis, 2007). Esses atrasos de comunicação são resultados do compartilhamento do meio de comunicação com outras malhas de controle como também do tempo de processamento requerido para codificação dos sinais e computação dos dados (Lian *et alli*, 2002).

Uma das principais dificuldades relacionadas ao projeto e utilização de sistemas de distribuídos é o desenvolvimento de modelos e ferramentas de análise para estudar a influência de parâmetros de configuração no desempenho do sistema e simular seu comportamento e operação (Cervin *et alli*, 2003). Esse fato tem motivado o desenvolvimento de vários trabalhos relacionados a modelagem e ferramentas de simulação para diferentes tipos de redes e protocolos de comunicação (Torngren *et alli*, 2006). A importância desse tipo de pesquisa sobre modelagem e simulação para sistemas de controle distribuídos com redes CAN pode ser demonstrada por trabalhos recentes como Nolte (2007), Rodriguez e Menendez (2007), Bago *et alli* (2007) e Kumar *et alli* (2009).

Nesse trabalho são apresentados dois modelos desenvolvidos (e validados em trabalhos anteriores dos autores), sendo um modelo matemático e um modelo com Redes de Petri Coloridas, para simulação de sistemas distribuídos baseados no protocolo CAN (Godoy, 2007 e Lopes, 2007). Os modelos desenvolvidos apresentaram características importantes como a incorporação ao modelo matemático de erros na transmissão de mensagens, tempo de retransmissão dessas mensagens e cálculo do valor de otimização da utilização da rede e a possibilidade de determinar, com o modelo em Redes de Petri, a quantidade de mensagens nos *buffers* de transmissão e recepção e a relação entre o tempo de espera e o tempo total de transmissão de cada mensagem simulada na rede CAN. Os modelos apresentados permitiram calcular e analisar diversos parâmetros de desempenho de sistemas distribuídos com redes CAN, como tempo de transmissão de mensagens, taxa de utilização da rede, atrasos de comunicação e verificação do cumprimento de *deadlines*, possibilitando simular a operação e o comportamento temporal desses sistemas. Um caso de estudo é proposto e demonstra o potencial de utilização dos modelos desenvolvidos na simulação e análise de um sistema distribuído com rede CAN para um robô móvel.

Esse artigo está organizado da forma descrita a seguir. Após esta introdução, a seção 2 apresenta uma revisão da literatura discutindo trabalhos relacionados ao assunto e expondo sobre o estado da arte na área. A seção 3 detalha o processo de desenvolvimento dos dois modelos. A seção 4 apresenta um caso de estudo proposto para demonstrar a aplicação dos modelos e analisa resultados obtidos referentes a parâmetros de desempenho relacionados a sistemas distribuídos com redes CAN. A seção 5 encerra o artigo, apresentando as conclusões do trabalho.

2 REVISÃO DE LITERATURA

O aumento crescente na utilização de sistemas distribuídos baseado em redes de comunicação em aplicações de controle realça a importância de pesquisas relacionadas a determinação de métricas de desempenho da rede de comunicação e do sistema de controle distribuído (Lian *et alli*, 2002). De acordo com Acton *et alli* (2006), métricas como taxa de utilização da rede, atrasos de comunicação, tempo de transmissão de mensagens, *jitter* e tempo de resposta estão relacionados com o desempenho da rede de comunicação. Em relação a sistemas distribuídos, as métricas de desempenho geralmente utilizadas são cumprimento de *deadline*, taxa de utilização do processador, tempo de resposta e latência de execução de tarefas, escalonamento e priorização de mensagens.

Lian *et alli* (2002) evidenciam que o desempenho de sistemas distribuídos pode ser influenciado por alterações em seus parâmetros de configuração como velocidade de transmissão de dados da rede, tamanho das mensagens, prioridades das mensagens e períodos de amostragem dos dispositivos. Nolte (2007) afirma que a presença do *jitter* nas tarefas de computação (codificação) e comunicação podem causar uma deterioração no desempenho de controle. Características específicas de redes CAN como o acesso priorizado à rede e método de arbitragem não destrutiva norteiam sua aplicação em sistemas distribuídos onde é necessário o cumprimento de requisitos temporais (*deadline*) de mensagens (Kumar *et alli*, 2009). Nesses sistemas, as mensagens têm que ser transmitidas corretamente em um tempo limitado e menor que seu período de amostragem. Caso as mensagens não sejam transmitidas ou o sistema apresente um alto valor para seu tempo de transmissão, sobreposição e perdas de mensagens podem ocorrer deteriorando o desempenho do sistema distribuído (Al-Hammouri *et alli*, 2008). Essas constatações enfatizam a importância de pesquisas para determinação do tempo de transmissão de mensagens, atrasos de comunicação e cumprimento de *deadlines* em sistemas distribuídos com redes CAN (Nolte, 2007 e Xiaodong e Yanjun, 2002).

Para determinar essas métricas de desempenho e assim estudar a influência de parâmetros de configuração no desempenho de sistemas distribuídos, diversos trabalhos têm desenvolvido modelos específicos para simulação de redes de comunicação (Torngren *et alli*, 2006). No caso específico de redes CAN, os trabalhos de Tindell *et alli* (1995), que usa técnicas determinísticas (pior caso) para desenvolver equações relacionadas à transmissão de mensagens na rede, e o trabalho de Cavalieri *et alli* (1996), que aborda com sucesso a utilização de Redes de Petri Estocásticas na modelagem de uma rede CAN, tornaram-se referência para o desenvolvimento de outros modelos e ferramentas de análise de redes CAN (Davis *et alli*, 2006 e Portugal *et alli*, 2005).

Seguindo as discussões e equações originais do trabalho de Tindell *et alli* (1995), Punnekkat *et alli* (2000) propõem termos adicionais para as equações para analisar o efeito na rede CAN da presença de erros na transmissão de mensagens devido a interferências eletromagnéticas. Parâmetros de configuração de redes CAN que afetam o tempo de transmissão de mensagens e condições para garantir o cumprimento de *deadlines* são investigados por Xiaodong e Yanjun (2002). Davis *et alli* (2006) revisam as equações originais e sugerem modificações para lidar corretamente com o mecanismo de prioridades de redes CAN. Nolte (2007) propõe um modelo probabilístico para investigar os efeitos do bit *stuffing* no tempo de transmissão de mensagens em redes CAN e determinar valores mais realistas para a taxa de utilização da rede.

O conceito de Redes de Petri proposto por Cavalieri *et alli* (1996) para modelagem de redes CAN também foi utilizado por outros trabalhos. Portugal *et alli* (2005) desenvolvem um modelo em Redes de Petri Estocásticas para analisar a confiança da comunicação em redes CAN em cenários de presença de falhas. Bago *et alli* (2008) apresentam detalhadamente o desenvolvimento e a validação de um modelo em Redes de Petri Coloridas para sistemas com redes CAN. Kumar *et alli* (2009) afirmam que o tempo de transmissão de mensagens em redes CAN apresentam uma distribuição probabilística e apresenta um modelo baseado em Redes de Petri Temporizadas para a determinação desses valores em sistemas com mensagens periódicas.

Diferentes abordagens para modelagem e simulação de redes CAN também podem ser encontradas na literatura recente. Bago *et alli* (2007) utilizam a ferramenta TrueTime (Cervin *et alli*, 2003) para a simulação de redes CAN e Rodriguez e Menendez (2007) apresentam o desenvolvimento de três modelos para estimar a taxa de utilização da rede CAN e para prever futuros valores para os atrasos de comunicação da rede. Neste trabalho, os autores afirmam que o comportamento de redes CAN é estocástico e por isso os modelos foram construídos a partir de dados estatísticos de redes CAN e utilizaram técnicas de modelagem baseadas em Cadeias de Markov, em redes neurais artificiais recursivas e em abordagem de histogramas. A modelagem e simulação de redes CAN utilizando técnicas de sistemas a eventos discretos com o software OPNET é demonstrada em Bayilmis *et alli* (2006).

