

**Universidade Federal de Santa Catarina  
Programa de Pós-Graduação em Engenharia de Produção**

**Max Mauro Dias Santos**

**Metodologias de Projeto para Sistemas de  
Controle via Redes**

**Tese de Doutorado**

Florianópolis  
2004

**Max Mauro Dias Santos**

**Metodologias de Projeto para Sistemas de  
Controle via Redes**

Tese de doutorado apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina como requisito para obtenção do grau de Doutor em Engenharia de Produção.

Orientador: Prof. Marcelo Ricardo Stemmer, Dr.-Ing. (UFSC – Brasil)

Co-orientador: Prof. Francisco Vasques de Carvalho, PhD (FEUP – Portugal)

Florianópolis  
1 de outubro de 2004

## **FICHA CATALOGRÁFICA**

Santos, Max Mauro Dias  
Metodologias de Projeto de Sistemas de Controle via  
Redes / Max Mauro Dias Santos, 2004. 126 p.  
Tese – Universidade Federal de Santa Catarina. Programa de Pós-  
Graduação em Engenharia de Produção

1. Sistema de controle via redes. 2. Sistemas de controle. 3. Sistemas de  
tempo real

Max Mauro Dias Santos  
**Metodologias de Projeto para Sistemas de Controle via Redes**

‘Esta Tese de Doutorado foi examinada e julgada adequada para obtenção do título de Doutor em Engenharia de Produção, área de concentração em Inteligência Aplicada e aprovada na sua forma final pelo Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina’.

Florianópolis, 1 de outubro de 2004

---

Prof. Marcelo Ricardo Stemmer, Dr.-Ing. - Orientador  
Departamento de Automação e Sistemas - UFSC

---

Prof. Edson Pacheco Paladini, Dr. - Coordenador  
Programa de Pós-graduação em Engenharia de Produção - UFSC

**Banca Examinadora:**

---

Prof. Francisco Vasques, PhD. – Co-orientador  
Faculdade de Engenharia da Universidade do Porto - FEUP

---

Prof. Marcelo Vieira Corrêa, Dr.  
Centro Universitário do Leste de Minas Gerais - UnilesteMG

---

Prof. Lau Cheuk Lung, Dr.  
Programa de Pós-Graduação em Informática Aplicada – PUCPR

---

Prof. Carlos Barros Montez, Dr.  
Departamento de Automação e Sistemas - UFSC

---

Prof. Rômulo Silva de Oliveira, Dr.  
Departamento de Automação e Sistemas - UFSC

---

Prof. Jorge Luis Roel Ortiz, Dr.  
Centro Tecnológico de Ciências da Terra e do Mar - UNIVALI

---

Prof. Frank Siqueira, PhD  
Departamento de Informática e Estatística - UFSC

A Meire, João Vitor e Maria Eduarda que são a prova de que a receita para o sucesso profissional é também o apoio de uma família sólida e estável.

Ao amigo e professor Marcelo Ricardo Stemmer, pelo incentivo e competência ; também pela amizade creditada a mim durante a realização deste trabalho, quando sacrificou interesses próprios para projetar me como um profissional centrado em estudos e pesquisas.

## **Agradecimentos**

A Deus, criador da vida...

Aos meus pais, José Pedroni dos Santos e D<sup>a</sup> Guilhermina Maria Dias dos Santos que estiveram sempre ao meu lado, incentivando-me. O orgulho que sempre demonstraram foi extremamente decisivo para que eu continuasse minha caminhada.

Ao excepcional co-orientador professor Francisco Vasques da Faculdade de Engenharia da Universidade do Porto (FEUP), que direcionou e orientou os fundamentos deste trabalho, mesmo estando do outro lado do oceano, expressava o calor da acolhida de um sábio mestre diante de seu humilde aprendiz. Ensinou-me, principalmente ética e caráter profissional.

Ao meu amigo José Francisco de Resende que sempre me deu lições de vida e à Profa. Dra. Maria Aparecida Fernandes Almeida que sempre esteve ao meu lado, inclusive nos momentos mais difíceis da realização deste trabalho.

Aos meus sogros, Sr. Nadir Dornelas e D<sup>a</sup> Margarida pela intenso apoio e torcida durante o desenvolvimento deste trabalho.

A todos aqueles que direta ou indiretamente contribuíram de forma decisiva na execução deste trabalho. Todos foram imprescindíveis para o meu sucesso.

## Resumo

O sistema de controle via redes (*Network Control Systems - NCS*), consiste em malhas de controle realimentado sob um sistema de comunicação, sendo uma área de pesquisa multidisciplinar que integra as redes de comunicação, sistemas de tempo real e sistemas de controle.

Neste trabalho é demonstrada a análise do impacto dos atrasos induzidos pelas redes de comunicação em *NCS*, como também a apresentação de metodologias de projetos com a simulação deste sistema, objetivando garantir desempenho e estabilidades aceitáveis em malhas de controle.

As redes de comunicação são estudadas através das propriedades temporais do escalonamento de mensagens em tempo real, análogas ao escalonamento de tarefas de tempo real numa *CPU (Central Processing Unit)*. São abordados três tipos de redes de comunicação em tempo real de padrão aberto e analisado o tempo de resposta das mensagens.

Analisou-se o impacto do instante de amostragem e atraso sobre o desempenho de controle em sistemas de controle por computador e a problemática de sistemas de controle via redes, através de diferentes arquiteturas de *NCS*; além de ter-se demonstrado um estudo de caso do impacto das redes sobre o desempenho de *NCS*. Apresentou-se também métodos de projeto *NCS* baseado no *WCRT (Worst Case Response Time)* ou tempo de resposta das instâncias de mensagens no pior caso. O método apresentado permite ao projetista especificar um controlador em função do instante de amostragem e atraso de entrada-saída.