3 MODELAGEM DE SISTEMAS DISTRIBUÍDOS COM REDES CAN

3.1 Protocolo CAN

O CAN (Bosch, 2006) é um protocolo de comunicação digital serial e não determinístico, onde a comunicação de dados

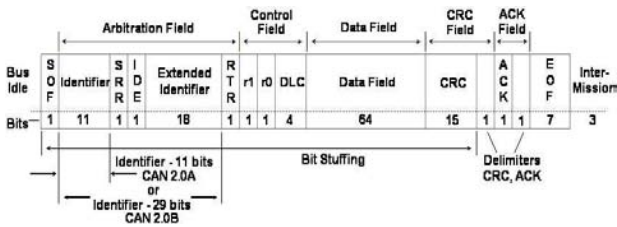


Figura 1: Formatos do Quadro de Dados de uma Mensagem CAN

é baseada em mensagens formadas por quadros de bits com determinada função. Entre esses quadros de bits, existe o campo identificador (*identifier ID*) que caracteriza e define a prioridade de cada mensagem. O valor do identificador de uma mensagem em uma rede CAN é exclusivo e quanto mais baixo seu valor, maior será a prioridade da mensagem. Os sinais elétricos digitais do CAN são representados pelo nível recessivo (nível lógico 1) e nível dominante (nível lógico 0), sendo eles sinais diferenciais entre os dois fios da rede.

O mecanismo de acesso ao meio é baseado no conceito CSMA/CD com NBDA (*Carrier Sense Multiple Access / Collision Detection with Non-Destructive Bitwise Arbitration*), que significa que os nós CAN possuem acesso à rede com prioridades definidas. Ao verificar o status do barramento, os módulos iniciam a transmissão de suas mensagens. De acordo com o valor do identificador, o módulo com a mensagem de prioridade menor cessa sua transmissão e o módulo com a mensagem de maior prioridade continua enviando sua mensagem deste ponto, sem ter que reiniciá-la. Isto é realizado pelo processo de arbitragem bit a bit não destrutivo, ou lógica "E" por fios, quando dois ou mais módulos iniciam a transmissão simultaneamente. Cada bit transmitido é comparado, sendo que o dominante sobrepõe o recessivo.

Dentre as especificações do protocolo CAN em relação à camada de enlace de dados, estão os formatos existentes do quadro de dados. São definidos dois formatos de quadros de mensagem, onde a única diferença está no tamanho do identificador, sendo CAN 2.0 A *Standard* (ID 11 bits) e CAN 2.0 B *Extended* (ID 29 bits) especificados segundo a Figura 2.

3.2 Modelo Matemático

Godoy (2007) apresenta uma revisão sobre equações de análise e sistematiza um modelo matemático de redes CAN para calcular métricas de desempenho como taxa de utilização da rede, tempo de transmissão de mensagens e cumprimento de *deadlines*. A Figura 3 apresenta os tempos encontrados na transmissão de uma mensagem CAN pela rede, desde sua inicialização ou começo da transmissão por um nó de origem

até o término da recepção da mensagem pelo nó de destino.

O tempo total de transmissão da mensagem, T_{delay} , pode ser dividido em três partes: atrasos de comunicação no nó de origem, T_{src} , no barramento CAN, T_{bus} , e no nó de destino das mensagens, T_{dest} , como mostrado na Figura 3. O atraso do nó de origem, T_{src} , é caracterizado pelo tempo de pré-processamento, T_{pre} , que é a soma do tempo de computação, T_{scomp} , com o tempo de codificação, T_{scode} , realizado no início de cada mensagem. O tempo de espera total, T_{wait} , é caracterizado pela soma do tempo de espera na fila, T_{queue} , com o tempo de bloqueio, T_{block} . O tempo de espera na fila, T_{queue} , é o tempo que uma mensagem espera no buffer do nó de origem enquanto outra mensagem da fila está sendo transmitida. Esse valor depende do tempo de bloqueio das outras mensagens na fila, do período de amostragem das mensagens e da carga de dados a ser processada. O tempo de atraso do barramento CAN, T_{bus} , é caracterizado pela soma do tempo total de transmissão da mensagem, T_{frame} , com o atraso de propagação da rede, T_{prop} . Esse valor depende do tamanho da mensagem, da velocidade de transmissão e do tamanho do barramento CAN. O atraso no nó de destino, T_{dest} , é caracterizado pelo tempo de pós-processamento, T_{post} , que é a soma do tempo de decodificação, T_{dcode} , com o tempo de computação, T_{dcomp} , realizado no final da transmissão de cada mensagem.

A equação (1) apresenta resumidamente o tempo total de atraso na rede de controle CAN. Os tempos T_{pre} e T_{post} são relacionados com os tempos de processamento e transformação dos dados entre o barramento e o dispositivo que utiliza esses dados. Estes tempos estão totalmente relacionados e dependem das características do software e do hardware utilizados nas unidades de controle CAN. Segundo Lian *et alii* (2001), estes valores T_{pre} e T_{post} são tipicamente constantes e pequenos. Por causa desse fato e para não excluí-los dos cálculos, uma relação desses valores com o *Jitter* (variação do tempo nominal esperado de mensagens serem inseridas e retiradas da fila de mensagens) da rede foi adotada.

$$\begin{aligned}
 T_{delay} &= T_{src} - T_{dest} \\
 &= T_{pre} + T_{wait} + T_{bus} + T_{post} \\
 &= \underbrace{T_{pre} + T_{post}}_{J_m} + \underbrace{T_{wait}}_{Q_m} + \underbrace{T_{bus}}_{C_m}
 \end{aligned}
 \tag{1}$$

As equações que definem os termos mostrados em (1) são descritos para redes de controle CAN e uma relação entre (1) e (2) é apresentada. De acordo com as equações sistematizadas em Godoy (2007), a equação (2) representa o tempo de transmissão total de uma mensagem (m) em uma rede CAN.

$$R_m = J_m + Q_m + C_m
 \tag{2}$$

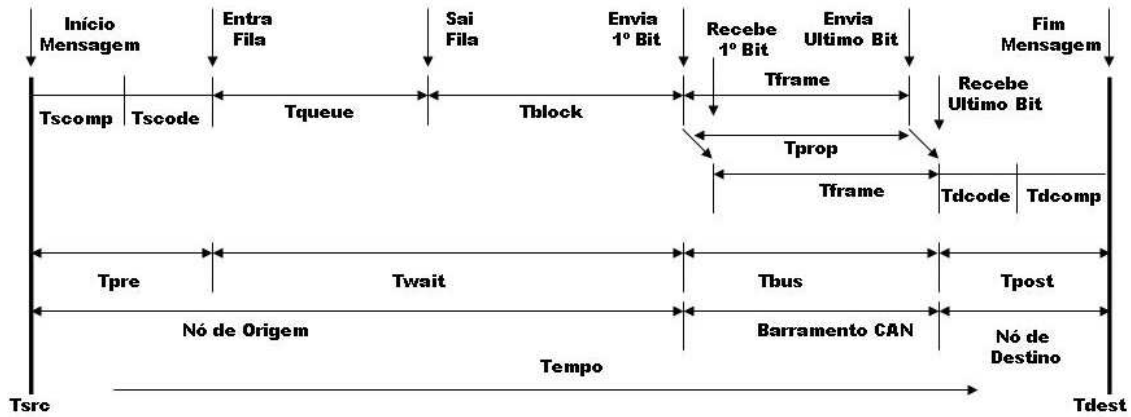


Figura 2: Diagrama Temporal da Transmissão de uma Mensagem CAN (Adaptado de Lian *et alli*, 2001)

O termo J_m representa o *Jitter* da mensagem e é determinado empiricamente, utilizando-se J_m igual 0,1 milissegundo (ms). O termo Q_m representa o tempo gasto por uma mensagem na fila de espera com a rede operando sob condições de erro (o maior tempo entre a inserção de uma mensagem na fila de prioridades e o início de sua transmissão) e o termo C_m representa o tempo gasto para se transmitir uma mensagem fisicamente sobre a rede.