Baseado nos conceitos de *Margem de Atraso* e *Margem de Jitter* para análise de estabilidade e projeto de sistemas de controle em tempo real, desenvolveu-se nesta tese uma inovação na aplicação de *NCS*.

## Abstract

Network Control Systems – *NCS* consist in feedback control system closed over a communication network and is a multidisciplinary area that integrates the communication networks, real-time systems and control systems.

In this work, are shown analysis of impact induced for delays in communication networks supporting *NCS*, as also the presentation of projects methodologies with system simulation, goal to guarantee acceptable performance and stabilities in control loops.

The communication networks are studied with its real-time scheduling properties of messages, with analogous properties of real-time scheduling for tasks in CPU (*Central Processing Unit*). Three types real-time communication networks are defined and used the worst case response time over messages are studied.

The analysis of impact at sampling instant and control delay over control performance in computer control systems and an extension to network control systems were realized, across the different *NCS* architectures, showing the case studies of network impact over *NCS* performance. Some *NCS* projects methodology based on WCRT (*Worst Case Response Time*) were presented and other method that allows designer to specify a controller in function of sampling period and input-output delay.

# Índice Analítico

<b>1</b>	<b>Introdução .....</b>	<b>1</b>
1.1	Motivação .....	1
1.2	O Projeto de Sistemas de Controle via Redes .....	3
1.3	Objetivos e Contribuições.....	4
1.4	Estrutura da Tese .....	5
1.5	Publicações.....	7
<b>2</b>	<b>Estado da Arte de Sistemas de Controle via Rede.....</b>	<b>9</b>
2.1	Introdução.....	9
2.2	Trabalhos Relacionados.....	10
2.3	Conclusão .....	12
<b>3</b>	<b>Redes de Comunicação em Tempo real.....</b>	<b>13</b>
3.1	Introdução.....	13
3.2	Tarefas e Mensagens de Tempo real.....	15
3.3	CAN (Controller Area Network) .....	20
3.4	Token Passing.....	26
3.5	TDMA (Time Division Medium Access) .....	28
3.6	Conclusão .....	32
<b>4</b>	<b>Sistemas de Controle via Redes (NCS).....</b>	<b>34</b>
4.1	Introdução.....	34
4.2	Sistemas de Controle por Computador .....	35
4.3	Sistema de Controle via Redes .....	43
4.4	Estudo de Caso .....	50
4.5	Conclusão .....	64
<b>5</b>	<b>Projeto de NCS Baseado no Tempo de Resposta de Mensagens.....</b>	<b>65</b>

5.1	Introdução.....	65
5.2	Método de Controle Tradicional com Atraso Constante .....	66
5.3	Projeto de NCS Considerando o WCRT de Mensagens .....	67
5.4	Exemplo 5.1.....	69
5.5	Conclusão .....	73
<b>6</b>	<b>Projeto de NCS Baseado na Margem de Jitter.....</b>	<b>74</b>
6.1	Introdução.....	74
6.2	Margem de Atraso e Margem de Jitter .....	75
6.3	Margem de Jitter Aplicado a Sistemas de Controle via Redes .....	79
6.4	Análise do Jitter em Mensagens.....	81
6.5	Exemplo de Projeto .....	84
6.6	Conclusão .....	89
<b>7</b>	<b>Conclusão e Trabalhos Futuros .....</b>	<b>90</b>
7.1	Conclusão Geral.....	90
7.2	Sugestões para Trabalhos Futuros .....	91
<b>8</b>	<b>Referências .....</b>	<b>92</b>

## Índice Figuras

Figura 3.1: Arquitetura em camadas de uma rede industrial clássica .....	14
Figura 3.2: Propriedades temporal de uma tarefa periódica de tempo real.....	17
Figura 3.3: Arquitetura de rede de comunicação de dados baseado no modelo em 4 camadas básico. ....	19
Figura 3.4: Propriedades temporal de uma mensagem periódica de tempo real.....	20
Figura 3.5: Processo de arbitragem no barramento CAN.....	22
Figura 3.6: Esquema de acesso ao meio do protocolo TDMA .....	29
Figura 4.1: (a) Diagrama em blocos de um sistema de controle por computador e (b) Diagrama temporal de um controlador digital. ....	36
Figura 4.2: (a) Algoritmo de controle clássico e (b) Diagrama temporal de um controlador digital clássico. ....	37
Figura 4.3: (a) Algoritmo de controle modificado e (b) Diagrama temporal de um controlador digital modificado.....	37
Figura 4.4: Desempenho em malha fechada do motor-dc para um sinal de referência $y_{ref}(t)$ .	38
Figura 4.5: (a) $h = 5$ ms e $L_{IO} = 1$ ms, (b) $h = 5$ ms e $L_{IO} = 3$ ms, (c) $h = 5$ ms e $L_{IO} = 5$ ms.	40
Figura 4.6: (a) $h = 10$ ms e $L_{IO} = 1$ ms, (b) $h = 10$ ms e $L_{IO} = 5$ ms, (c) $h = 10$ ms e $L_{IO} =$ 10 ms. ....	40
Figura 4.7: (a) $h = 15$ ms e $L_{IO} = 1$ ms, (b) $h = 15$ ms e $L_{IO} = 8$ ms, (c) $h = 15$ ms e $L_{IO} =$ 15 ms. ....	40
Figura 4.8: O gráfico da função $J(h_k, L_{IO})$ para o sistema de controle da Tabela 4.1.....	42
Figura 4.9: Arquiteturas de NCS com a representação de tarefas e mensagens .....	44
Figura 4.13: Desempenho em malha fechada do motor-dc para a resposta a um degrau ....	51
Figura 4.14: Arquitetura de rede com NCS, carga de mensagem periódica e carga extra....	51
Figura 4.15: Resposta do NCS-1 e escalonamento de mensagens sob uma rede CAN a 500 Kbps .....	54
Figura 4.16: Resposta do NCS-1 e escalonamento de mensagens sob uma rede CAN a 1 Mbps.....	55

Figura 4.17: Resposta do NCS-1 e escalonamento de mensagens sob uma rede Token-passing a 500 Kbps.....	58
Figura 4.18: Resposta do NCS-1 e escalonamento de mensagens sob uma rede Token-passing a 1 Mbps .....	59
Figura 4.19: Resposta do NCS-1 e escalonamento de mensagens sob uma rede TDMA a 500Kbps e 1Mbps, sem carga extra e apenas cinco nós computacionais de acesso ao meio .....	61
Figura 4.20: Resposta do NCS e escalonamento de mensagens sob uma rede TDMA a 500Kbps e 1Mbps, com carga extra e apenas cinco nós computacionais de acesso	62
Figura 4.21: Resposta do NCS e escalonamento de mensagens sob uma rede TDMA a 500Kbps e 1Mbps, com carga extra e seis nós computacionais de acesso ao meio .	63
Figura 5.1: Arquiteturas de <i>NCS</i> com implementação de controlador <i>LQG</i> .....	67
Figura 5.2: Configuração de três pêndulos invertidos na forma de <i>NCS</i> .....	70
Figura 5.3: Simulação no TrueTime: (a) desempenho de controle e (b) escalonamento de mensagens. ....	73
Figura 6.1: Sistema de controle por computador com uma planta contínua no tempo $P(S)$ , amostrador periódico $S_h$ , controlador discreto no tempo $K(Z)$ , segurador de ordem-zero e atraso variante no tempo ? .....	75
Figura 6.2: O atraso de entrada-saída pode ser dividido na forma de atraso constante ( $L$ ), e um jitter ( $J$ ). ....	76
Figura 6.3: (a) Arquitetura de um <i>NCS</i> , (b) Diagrama em blocos do <i>NCS</i> .....	80
Figura 6.5: O atraso de entrada-saída pode ser dividido na forma de atraso constante ( $L$ ) e um jitter ( $J$ ). ....	81
Figura 6.6: Resposta das malhas de controle contínuo no tempo. ....	86
Figura 6.7: Configuração de <i>NCS</i> 's sob uma rede CAN.....	87
Figura 6.8: Respostas dos <i>NCS</i> 's sob a rede CAN: a) primeira iteração, b) décima iteração.	89

## Índice Tabelas

Tabela 2.1: Parâmetros de uma tarefa periódica de tempo real.....	17
Tabela 2.2: Parâmetros de mensagem de comunicação de tempo real .....	20
Tabela 4.1. Parâmetros de projeto do controlador PD. ....	38
Tabela 4.2. Parâmetros de projeto do controlador PD.....	50
Tabela 4.3: Características dos fluxos de mensagens periódicas.....	52
Tabela 4.4: Características das mensagens sob a rede CAN .....	53
Tabela 4.5: Atraso de entrada-saída do NCS-1 a partir dos atrasos de execução tarefas e transmissão de mensagens .....	53
Tabela 4.6: Características das mensagens na rede Token-passing .....	56
Tabela 4.7: Atraso de entrada-saída do NCS-1 a partir dos atrasos de execução tarefas e transmissão de mensagens .....	57
Tabela 4.8: Características das mensagens na rede TDMA.....	60
Tabela 4.9: Atraso de entrada-saída do NCS a partir dos atrasos de tarefas e mensagens ...	61
Tabela 5.1. Cálculo do tempo de resposta na rede CAN.....	70
Tabela 6.1: Largura de banda e margem de fase das malhas de controle. ....	86
Tabela 6.2. Características dos fluxos de mensagens – Rede CAN (1Mbps).....	87
Tabela 6.3: Resultados do projeto integrado de NCS sob uma rede CAN: a) primeira iteração, b) décima iteração. ....	88

# 1 Introdução

## 1.1 Motivação

O sistema de controle por computador com dispositivos microprocessados é conseqüência de um crescente aumento da capacidade de processamento e redução dos custos e está presente em uma vasta área de aplicações em sistemas de nossa vida cotidiana. Como exemplos, temos a sua utilização em sistemas de automação industrial, automotivos, aeronaves, aeroespaciais, Internet, aparelhos de DVD, CD-ROM entre outros. As aplicações modernas geralmente requerem que um sistema computacional seja capaz de executar tarefas de controle em paralelo com outras tarefas, inclusive de atualização de telas e comunicação, evidenciando assim situações de concorrência aos recursos computacionais. Desta forma cria-se um problema de controle e escalonamento integrados (Cervin, 2003).

Através da grande disponibilidade e do baixo custo da atual tecnologia das redes de comunicação de dados, a comunidade de sistemas de controle observou uma tendência crescente de utilização destas na interconexão de sensores, atuadores e controladores em sistemas de controle realimentado que, conseqüentemente, contribui para a distribuição de funções e tarefas, garantindo o desempenho e estabilidade. Este tipo de implementação, onde as malhas de controle são fechadas, sob uma rede de comunicação, é habitualmente denominada de Sistemas de Controle via Redes (ou *NCS*, para *Networked Control Systems*). Como exemplo, pode se demonstrar que a utilização de *NCS* em aplicações automotivas contribui para a redução da utilização de cabos e conectores influenciando diretamente na diminuição do peso e volume do automóvel, bem como do tempo e do custo de produção.