O formato do quadro de dados de uma mensagem CAN contém uma largura de *stuff* (quantidade de bits consecutivos que podem ser transmitidos com o mesmo valor 0 ou 1) de 5 bits e uma quantidade O de bits de *overhead* (requerido para verificação de erros e controle de transmissão) por mensagem. Apenas um número T desta quantidade O de bits de *overhead* está sujeito ao *stuffing* (campos não sujeitos ao bit *stuffing* na Figura 1: *Delimiters CRC, ACK e EOF*). Assim, a equação (3) define o termo C_m da equação (2) apresentada.

$$C_m = \left(\left\lfloor \frac{T + 8S_m - 1}{4} \right\rfloor + O + 8S_m \right) \cdot \tau_{bit} \quad (3)$$

em que T = Quantidade de bits de *overhead* sujeita ao *bit stuffing*,

O = número máximo de bits de *overhead*,

S_m = tamanho limitado da mensagem (m) em bytes (máximo de 8 bytes);

τ_{bit} = tempo para transmitir um bit sobre o barramento (encontrado de acordo com a velocidade de transmissão: em bits por segundo – bit/s).

Godoy (2007) define que os valores das variáveis T e O são diferentes para os formatos do quadro de dados da mensagem CAN. Para o formato CAN 2A, os valores de T e O são

34 e 47 e para o formato CAN 2B, os valores são 54 e 67 respectivamente.

O tempo que uma mensagem CAN (m) fica na fila de espera depende do tempo de bloqueio por mensagens de menor prioridade que estejam ocupando o barramento no momento (B_m) e do tempo de transmissão das mensagens de maior prioridade. Este tempo Q_m é dado por uma relação de recorrência (4) com valor inicial $Q_m^0 = 0$ e com interações até a convergência (i.e. $Q_m^{n+1} = Q_m^n$).

$$Q_m^{n+1} = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{Q_m^n + J_j + \tau_{bit}}{T_j} \right\rceil \cdot C_j + E_m(Q_m + C_m) \quad (4)$$

em que $hp(m)$ = conjunto de mensagens no sistema com prioridade maior que a mensagem (m),

B_m = tempo de bloqueio no pior caso da mensagem (m) e é dado por (5),

T_j = período de uma mensagem (j),

J_j = *Jitter* de uma mensagem (j).

$$B_m = \max_{\forall k \in lp(m)} (C_k) \quad (5)$$

na qual que $lp(m)$ = conjunto de mensagens com prioridade menor que a mensagem (m) (se a mensagem m for a de menor prioridade, então B_m é zero).

Os valores de C_j , C_k utilizados nas equações (4) e (5) são obtidos da equação (3).

A equação (6) apresenta a função de recuperação de erro dada pelo termo $E_m(t)$, que define o valor de bits de *overhead* de erros que podem ocorrer num intervalo de tempo (t).

$$E_m(t) = \left(n_{error} + \left\lceil \frac{t}{T_{error}} \right\rceil - 1 \right) \cdot (31 \cdot \tau_{bit} + \max_{\forall K \in hp(m) \cup \{m\}} (C_k)) \quad (6)$$

em que n_{error} = número de erros em seqüência que podem ocorrer em um intervalo arbitrário,

T_{error} = período de ocorrência do erro.

São definidos os valores de n_{error} igual a 1 e T_{error} igual 100 ms, de acordo com Tindell *et alli* (1995), para utilização da equação (6).

Em cada erro o *overhead* de recuperação de erros pode ser aumentado em 31 bits, seguido pela retransmissão da mensagem. Apenas mensagens de prioridade maior que a mensagem (m) e ela mesma podem ser retransmitidas e atrasar a mensagem (m). A equação (7) determina a maior destas mensagens.

$$\max_{\forall K \in hp(m) \cup \{m\}} (C_k) \quad (7)$$

Outro parâmetro muito utilizado para avaliação de desempenho de redes de controle CAN é a taxa de utilização do barramento. A equação (8) define esse o cálculo desse parâmetro.

$$U = \sum_{i=1}^N \frac{C_i}{T_i} \quad (8)$$

em que C_i = tempo de transmissão da mensagem i no barramento e é dado por (3);

T_i = período de amostragem da mensagem i ;

N = número total de mensagens na rede.

O conjunto de equações apresentado constitui um modelo matemático, o qual foi implementado computacionalmente e validado a partir da comparação entre dados obtidos com o modelo e dados obtidos de uma rede CAN real em laboratório (Godoy, 2007). O software com o modelo matemático e instruções de uso está disponível para download em: <http://www.simulacao.eesc.usp.br/~egodoy/mestrado.html>. A partir desse modelo matemático é possível obter parâmetros e informações de operação do sistema para serem

utilizados na análise de desempenho. Entre esses parâmetros estão:

- Tempo de Transmissão de cada Mensagem: Tempo entre uma mensagem começar a ser transmitida por um dispositivo de origem e ser recebida pelo dispositivo de destino;
- Tempo de Resposta do Sistema: Tempo total de leitura ou transmissão de mensagens de todos os dispositivos da rede;
- Taxa de Utilização da Rede / das Mensagens: Porcentagem utilizada pelo tráfego de mensagens (dados + *overhead*) em relação à capacidade total de uma rede CAN e porcentagem utilizada pela transmissão de mensagens de dados em relação à capacidade total de uma rede CAN;
- Valor de otimização da utilização da rede: corresponde ao valor pelo qual se podem dividir todos os períodos de amostragem das mensagens para otimizar e atingir a taxa de utilização máxima da rede CAN;
- Cumprimento de *Deadline*: Verifica-se o cumprimento do requisito temporal de cada mensagem, o qual demanda que seu tempo de transmissão seja menor que o seu período de amostragem para que não ocorra sobreposição e perda de mensagens na rede CAN.

3.3 Modelo com Redes de Petri Coloridas

As redes de Petri constituem um formalismo matemático e gráfico bastante eficaz na modelagem de sistemas que apresentam atividades concorrentes e competição na utilização de recursos limitados (Murata, 1989).

Um modelo de Rede de Petri é descrito por quatro elementos básicos: Lugar, Transição, Arco e Marca.

- Lugares: graficamente são representados por um círculo e modelam os estados do sistema. Representam as condições para que um determinado evento aconteça;
- Transições: graficamente são representados por um retângulo e modelam os eventos do sistema, que representam as mudanças de estados;
- Arcos: Graficamente são representados por uma seta, interligam um lugar e uma transição, e representam como é a seqüência com que acontecem as mudanças de estado dentro do sistema;
- Marcas: Graficamente são representadas por um círculo de cor preta e representam o comportamento dinâmico

do sistema, indicando os estados ativos, ou seja, que estão sendo utilizados pelo sistema;

Neste trabalho, utiliza-se a abordagem de Cavalieri *et alli* (1996) como orientação ao desenvolvimento de um modelo para análise e simulação de sistemas distribuídos com redes CAN. Diferencia-se, porém do trabalho citado por utilizar o formalismo da Rede de Petri Colorida (CPN – *Colored Petri Nets*) e não o formalismo de Rede de Petri Estocástica ou da Rede de Petri Temporizada.