Uma abordagem de projeto integrado de um *NCS* deve envolver as áreas de *sistemas de controle*, *sistemas de tempo real* e *redes de comunicação*, considerando que as redes de comunicação devem oferecer um serviço para transmissão de mensagens em tempo real.

Em torno do *NCS* se desenvolve uma nova área de pesquisa multidisciplinar, envolvendo as comunidades de *sistemas de controle*, *sistemas de tempo real* e *redes de comunicação* (Ärzén et al., (1999)). Tradicionalmente, as diversas comunidades realizam suas tarefas de forma independente, com a subutilização inadequada de recursos como largura de banda da rede e/ou tempo de computação da CPU, o que concorre para a perda de desempenho e instabilidade de um *NCS*. É comum verificar abordagens simplificadoras ao projeto de *NCS*, tais como:

- a idéia equivocada de que a rede de comunicação do tipo compartilhada oferece atrasos de transmissão constante, o que não se confirma numa implementação real;
- a visão simplificada do algoritmo de controle como um simples programa que necessita de um determinado tempo de computação a ser executado de forma lógica;
- a falsa crença de que as rede de comunicação com uma alta largura de banda garante atrasos de transmissão de mensagens limitados, entre os elementos (*nós computacionais* computacionais: sensor, controlador e atuador), para a aplicação de controle.

A *comunidade de sistemas controle* compreende um sistema computacional onde o controlador possa ser implementado de forma algorítmica e simples, e que as mensagens tenham um tempo de transmissão constante pela rede de comunicação. A *comunidade de sistemas de tempo real* considera um controlador como um pedaço de código que possui parâmetros fixos a serem respeitados em tempo de execução, tais como: período de ativação, tempo de computação e *deadline* (tempo máximo de resposta aceitável). A *comunidade de redes de comunicação de dados* considera apenas que as grandezas físicas devam ser transmitidas na forma de mensagens entre os elementos da rede de comunicação e uma alta taxa de transmissão seja suficiente.

Há aplicações de controle que são sensíveis a atrasos e variações de atrasos (*jitter*) entre o instante de medição e atuação na planta física. Um dos principais requisitos das aplicações de controle na forma de *NCS* com relação à rede de comunicação é que esta ofereça um serviço de comunicação em tempo real. Subjacente a este tipo de serviço deve estar a possibilidade de *suportar fluxos de mensagens periódicas*, de forma a transferir dados relacionados às variáveis de medição e atuação; a capacidade de garantir um *tempo*

*de resposta limitado superiormente* para transferência de mensagens entre os *nós computacionais* e a capacidade de garantir um *comportamento temporal previsível* na presença de carga de rede variável por causa do tráfego de mensagens não relacionado à aplicação de controle.

Existem diferentes redes de comunicação que concorrem para a utilização do suporte de *NCS*, devido ao fato de fornecerem serviços de comunicação de tempo real adequados. Com base nos paradigmas de acesso ao meio, podem-se classificar as redes como:

- a) acesso ao meio de forma aleatória;
- b) acesso ao meio com particionamento;
- c) acesso ao meio com passagem de permissão (revezamento).

As redes selecionadas e utilizadas ao longo desta tese, que satisfazem a classificação de acesso, conforme apresentado anteriormente, foram respectivamente: *CAN (Controller Area Network)* (ISO11898, 1993), *TDMA (Time Division Medium Access)* (Kopetz and Grunsteidl, 1994) e *Token-Passing* (nomeadamente implementado no Profibus (EN 50170, 1996)).

## **1.2 O Projeto de Sistemas de Controle via Redes**

Na área de *Escalonamento em Tempo Real e Controle Integrados* diversas propostas de projeto integrado foram apresentadas pela comunidade científica (ver um estudo detalhado em *Ärzén et al.*, (1999)). Destacam-se diversos trabalhos, valendo ressaltar as formas de controle e escalonamento, que foram formulados na forma de problema de *programação não linear*, cuja função objetivo é a de avaliar o desempenho das diversas malhas de controle, considerando como restrições às limitações dos recursos computacionais e temporais das tarefas de controle (*Seto et al.*, (1996), *Ryu et al.* (1997) and *Cervin* (2003)).

Para *NCS* o escalonamento de mensagens e tarefas é o principal fato para garantir comportamento temporal previsível de tarefas, tornando-se complexo pelo fato de as redes de comunicação serem sistemas não lineares e variantes no tempo. A formulação de um

problema de *programação não linear* para um *NCS* torna-se complexa, em consequência da natureza não linear do escalonamento e das possíveis arquiteturas que um *NCS* pode ter.

Um *NCS* do tipo *SISO* (*Single Input and Single Output*) ou *MIMO* (*Multiple Input and Multiple Output*) pode ter distintas arquiteturas e em consequência diferentes configurações de tarefas sob os nós computacionais e mensagens sob a rede de comunicação.

A utilização de sistemas operacionais (ou *kernel*: núcleo de um sistema operacional) de propósito geral não garantem um comportamento previsível sob a execução de tarefas e a mesma situação pode ser considerada para protocolos de comunicação. A previsibilidade na execução de tarefas e mensagens são requisitos fundamentais para que o atraso entre a medição e atuação em malhas de controle, possa ser conhecido a priori, pois existem malhas de controle sensíveis a *atrasos* e *variação de atrasos*, comprometendo o desempenho e estabilidade em um *NCS*.

### **1.3 Objetivos e Contribuições**

O objetivo desta tese é analisar o impacto da comunicação sob o desempenho e estabilidade de *NCS* e apresentar metodologias de projeto, considerando as propriedades temporais dos protocolos de acesso ao meio das redes de comunicação e as restrições temporais do *período de amostragem* e *atrasos de entrada-saída* para as malhas de controle.