A utilização da Rede de Petri Colorida amplia as possibilidades de modelagem, pois permite distinguir os elementos de acordo com suas funções específicas no processo de modelagem. Outras vantagens na utilização da Rede de Petri Colorida são a diminuição no tamanho e na complexidade (facilita compreensão) dos modelos utilizados para analisar grandes sistemas, além de oferecer recursos para analisar modelos com maior poder de abstração (Girault e Valk, 2003). A disponibilidade de um programa de edição e simulação de Redes de Petri Coloridas permite a simulação e a análise de desempenho dos modelos obtidos.

A ferramenta de modelagem e simulação utilizada neste trabalho foi a DesignCPN, que utiliza o formalismo da Rede de Petri Colorida (Christensen *et alli*, 1997). A utilização desse software se justifica por disponibilizar grande quantidade de ferramentas que facilitam a elaboração de modelos, manipulação e principalmente a extração das informações coletadas durante as simulações.

Com o propósito de elaborar um modelo mais simples e capaz de explorar uma quantidade maior de recursos oferecidos pelo formalismo da Rede de Petri Colorida e pela ferramenta de simulação DesignCPN, foi desenvolvido um modelo da rede CAN utilizando o conceito de multimarcas. O modelo é apresentado na Figura 4.

É importante enfatizar que com a possibilidade de utilizar lugares com diversas marcas (ou cores), o modelo desenvolvido possui uma única estrutura capaz de representar uma quantidade diversificada de dispositivos ou nós CAN configurados pelo usuário, por exemplo, sensores, atuadores e controladores. Essa característica obtida com a utilização do conceito de multimarcas fornece grande flexibilidade de alteração e expansão para o modelo da rede CAN desenvolvido em Redes de Petri Colorida neste trabalho, pois o usuário não precisa alterar a estrutura do modelo para inserir ou retirar dispositivos na rede CAN ou para alterar suas configurações como prioridade de acesso a rede e período de amostragem.

As principais definições de cores utilizadas no modelo da rede CAN desenvolvido e mostrado na Figura 4 são:

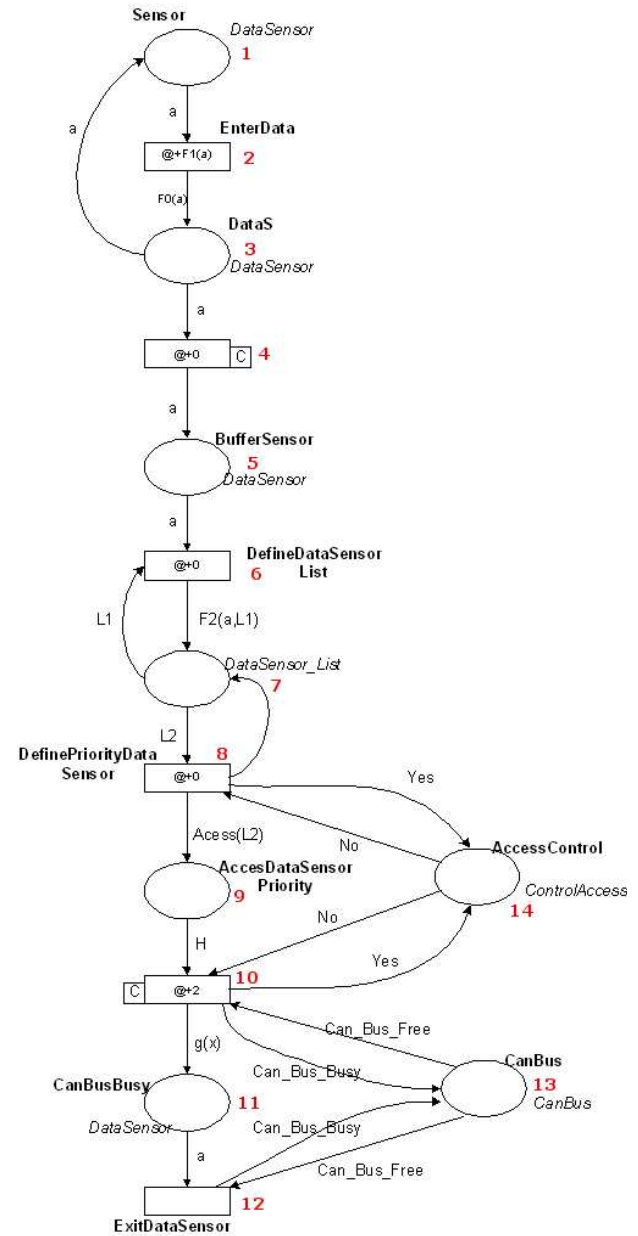


Figura 3: Modelo em Rede de Petri Colorida para um Sistema Distribuído com Rede CAN (Lopes, 2007) – Legenda na Tabela 1

- DataSensor: são as cores que diferenciam os dispositivos que serão utilizados na rede de comunicação, no caso temos Data_S1, Data_S2, Data_S3, Data_S4, Data_S5, Data_S6;
- CanBus: cor que identifica o estado do barramento, que pode ser Can_Bus_Busy para representar que o mesmo esteja ocupado, ou Can_Bus_Free para representar que o barramento esteja livre;
- Data_Sensor_List: define uma estrutura de lista para ar-

Tabela 1: Descrição dos Elementos Gráficos do Modelo da Figura 4

Nº	Tipo	Nome	Descrição
1	Lugar	Sensor	Representa geração de um pacote de dados de diferentes dispositivos
2	Transição	EnterData	Atribui um período de amostragem para os pacotes gerados
3	Lugar	DataS	Realimentação para o Lugar “Sensor”
4	Transição	DataSTrue	Controle na geração das mensagens
5	Lugar	BufferSensor	Representa a Fila de Mensagens no Buffer
6	Transição	DefineDataSensorList	Monta a lista de mensagens armazenadas no Buffer
7	Lugar	DataSensor_List	Atualiza o Buffer de Mensagens
8	Transição	DefinePriorityDataSensor	Método de Arbitragem
9	Lugar	AccessDataSensorPriority	Arbitragem - Define qual mensagem irá obter o acesso a rede CAN
10	Transição	EnterDataSensor	Arbitragem - Define quanto tempo a mensagem ocupará a rede
11	Lugar	CanBusBusy	Representa que o Status da rede CAN está Ocupado ou Livre
12	Transição	ExitDataSensor	Essa transição retira a mensagem da rede CAN (recebimento pelo destinatário)
13	Lugar	CanBus	Representa o Barramento ou Rede CAN
14	Lugar	AccessControl	Define qual mensagem terá acesso a rede CAN

mazenar as cores dos diferentes sensores.

A Tabela 1 apresenta uma legenda com a descrição básica de cada elemento do modelo desenvolvido para um sistema distribuído com rede CAN. Uma explicação detalhada desse modelo bem como a implementação das funções para cálculo e obtenção das métricas de desempenho requeridas, como por exemplo, tempo de transmissão de mensagens e taxa de utilização da rede CAN, pode ser encontrada em Lopes (2007).

O modelo apresentado segue os padrões definidos pelo formalismo da Rede de Petri Colorida e do software de simulação Design CPN. De acordo com o software utilizado, os lugares são identificados no modelo da Figura 4 pelos textos em negrito. As marcas, ou cores, possuem seus nomes apresentados em itálico.

A partir do modelo em Redes de Petri Colorida desenvolvido é possível selecionar parâmetros de configuração, determinar métricas de desempenho e obter informações de operação do sistema distribuído com rede CAN para serem utilizados nas simulações e análise de desempenho. Entre esses parâmetros estão:

- Quantidade de mensagens geradas por dispositivo - armazenamento e análise da quantidade de mensagens geradas pelos dispositivos durante um intervalo de simu-

lação;

- Quantidade de Mensagens no *Buffer* - parâmetro muito importante para análise de desempenho de uma rede CAN. O *buffer* é uma memória para armazenamento de mensagens que não conseguiram acesso ao barramento CAN, evitando que as mesmas sejam perdidas. Porém o tamanho do *buffer* varia de acordo com a aplicação e com recursos de hardware e software, utilizados para configuração da rede CAN;
- Quantidade de Acessos ao Barramento - essa informação será utilizada para analisar a taxa de ocupação do barramento durante seu funcionamento, permitindo a análise da quantidade de acessos de todos os dispositivos e também analisar quanto cada dispositivo está ocupando o barramento.
- Taxa de Utilização da Rede - porcentagem utilizada pela transmissão de mensagens de dados em relação à capacidade total de uma rede CAN;
- Intervalo Médio de Acessos – quando uma rede CAN é utilizada para finalidades de controle, as mensagens precisam ser entregues ao seu destinatário dentro de um intervalo de tempo definido para que o processo de controle não tenha seu desempenho deteriorado (*deadline*). Devido à importância da análise do tempo de espera da mensagem para ser transmitida na rede, foi imple-

mentada uma maneira de analisar o intervalo médio de acesso por dispositivo simulado.

4 SIMULAÇÃO DE SISTEMAS DISTRIBUÍDOS COM REDES CAN

4.1 Proposta de Caso de Estudo

Para a utilização dos modelos (matemático e Redes de Petri Coloridas) desenvolvidos, torna-se necessário a definição de parâmetros de configuração referentes ao protocolo CAN e ao sistema de controle que será analisado. Esses parâmetros necessários são: velocidade de transmissão de dados da rede CAN, número de mensagens do sistema de controle, formato do quadro de dados da mensagem CAN (CAN 2.0A / CAN 2.0B), período de amostragem (ms) das mensagens dos dispositivos, tamanho (bits) do campo de dados das mensagens, esquema de prioridade de acesso das mensagens à rede CAN.

Para demonstrar a aplicação dos modelos desenvolvidos, foram propostas suas utilizações para obtenção dados relacionados à simulação de operação de uma rede CAN. Essa rede CAN é formada por um determinado conjunto de mensagens referente aos equipamentos conectados na estrutura de um robô móvel, conforme mostrado na Figura 5. Os dados obtidos visam contribuir para o estudo da utilização e do comportamento temporal da rede CAN projetada, da influência dos parâmetros de configuração no desempenho do sistema de controle e de sua viabilidade de aplicação para o controle do robô.

A Tabela 2 apresenta uma descrição das mensagens e seus respectivos parâmetros de configuração propostos para o caso de estudo do robô. Na Tabela 2, a coluna P representa o esquema de prioridades das mensagens e a coluna T representa o período de amostragem da mensagem. Essa configuração de prioridades das mensagens foi determinada, selecionando-se maior prioridade para mensagens de controle e com amostragem mais rápida. Para melhor entendimento do funcionamento do robô, por exemplo, o dispositivo N° 11 (monitoramento) é responsável por medir a velocidade do motor de propulsão 3 através de um encoder e transmitir essa informação na rede CAN em uma mensagem com prioridade 11 e período de amostragem de 100 ms. O dispositivo N° 1 (controle) é responsável pelo acionamento e controle do motor de propulsão 1 e recebe uma mensagem CAN com prioridade 1 a cada 50 ms contendo a informação de *setpoint* requerido.

4.2 Análise de Resultados

A Tabela 3 sintetiza os parâmetros de configuração, adicionais aos do conjunto de mensagens CAN da Tabela 2, usados

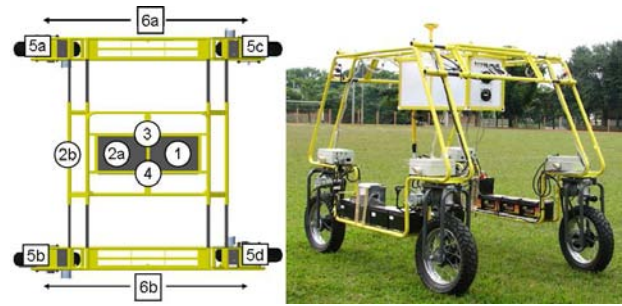


Figura 4: Estrutura do Sistema de Controle Distribuído com Rede CAN do Robô Móvel (Godoy, 2007)

1	Computador principal
2a	Computador do sistema de visão
2b	Câmera monocromática com filtro para luz visível
3	DGPS
4	Bússola Digital
5a,b,c,d	Conjunto composto por 2 controladores de motor (guiagem e propulsão), sensor potenciométrico e encoder
6a,b	Conjunto de Sonares (3 cada lado)

pelos modelos matemático (MT) e em Redes de Petri Coloridas (RP) nas simulações realizadas para o caso de estudo do robô móvel.

A idéia deste caso de estudo é demonstrar a aplicação dos modelos matemático e em Redes de Petri desenvolvidos. A partir das simulações realizadas será possível determinar métricas de desempenho como tempos de transmissão de mensagens, taxa de utilização da rede CAN e cumprimento de *deadlines* do sistema e analisar a operação e o comportamento temporal da rede CAN para o sistema distribuído do robô móvel. As simulações também serão utilizadas para analisar a viabilidade de utilização da rede CAN como solução de sistema de controle distribuído para o robô. Os primeiros resultados obtidos da simulação do sistema de controle distribuído com rede CAN do robô são mostrados na Tabela 4.

As Figuras 6 e 7 apresentam os resultados obtidos, para o parâmetro taxa de utilização da rede CAN do robô, através do modelo desenvolvido com Redes de Petri. Comparando-se os valores da Figura 6 com os da Tabela 4, é possível constatar que os valores obtidos por ambos os modelos são praticamente os mesmos. A Figura 7 apresenta também os resultados referentes à taxa de utilização da rede CAN para cada dispositivo conectado. Como era esperado, pode-se verificar que os dispositivos com maiores prioridades e taxas de amostragem mais rápidas possuem as maiores taxas de utilização