As principais contribuições desta tese são apresentadas a seguir:

- Definição e análise das redes de comunicação em difusão que oferecem serviços de transmissão de mensagens em tempo real, sendo escolhidas para isto as redes CAN (*Controller Area Network*), TDMA (*Time Division Medium Access*) e Token-passing;
- Avaliação da sensibilidade das malhas de controle por computador sensível aos *atrasos de entrada-saída* e *período de amostragem* sob o desempenho e

estabilidade. Utilização dos métodos (*regra de thumb* e *critério de desempenho quadrático*) para seleção do *período de amostragem* e *atraso de entrada-saída*.

- Apresentação de diferentes arquiteturas de *NCS* na forma *SISO*, quando são fechadas sob redes de comunicação do tipo difusão e avaliar o impacto da comunicação sob o desempenho e estabilidade de *NCS* sob redes de comunicação em tempo real CAN, Token-passing e TDMA;
- Metodologia para o projeto de *NCS* com *período de amostragem* e *atraso de entrada-saída* pré-definidos, a partir do *tempo de resposta no pior caso (WCRT)* de mensagens.
- Projeto de *NCS* baseado no conceito de *margem de jitter*, através de um algoritmo interativo para seleção do *período de amostragem* com diversas malhas de controle fechadas sob uma rede de comunicação em tempo real.

## **1.4 Estrutura da Tese**

### **Capítulo 2: Redes de Comunicação em Tempo real**

Pretende-se apresentar uma introdução às propriedades temporais das tarefas de tempo real em sistemas computacionais e mensagens de tempo real em sistemas de comunicação, além de fazer análise das propriedades entre ambas. São apresentadas ainda três redes de comunicação do tipo difusão que oferecem serviços para transmissão de mensagens em tempo real, seguidas da sua aplicação em alguns protocolos de comunicação de padrão aberto. Com base na análise e projeto de um *NCS*, realiza-se um *survey* das propriedades temporais e tempo de resposta na transmissão de mensagens para as redes de comunicação CAN, Token-passing e TDMA.

### **Capítulo 3: Estado da Arte de Sistemas de Controle via Redes**

A partir do desenvolvimento e apresentação de trabalhos e pesquisa em conferências e periódicos, este capítulo apresenta um estudo do estado da arte para sistemas de controle via redes na forma cronológica.

#### **Capítulo 4: Sistema de Controle via Redes**

Este capítulo apresenta um estudo do impacto do *período de amostragem* e *atrasos de entrada-saída* sob o desempenho e estabilidade de sistemas de controle por computador. São apresentadas três arquiteturas de *NCS* para plantas físicas do tipo *SISO* com as propriedades temporais relacionadas às malhas de controle. O impacto da comunicação sobre o desempenho e estabilidade de um *NCS* é realizado considerando diferentes tipos de redes de comunicação em tempo real CAN, Token-passing e TDMA, com apresentação de exemplos ilustrativos.

#### **Capítulo 5: Projeto de NCS Baseado na Análise do Tempo de Resposta de Mensagens**

Apresenta-se o método de projeto do controlador flexível em sistemas de controle em tempo real, com uma extensão ao caso de *NCS* com *período de amostragem* e *atraso de entrada-saída* pré-definidos a partir do *tempo de resposta no pior caso (WCRT)* de mensagens. No esforço de auxiliar o entendimento desta abordagem, apresenta-se um exemplo ilustrativo baseado no *período de amostragem* e *atraso de entrada-saída* para o projeto de *NCS*.

#### **Capítulo 6: Aplicando Margem de Jitter no Projeto de NCS**

A aplicação do conceito de *margem de jitter* ao caso de *NCS*, onde o controlador (tarefa única que utiliza todos os recursos do computador de forma isolada) é implementado como uma tarefa em um nó computacional com um *kernel* de tempo real, o que dá margem ao surgimento de uma dúvida quanto à segurança do escalonamento de mensagens. Baseado no teorema de estabilidade em malhas de controle com atrasos de entrada-saída variantes no tempo e sob *margem de jitter* para tarefas de controle, a *margem de jitter* combinada com a teoria de escalonamento em tempo real para

transmissão de mensagens é aplicado ao caso de *NCS*, de forma a garantir o desempenho e estabilidade da planta física.

## **Capítulo 7: Conclusões e Trabalhos Futuros**

Nesta tese foram apresentados os requisitos de comunicação de um *NCS*, analisando as propriedades de comunicação das redes de tempo real CAN, TDMA e Token-passing, avaliando-se o impacto da comunicação sob o desempenho e estabilidade de *NCS* e apresentando-se duas metodologias de projetos para *NCS*. A partir dos trabalhos desenvolvidos durante esta tese, sugestões para trabalhos futuros foram apresentadas com o objetivo em explorar mais esta área de pesquisa emergente.

### **1.5 Publicações**

Este item apresenta as principais publicações realizadas pelo autor, durante os quatro anos de estudo do doutoramento.

#### **Periódicos**

Santos, M. M. D.; Vasques, F.; Stemmer, M. R. (2003): "Avaliação das Propriedades Temporais de Duas Redes de Controle: CAN e PROFIBUS." *Acta Scientiarum*. Universidade Estadual de Maringá (UEM), v. 25, nº 2, p. 193-201, 2003.

#### **Conferências**

Santos, M. M. D.; Vasques, F.; Stemmer, M. R. (2004): "Analysis of Timing Properties from Network Control Systems." In *Proceedings of 6th Brazilian Workshop on Real-Time Systems*. Gramado, Brazil.