Tabela 2: Caracterização do Conjunto de Mensagens da Rede CAN do Robô Móvel

N°	Descrição das Mensagens	P	T (ms)
1	Controle Motor de Propulsão 1	1	50
2	Controle Motor de Propulsão 2	2	50
3	Controle Motor de Propulsão 3	3	50
4	Controle Motor de Propulsão 4	4	50
5	Controle Motor de Guiagem 1	5	50
6	Controle Motor de Guiagem 2	6	50
7	Controle Motor de Guiagem 3	7	50
8	Controle Motor de Guiagem 4	8	50
9	Velocidade Motor de Propulsão 1	9	100
10	Velocidade Motor de Propulsão 2	10	100
11	Velocidade Motor de Propulsão 3	11	100
12	Velocidade Motor de Propulsão 4	12	100
13	Posicionamento Motor de Guiagem 1	13	100
14	Posicionamento Motor de Guiagem 2	14	100
15	Posicionamento Motor de Guiagem 3	15	100
16	Posicionamento Motor de Guiagem 4	16	100
17	Sonares Lado Direito	17	150
18	Sonares Lado Esquerdo	18	150
19	Posicionamento GPS	19	150
20	Orientação Bússola Digital	20	150
21	Movimentação da Câmera	21	500
22	Controle do PC Industrial	22	150
23	Monitoração e Gerenciamento	23	250
24	Monitoração e Gerenciamento	24	250
25	Administração de Rede	25	250
26	Dispositivo genérico 1	26	250
27	Dispositivo genérico 2	27	250
28	Dispositivo genérico 3	28	500
29	Dispositivo genérico 4	29	500
30	Dispositivo genérico 5	30	500
31	Dispositivo genérico 6	31	1000
32	Dispositivo genérico 7	32	1000

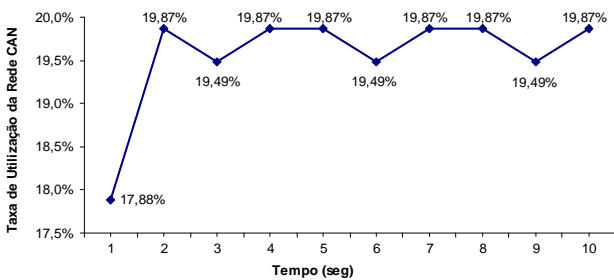


Figura 5: Resultados para Taxa de Utilização da Rede CAN obtidos com o Modelo com Redes de Petri

da rede CAN.

Tabela 3: Parâmetros adicionais à Tabela 2 usados pelos Modelos desenvolvidos nas Simulações

Parâmetro	Valor	Modelo
Formato do quadro de dados CAN	CAN 2.0B	MT/RP
Tamanho das mensagens de dados	64 bits	MT/RP
Quantidade de mensagens nos buffers	3	RP
Velocidade de transmissão de dados	250 kbits/s	MT/RP
Termo J_m – Equação (1)	0,1 ms	MT
Termo T – Equação (3)	54	MT
Termo O – Equação (3)	67	MT
Termo τ_{bit} – Equação (3)	4 μ s	MT
Termo n_{error} – Equação (6)	1	MT
Termo T_{error} – Equação (6)	100 ms	MT
Tempo de Simulação	10 s	RP

Tabela 4: Resultados obtidos com o Modelo Matemático

Parâmetros Analisados da Rede CAN	Resultados
Taxa de Utilização da Rede (%)	19,41
Taxa de Utilização das Mensagens (%)	7,77
Valor para Otimização da Utilização da Rede	5,05
Tempo de Resposta do Sistema (ms)	385,41

A simulação do sistema com o modelo com Redes de Petri permitiu também a análise do número de mensagens transmitidas por dispositivo conectado na rede, mostrado na Figura 8, e o monitoramento dos *buffers* de transmissão e recepção de mensagens dos dispositivos. Na simulação realizada, a quantidade máxima de mensagens armazenadas nos *buffers* dos dispositivos não ultrapassou o valor de uma (1) mensagem, inferior ao valor determinado (três) de acordo com as especificações do sistema. A partir desta informação pode-se concluir que não ocorrerá sobreposição de mensagens nos

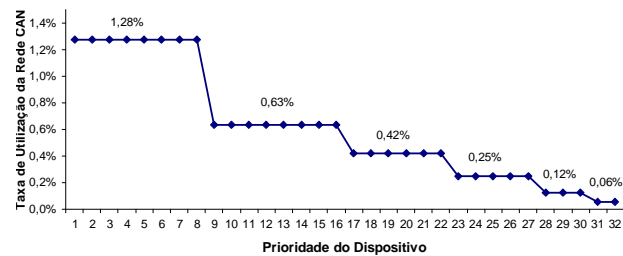


Figura 6: Utilização da Rede CAN por Dispositivo do Robô

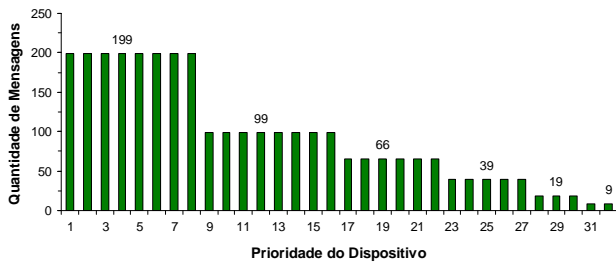


Figura 7: Mensagens Transmitidas pela Rede CAN

buffers durante a comunicação do sistema de controle.

Os resultados das análises de tempo de resposta (tempo de transmissão total das mensagens), apresentado na Tabela 4 e dos tempos de transmissão de cada mensagem da rede CAN, apresentado na Figura 9, demonstram que o sistema de controle distribuído proposto para o robô apresenta tempo de resposta satisfatório para cumprir o requisito temporal de cada mensagem. Tal requisito demanda que o tempo de transmissão de cada uma das mensagens tem que ser menor que o seu período de amostragem ou *deadline*, de forma a não acarretar em sobreposição e perda de mensagens transmitidas na rede CAN. Esse fato comprova a utilização dos modelos desenvolvidos para selecionar uma configuração de mensagens aceitável para o funcionamento do robô. O gráfico da Figura 9 permite também verificar a relação entre os tempos de espera e os tempos de transmissão de cada mensagem da rede CAN, possibilitando investigar quanto tempo cada mensagem teve de esperar para ser transmitida devido a configuração de prioridades utilizada e a política de acesso ao meio da rede CAN. Através das simulações realizadas foi possível verificar que o tempo de espera de uma mensagem está relacionado principalmente com a carga de dados da rede (quantidade de mensagens com maior prioridade que necessita ser transmitida antes da mensagem em análise no momento).

Diante dos resultados obtidos, é possível concluir que existe viabilidade para a aplicação da rede CAN analisada ao controle do robô móvel. Esse fato é confirmado pelos baixos valores referentes às taxas de utilização do barramento CAN mostrado na Tabela 4 e Figura 6. Esses dados demonstram, por exemplo, que o sistema apresenta capacidade disponível para futuras expansões e conexões de equipamentos, aumento da carga de dados (quantidade de bits enviados por mensagem) e aumento do tráfego (número de mensagens) de mensagens na rede CAN.

Os baixos valores obtidos para a taxa de utilização da rede CAN (até 20%) podem ser explicados pelo fato de o sistema proposto apresentar baixo tráfego (número) de mensagens e tais mensagens não terem períodos de amostragem críticos (tempo real - 5 a 10ms), como encontrado, por exemplo, em

aplicações automotivas. Sistemas com essas características podem apresentar taxas de utilização muitas vezes próximas de 100%, reafirmando a importância do desenvolvimento e da utilização deste tipo de modelagem e simulação em sistemas distribuídos com redes CAN.