Santos, M. M. D.; Vasques, F.; Stemmer, M. R. (2004): "Design of Network Control Systems – CAN and Token-Passing." In *Proceedings of VI IEEE International Conference on Industrial Applications*. Joinville, Brazil.

- Santos, M. M. D.; Vasques, F.; Stemmer, M. R. (2004): "O Impacto da Comunicação em Sistemas de Controle via Redes CAN." In *Proceedings of XV Brazilian Automatic Congress*. Gramado, Brazil.
- Santos, M. M. D.; Ataíde, F. H.; Vasques, F.; Stemmer, M. R. (2003): "Compensação de Atrasos em Sistemas de Controle via Redes - Rede CAN." In *Proceedings of IEEE Seminary in 1st Power Systems, Automation and Control*. Juiz de Fora, Brazil.
- Santos, M. M. D.; Vasques, F.; Stemmer, M. R. (2003): "Evaluation of the Timing Properties of Two Control Networks: CAN and PROFIBUS." In *Proceedings of IEEE International Symposium in Industrial Electronics*. Rio de Janeiro, Brazil.
- Santos, M. M. D.; Vasques, F.; Stemmer, M. R.; Ataíde, F. H. (2003): "O Impacto da Rede no Atraso de Controle em um Sistema de Controle via Redes." In *Proceedings do III Congresso Brasileiro de Computação*. Itajaí, Brazil.
- Santos, M. M. D.; Stemmer, M. R.; Vasques, F. (2002): "Análise de Escalonabilidade de Mensagens para uma Rede CAN Aplicada a uma Aeronave não Tripulada - O Helicóptero HELIX." In *Proceedings of IV Workshop of Real-Time Systems*. Búzios, Brazil.
- Santos, M. M. D.; Stemmer, M. R.; Vasques, F. (2002): "Schedulability Analysis of Messages in a CAN Network Applied to an Unmanned Airship." In *Proceedings of 28th IEEE Conference on Industrial Electronic Society*. Madrid, Espanha.
- Santos, M. M. D.; Stemmer, M. R.; Vasques, F. (2002): "Uma Rede CAN Aplicada ao Controle de uma Aeronave não Tripulada - O Helicóptero HELIX." In *Proceedings of XIV Brazilian Automatic Congress*. Rio Grande do Norte, Brazil.

# 2 Estado da Arte de Sistemas de Controle via Rede

## 2.1 Introdução

Um sistema de controle por computador com arquitetura centralizada consiste em três partes principais: o elemento sensor para aquisição de dados, o computador para execução do algoritmo de controle e o elemento atuador para a transmissão do sinal de controle à planta física. Os dispositivos de campo (sensor e atuador) são conectados ao sistema computacional através de um enlace de comunicação do tipo ponto-a-ponto.

Se uma malha de controle é fechada sob um computador, o algoritmo de controle opera de forma isolada e sem concorrência pelos recursos computacionais. O tempo de execução do algoritmo de controle pode ser facilmente determinado, visto que sempre todos os recursos computacionais estarão sempre disponíveis. Quando diversas malhas de controle são fechadas sob um computador, os recursos computacionais sofrem da concorrência pelos algoritmos de controle, os quais geram um problema de escalonamento em que os algoritmos de controle podem experimentar atrasos e variações de atrasos em sua execução.

Com a grande disponibilidade e o baixo custo da tecnologia de redes de comunicação de dados, a substituição do enlace de comunicação ponto-a-ponto por um enlace de comunicação serial com transmissão por difusão é motivada pela redução dos custos da fiação, manutenção e descentralização do controle.

A malha de controle de sistema de controle por computador é fechada sob uma rede de comunicação de dados, onde os elementos sensor, atuador e controlador trocam informações através da passagem de mensagens. A este tipo de implementação é habitualmente denominado por Sistemas de Controle via Redes (ou *NCS*, para “*Networked Control Systems*”).

## 2.2 Trabalhos Relacionados

Sistemas de Controle via Redes (*NCS*) constitui numa classe de sistemas de controle diferente dos tradicionais sistemas de controle por computador, porque a rede de comunicação do tipo compartilhada e transmissão na forma de difusão introduzem atrasos na transmissão de mensagens, o que diretamente induz atrasos de amostragem, atuação ou execução nas malhas de controle.

Existe uma classe de malhas de controle que são sensíveis aos atrasos e variações de atrasos, comprometendo assim o desempenho e estabilidade da planta física.

Os domínios para aplicações de *NCS* são: motor DC (Tipsuwan and Chow, 2001; Kim, Park and Kwon, 1998), automotivo (Boustany et al., 1992; Ozguner et al., 1992), aeronaves (Ray, 1987), robôs móveis (Wargui, Tadjine and Rachid, 1996; Tipsuwan and Chow, 2002), manipulador robótico (Tarn and Xi, 1998) e ensino a distância (Kondraske et al. 1993, Overstreet and Tzes, 1999).

As pesquisas sob análise e modelagem de *NCS* foram conduzidas baseadas no modelo de tempo contínuo e tempo discreto, sendo mais natural realizarem a análise de *NCS* do ponto de vista de tempo discreto, porque os sistemas físicos são amostrados periodicamente, algoritmos de controle são executados e atuados na forma digital após um curto atraso de tempo.

Diversos resultados de pesquisa mostraram que os pesquisadores assumiam uma rede de comunicação síncrona a taxas de amostragem dos sensores, controladores e atuadores como sendo o mesmo, o que não pode ser realizável numa implementação real. Na forma de ordem cronológica foram apresentadas as principais contribuições realizadas até o momento.

Halevi and Ray (1988) propuseram uma *metodologia baseada no estado aumentado do sistema* que inclui os sinais atrasados e derivada um modelo em malha fechada para os *NCS*. Em seguida Ray and Halevi (1988), sugeriram uma abordagem para melhorar o desempenho de *NCS* para correta seleção da fase dos sinais.