Para demonstrar que os modelos desenvolvidos neste trabalho também podem ser aplicados na simulação de sistemas distribuídos com tempo crítico, foi proposto uma simulação para o caso de estudo do robô com alteração nos tempos de amostragem de algumas mensagens. Nessa nova simulação, levou-se a rede CAN a condições extremas de utilização de forma a efetivamente validar o modelo, através da seleção de períodos de amostragem críticos (5 e 10ms) para mensagens de controle do robô (mensagens de 1 a 8 na Tabela 2) e períodos de amostragem mais rápidos (20 e 50ms) para mensagens de monitoramento (mensagens de 9 a 16 na Tabela 2). Os resultados obtidos com a utilização dos modelos são mostrados na Figura 10 que apresenta uma comparação entre o *deadline* e o tempo de transmissão de cada mensagem simulada na rede CAN. A partir dos modelos desenvolvidos é possível determinar uma taxa de utilização de 98,77% (aproximadamente 100%) da capacidade de transmissão de mensagens da rede CAN. No entanto, através da análise do gráfico da Figura 10 é possível verificar que os dispositivos N° 7, 13, 15, 16, 25, 26, 27 e 30 perderiam seus *deadlines*, portanto o sistema não cumpriria com os requisitos temporais dessas mensagens. Essas mensagens não seriam transmitidas na rede CAN prejudicando a operação do robô (por exemplo, o motor de guiagem 3, que é controlado pelas mensagens N° 7 e 15, não seria controlado). Este fato acarretaria saturação da rede CAN e perda de mensagens. Consequentemente, nesta situação simulada, o sistema distribuído com rede CAN não seria viável para ser aplicado ao robô.

É importante reiterar que as simulações mostradas no caso de estudo do robô móvel foi proposta com o intuito de demonstrar a potencialidade de aplicação dos modelos desenvolvidos para análise de sistemas distribuídos com redes CAN. Ressalta-se que outros parâmetros de configuração, diferentes dos selecionados na simulação mostrada anteriormente, poderiam ser considerados para novas simulações, de acordo com as necessidades do projetista do sistema e com a área de aplicação (robótica, industrial, agrícola). Entre esses parâmetros estão: outras velocidades de transmissão (125k, 500kbits/s), novas configurações de prioridades para o conjunto de mensagens, outro formato do quadro de dados do protocolo CAN (CAN 2.0A), tamanhos diferentes para as mensagens de dados dos dispositivos e períodos de amostragem diferentes para as mensagens.

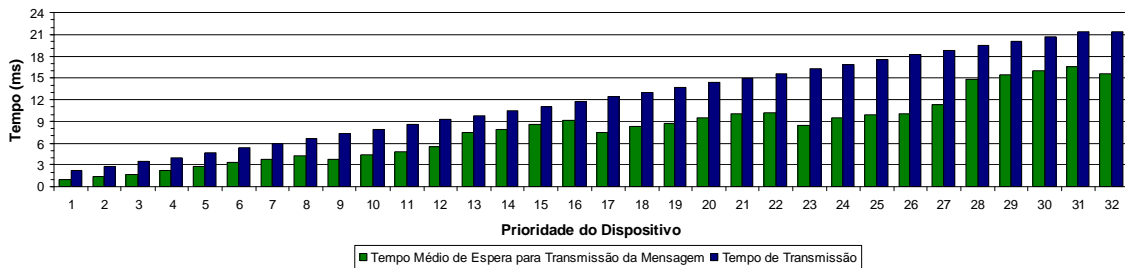


Figura 8: Comparação entre Tempo de Espera e Tempo de Transmissão para cada Mensagem da Rede CAN do Robô

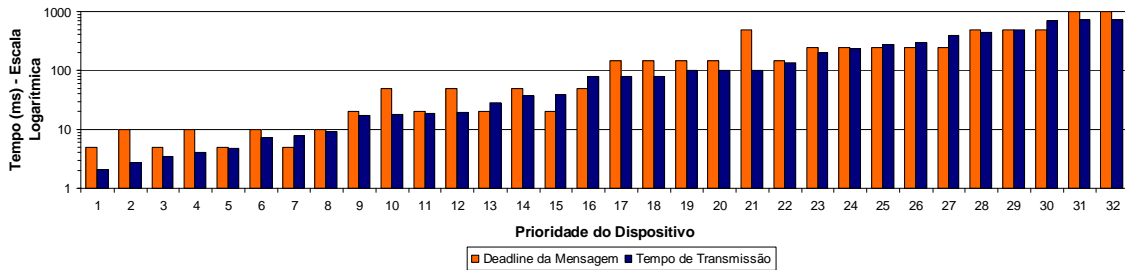


Figura 9: Comparação entre Deadline e Tempo de Transmissão para cada Mensagem da Rede CAN do Robô

5 CONCLUSÃO

Os objetivos deste trabalho foram atingidos, diante do desenvolvimento e da utilização dos modelos matemático e com Redes de Petri Coloridas propostos para a simulação de sistemas distribuídos com redes CAN. Os modelos desenvolvidos apresentaram características importantes como a incorporação ao modelo matemático de erros na transmissão de mensagens e tempo de retransmissão dessas mensagens e a possibilidade de determinar, com o modelo em Redes de Petri, a quantidade de mensagens nos *buffers* de transmissão e recepção e a relação entre o tempo de espera e o tempo total de transmissão de cada mensagem simulada na rede CAN. O desenvolvimento dos modelos permitiu o cálculo e a análise de diversos parâmetros relacionados ao desempenho e operação de sistemas distribuídos com redes CAN, como tempo de transmissão de mensagens, taxa de utilização da rede, atrasos de comunicação e verificação do cumprimento de *deadlines*, preenchendo as necessidades verificadas na literatura e correspondendo com as expectativas almejadas.

As técnicas de modelagem utilizadas (matemática e Redes de Petri Coloridas) mostraram-se adequadas e possibilitaram obter resultados como:

- flexibilidade de utilização (modelo matemático): permitindo ao usuário avaliar a influência de diversos parâmetros (formato da mensagem CAN 2B ou CAN 2B, diferentes velocidades de transmissão de dados da rede, diferentes quantidades de mensagens na rede e possibi-

lidade de definir as configurações de prioridade, período de amostragem e tamanho da mensagem para cada mensagem simulada na rede CAN) no desempenho e operação do sistema estudado;

- diminuição no tamanho do modelo (modelo com Redes de Petri Coloridas): a utilização do conceito de multimarcas ou multicores possibilitou ao modelo possuir uma única estrutura capaz de representar uma quantidade diversificada de dispositivos (sensores, atuadores, etc.), fornecendo grande flexibilidade de alteração e expansão para o modelo da rede CAN. O usuário não precisa alterar a estrutura do modelo para inserir ou retirar dispositivos na rede CAN ou para alterar suas configurações como prioridade de acesso a rede e período de amostragem;
- diminuição da complexidade do modelo (modelo com Redes de Petri Coloridas): facilitando a compreensão do funcionamento do sistema modelado;
- potencial de aplicação (ambos os modelos): permite sua utilização para a simulação e análise de qualquer tipo de aplicação de sistemas distribuídos com redes CAN (quadro de dados CAN 2B ou CAN 2B, velocidade de transmissão de dados entre poucos kbits/s até 1Mbits/s, qualquer quantidade de mensagens na rede e valores para período de amostragem das mensagens simuladas).

Um fato importante é que os modelos desenvolvidos podem ser utilizados separadamente para calcular parâmetros (des-

critos no final dos itens 3.2 e 3.3 para cada um dos modelos) relacionados ao desempenho do sistema em estudo. No entanto os resultados obtidos em uma eventual análise de um sistema distribuído com rede CAN são potencializados pela utilização conjunta dos dois modelos a partir do momento que uma maior quantidade de parâmetros (ou informações sobre o sistema) poderá ser calculada e analisada. A grande flexibilidade e potencial de aplicação dos modelos desenvolvidos foram comprovados em suas utilizações para o caso de estudo do robô móvel (permitindo até a simulação de condições extremas de operação da rede CAN). Essas utilizações simplificaram tarefas de análise relacionadas à rede CAN, gerando informações que auxiliaram o projetista no seu projeto e desenvolvimento. Os resultados das simulações realizadas também permitiram verificar a viabilidade da aplicação de uma rede CAN como solução de sistema distribuído para o robô e obter uma configuração de parâmetros otimizada para o funcionamento do sistema distribuído com rede CAN projetado.