Luck and Ray (1990, 1994) apresentaram uma metodologia denominada por *metodologia da fila*, que usa um observador de estados para estimar os estados da planta

física e um preditor para computar o controle preditivo baseado sob a medição das saídas passadas.

Hong (1995) desenvolveu uma *metodologia de escalonamentos do intervalo de amostragem* para um NCS de forma que os atrasos na rede não afetem significativamente o desempenho e estabilidade do sistema de controle.

Nilsson (1998) apresentou uma *metodologia de controle estocástico ótimo* para controlar um NCS sob uma rede com atrasos aleatórios. A metodologia trata o efeito dos atrasos da rede aleatória em um NCS como um problema LQG (*Linear Quadratic Gaussian*).

Walsh, Beldiman, Ye and Bushnell (1999a, 199b) usaram a teoria da perturbação e não linear para formular os efeitos de atrasos da rede em um NCS de um sistema de tempo contínuo sob a consideração de que não existe observação do ruído. Denominada como a *metodologia da perturbação*.

Goktas (2000) apresentou uma metodologia de projeto do controlador via redes no domínio da frequência, usando a teoria de controle robusto denominada por *metodologia de controle robusto*. A principal vantagem desta abordagem é que esta não requer informações a priori sob as distribuições de probabilidade dos atrasos de rede.

Tarn and Xi (1998) introduziram uma metodologia baseada em eventos para sistemas de controle via redes de um manipulador robótico sob a Internet. Ao invés de utilizar informações do tempo, a metodologia usa um sistema de movimentação como referência do sistema. Denominada como a *metodologia baseada em eventos*.

Almutairi et al. (2001) propuseram uma *metodologia de modulação por lógica fuzzy* para um NCS com uma planta física linear e um controlador PI modulado para compensar atrasos de rede baseado sob a lógica fuzzy (Zadeh, 1973).

Tipsuwan and Chow (2001) propuseram uma *metodologia de adaptação do controle no usuário final* para adaptar os parâmetros do controlador, ou seja, ajuste dos ganhos do controlador com respeito ao tráfego da rede de comunicação ou qualidade de serviço (QoS) da rede. Uma análise mais detalhada destas metodologias pode ser encontrada em (Tipsuwan and Chow, 2003).

## 2.3 Conclusão

Os trabalhos relacionados a sistemas de controle via redes apresentaram propostas e metodologias de análise e projeto de *NCS* sem levar em consideração as propriedades dinâmicas das redes de comunicação do tipo difusão e a sensibilidade dos atrasos e variação de atrasos na amostragem e processamento de entrada-saída das plantas físicas para o desempenho e estabilidade.

As redes de comunicação podem introduzir atrasos e variação de atrasos na transmissão de mensagens e, conseqüentemente, atrasos e variação de atrasos na amostragem e processamento de entrada-saída das plantas físicas.

A previsibilidade de uma rede de comunicação que ofereça serviços de transmissão de mensagens em tempo real é um requisito fundamental para que o tempo de resposta para um fluxo de mensagens seja determinado.

Nas metodologias apresentadas anteriormente, os pesquisadores preocupavam-se com o escalonamento de mensagens sob a rede e seleção do intervalo de amostragem de forma a garantir o desempenho e estabilidade do *NCS*. Consideravam ainda que a rede de comunicação possuía uma taxa de transmissão suficiente para a transmissão de mensagens, sendo o atraso entre a medição e atuação (atraso de entrada-saída) considerado desprezível se comparado ao *período de amostragem*.

Como existem plantas físicas sensíveis a atrasos e a variação de atrasos de entrada-saída e analisando que as abordagens de projeto para *NCS* sempre desconsideravam este fato, esta tese aborda o projeto de *NCS*, considerando o escalonamento de mensagens, período de amostragem e atraso de entrada-saída.

Esta tese apresenta metodologias de análise e projeto de *NCS* que consideram as propriedades dinâmicas da rede de comunicação e temporal das malhas de controle.

# 3 Redes de Comunicação em Tempo real

## 3.1 Introdução

A indústria ao nível de chão de fábrica está, cada vez mais, utilizando a tecnologia de redes de comunicação de dados (redes *Fieldbus*: barramento de campo) para interconectar dispositivos como sensores, atuadores e controladores. Isto faz com que as aplicações industriais apresentem uma arquitetura distribuída, perante a tradicional arquitetura centralizada (ponto-a-ponto de 4-20mA e psi), utilizada por décadas em ambiente industrial (Tovar, 1999).

As vantagens técnicas e econômicas da tecnologia *distribuída* são extremamente consideráveis quando se compara com a tecnologia *centralizada* (Day and Zimmermann, 1993), além do fato de existir a motivação de um conjunto de empresas com objetivo de padronização. Em determinadas aplicações às redes *Fieldbus* podem oferecer redução no peso e volume de cabeamento, facilidade de manutenção, diagnóstico computacional, motivados também pelo baixo custo das tecnologias de microprocessadores e microcontroladores, gerenciamento de dispositivos entre outros fatores, aumentando a segurança e confiabilidade, tornando-as cada vez mais atraentes.

Em certas aplicações industriais os requisitos de tempo consistem em um dos principais fatores de projeto, exigindo assim que a rede *Fieldbus* ofereça garantia de serviços de transmissão de dados em tempo real. Este fato é importante, visto que as aplicações convivem em um ambiente totalmente concorrente e com imprevisibilidade de carga, fazendo-se necessário uma política adequada de escalonamento de tarefas e mensagens em tempo real.