Os próximos desenvolvimentos relacionados a esse trabalho focarão no sentido de tentar disponibilizar uma ferramenta de simulação que englobe ambos os modelos desenvolvidos, de forma a potencializar os resultados obtidos, e que possa ser utilizado sem a necessidade de ambientes de edição de Redes de Petri, de forma a facilitar sua utilização e contribuir com desenvolvedores de sistemas distribuídos com redes CAN.

REFERÊNCIAS

- Acton, K., M. Antolovic, N. Kalappa, J. Luntz, J.R. Moyne and D.M. Tilbury (2006). Practical Metrics for Evaluating Network System Performance, *Proc. of the UM-ERC/RMS Network Performance Workshop*, University of Michigan, Disponível em: <<http://erc.engin.umich.edu/publications/NPWPaper.pdf>>. Acesso em: Janeiro, 2008.
- Al-Hammouri, A., M.S. Branicky and V. Liberatore (2008). Co-simulation Tools for Networked Control Systems, Springer-Verlag, pp. 16-29, Berlin.
- Bago, M., S. Marijan and N. Peric (2007). Modeling Controller Area Network Communication, *Proc. of the 5th IEEE International Conference on Industrial Informatics*, 2007, Vol. 1, pp. 485-490.
- Bago, M., N. Peric and S. Marijan (2008). Modeling Bus Communication Protocols Using Timed Colored Petri Nets - The Controller Area Network Example, *Proc. of the 9th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, University of Aarhus, pp.103-121.
- Baillieul, J. & P.J. Antsaklis (2007). Control and Communication Challenges in Networked Real Time Systems, *Proc. of IEEE Technology of Networked Control Systems*, Vol. 95, No. 1, pp. 09-28.
- Bayilmis, C., I. Ertürk, C. Çeken and I. Özçelik (2006). Modeling Controller Area Networks Using Discrete Event Simulation Technique. *Proc. of the Physics Complex Computing-Networks*, Vol. 104, No. 2, pp. 353-358.
- Bosch (2006). CAN Specification Version 2.0. Disponível em: <<http://www.can.bosch.com>>. Acesso em: Julho.2006.
- Cavaliere, S., A. Distefano, L.L. Bello and O. Mirabella (1996). CAN assessment in time-critical cyclic applications through Petri Net model. *Proc. Of the 2nd IEEE International Conference on Industrial Electronics*, Vol. 2, pp. 922-927.
- Cervin, A., D. Henriksson, B. Lincoln, J. Eker and K.E. Arzen (2003). How does control timing affect performance? Analysis and simulation of timing using Jitterburg and Truetime. *IEEE Control System Magazine*, Vol. 23, pp. 16-30.
- Christensen, S., J.B. Jorgensen and L. Kristensen (1997). Design/CPN - A computer tool for Colored Petri Nets. *Proc. of the 3rd International Workshop on Tools and Algorithms for Construction and Analysis of Systems*, Vol. 1217, pp. 209-223.
- CIA. (2005). CAN in Automation. Disponível em: <<http://www.can-cia.org/>>. Acesso em: Julho. 2009.
- Davis, R.I.; A. Burns, R.J. Brill and J.J. Lukkien (2006). Controller Area Network (CAN) Schedulability Analysis: Refuted, Revisited and Revised. Technical Report, University of York. Disponível em: <<http://www.cs.york.ac.uk/ftplib/reports/2006/YCS/408/YCS-2006-408.pdf>>. Acesso em: Agosto. 2009.
- Girault, C. & R. Valk (2003). Petri nets for systems engineering: A guide for modeling, verification and application. Springer Verlag.
- Godoy, E.P. (2007). Desenvolvimento de uma Ferramenta de Análise de Desempenho de Redes CAN para Aplicações em Sistemas Agrícolas, Dissertação de Mestrado, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos – SP.
- Huo, Z., H. Fang and C. Ma (2004). Networked control system: state of the art. *Proc. of the 50th World Congress on Intelligent Control and Automation, WCICA 2004*, Vol. 2, pp. 1319- 1322, June.

- Kumar, M., A.K. Verma and A. Srividya (2009). Response-Time Modeling of Controller Area Network (CAN). *Proc. of the 10th International Conference on Distributed Computing and Networking Section: Reliability and Security*, Lecture Notes In Computer Science, Vol. 5408, pp. 163–174, Springer-Verlag.
- Lian, F.L., J.R. Moyne and D.M. Tilbury (2002). Network Design Consideration for Distributed Control Systems. *IEEE Transactions on Control Systems Technology*, Vol. 10(2), pp. 297-307.
- Lian, F.L.; J.R. Moyne and D.M. Tilbury (2001). Performance evaluation of control networks: Ethernet, Controlnet and Device Net. *IEEE Control Systems Magazine*, Vol. 21, pp. 66-83.
- Lopes, W.C. (2007). Análise de Desempenho do Protocolo CAN para Aplicação na Área Agrícola utilizando Rede de Petri Colorida, Dissertação de Mestrado, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos SP.
- Moyne, J.R. & D.M. Tilbury (2007). The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data. *Proc. of IEEE Technology of Networked Control Systems*, Vol. 95, No. 1, pp. 29-47.
- Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proc. of the IEEE*, Vol. 77, pp. 541–580.
- Nolte, T. (2007) Reducing pessimism in CAN response time analysis. Mälardalen University, Sweden. Disponível em: <<http://www.mrtc.mdh.se/publications/0384.pdf>>. Acesso em: Maio. 2009.
- Othman, H.F., Y.R Aji, F.T. Fakhreddin and A.R. Al-Ali (2006). Controller Area Networks: Evolution and Applications. *Proc. of the 2nd International Conference on Information and Communication Technologies*, Vol. 2, pp. 3088-3093.
- Portugal, P., A. Carvalho and F. Vasques (2005). A model based on a stochastic Petri Net approach for dependability evaluation of controller area networks. *Proc. of the 6th IFAC International Conference on Fieldbus Systems and their Applications*, Vol. 6, No. 1.
- Punnekkat, S., H. Hansson and C. Norstrom (2000). Response Time Analysis under Errors for CAN. *Proc. of the 6th IEEE Real-time Technology and Applications Symposium - RATS 2000*, pp. 258–265.
- Rodriguez, R.V., & R.M. Menendez (2007). Network-induced delay models for can-based networked control systems. *Proc. of the 7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems, Fieldbuses and Networks in Industrial and Embedded Systems*, Vol. 7, No.1.
- Tindell, K., A. Burns and A. Wellings (1995). Calculating Controller Area Network (CAN) Message Response Time. *Control Engineering Practice*, Vol. 3, No. 8, pp. 1163-1169.
- Torngren, M., D. Henriksson, K.E. Arzen, A. Cervin and Z. Hanzalek (2006). Tool supporting the co-design of control systems and their real-time implementation: current status and future directions. *Proc. of the 2006 IEEE International Symposium on Intelligent Control*, Munich, Germany, 4-6 October, pp. 1173-1180.
- Xiaodong, N. & Z. Yanjun (2002). Determining Message Delivery Delay of Controller Area Networks. *Proc. of the IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering - TENCON 2002*, Vol. 2, pp. 767–771.
- Yang, T.C. (2006). Networked control system: a brief survey. *IEEE Proceedings of Control Theory and Applications*, Vol. 153, No 4, July, pp. 403–412.