Uma rede de comunicação de tempo real, além de oferecer serviços básicos de transmissão de mensagens para as aplicações, deve ainda garantir latências bem definidas na transmissão de mensagens de forma que o paradigma de tempo real possa ser aplicado em projeto com o intuito de otimização de recursos computacionais e compromisso temporal das aplicações.

Em um ambiente industrial, está se tornando cada vez mais difundido o convívio entre diferentes aplicações com múltiplos requisitos (CPU, rede e outros recursos). De acordo com os requisitos das aplicações de automação industrial, as redes de comunicação podem ser representadas por uma arquitetura em camadas reduzidas, baseada em uma simplificação do modelo *RM-OSI (Reference Model – Open Systems Interconnection)*, conforme Fig. 3.1.



Figura 3.1: Arquitetura em camadas de uma rede industrial clássica

A *Camada de Aplicação* é responsável pelas aplicações industriais utilizadas, onde se implementa mecanismos de comunicação como produtor-consumidor, cliente-servidor, mestre-escravo entre outros. A camada *LLC (Logical Link Control)* possui mecanismo de controle de erro e fluxo, mecanismos de reconhecimento e de retransmissão de mensagens, aliviando assim a carga de trabalho da camada de aplicação. A camada *MAC (Medium Access Control)* é responsável pela política de acesso ao meio de comunicação, podendo ser determinístico ou não-determinístico. A *Camada Física (PHY)* é responsável pelas especificações elétrica, mecânica, sinalização e outros com relação à transmissão física de um sinal.

A arquitetura em camadas da Fig. 3.1 pode garantir a transmissão de mensagens em tempo real, através de mecanismos de escalonamento de mensagens e acesso ao meio determinista das subcamadas *LLC* e *MAC*.

Para isto existem diversas propostas de arquiteturas de redes de comunicação de tempo real que oferecem atrasos de transmissão de mensagens, mesmo com a rede sob condições de carga adversas.

Diversos mecanismos foram propostos e podem fazer com que as redes ofereçam serviços de transmissão de mensagens em tempo real, porém direcionando ao nosso foco de pesquisa, vai nos ater apenas a redes de comunicação de dados que oferecem mecanismos de acesso ao meio em tempo real, com ferramentas para a análise de previsibilidade e de tempo de respostas de mensagens no pior caso.

Neste trabalho estudam-se as redes de comunicação de dados que, além de oferecer garantia de transmissão de mensagens em tempo real, apresentam-se mecanismos de análise de escalonabilidade de mensagens cientificamente bem consolidadas. As redes analisadas são: CAN, TDMA e Token-Passing.

### **3.2 Tarefas e Mensagens de Tempo real**

Uma rede de comunicação de dados do tipo difusão é semelhante a um sistema de computação, onde tarefas concorrem pela sua execução em uma CPU.

O conceito da teoria de escalonamento de tarefas em tempo real pode ser estendido a sistemas de comunicação com escalonamento de mensagens sob uma rede do tipo difusão, visto que apresentam as mesmas características básicas (Zhang, 2001).

O escalonamento de CPU e de rede envolve as mesmas características, onde um recurso compartilhado é requisitado através de invocações freqüente de tarefas para a CPU e mensagens para a rede.

Entretanto, em sistemas de computação em tempo real o recurso compartilhado é o processador com a execução de *tarefas de tempo real* com propriedades temporais, o mesmo podendo ser aplicado a um sistema de comunicação em tempo real onde recurso compartilhado é a rede de comunicação com a transmissão de *mensagens de tempo real* com propriedades temporais análogas ao caso de tarefas.

#### **3.2.1 Escalonamento de Tarefas em Tempo real**

A teoria de escalonamento em sistemas de tempo real baseia-se na teoria de escalonamento clássico (Liu and Layland, 1973), onde para um dado conjunto de tarefas executadas concorrentemente com um algoritmo de escalonamento é definido para que o

sistema computacional não apenas garanta um resultado lógico correto como também seja executado dentro de um prazo limite bem definido (*deadline*). Escalonamento de tempo real envolve a alocação de recursos computacionais e de tempo de forma que os *deadlines* do sistema sejam satisfeitos (Liu, 2000).

Em função do não cumprimento de *deadline* na execução de tarefas, os sistemas de tempo real podem ser classificados em:

- *Sistemas de tempo real críticos (Hard real-time systems)* – são sistemas em que as tarefas devem ser executadas antes de um *deadline*, e a falta deste pode levar a consequências desastrosas, como um sistema de controle de direção automotiva, freio ABS (*Antilock Brake Systems*) ou usina nuclear.
- *Sistemas de tempo real não críticos (Soft real-time systems)* – são sistemas em que as tarefas executantes possuem um *deadline* associado que pode ou não ser respeitado, podendo o sistema manter sua operação com um desempenho degradado em caso de perda do *deadline*, mas sem sofrer nem causar danos ao sistema.

Em sistemas de tempo real com relação a utilidade do resultado de uma tarefa para a aplicação, pode-se definir o seguinte:

- *Tarefa não-crítica (“soft task”)*, quando apesar da ultrapassagem de uma meta temporal, ainda existe algum valor associado ao serviço prestado fora de prazo;
- *Tarefa firme (“firm task”)*, quando a ultrapassagem de uma meta temporal não tem consequências desastrosas, mas o valor do serviço prestado pode ser considerado nulo;
- *Tarefa crítica (“hard task”)*, quando a ultrapassagem de uma meta temporal pode ter consequências desastrosas, ou seja, o valor do serviço prestado fora de prazo é muito menor que o valor negativo do prejuízo causado;

Em sistemas de controle de tempo real, o algoritmo de controle é caracterizado por um algoritmo computacional ativado periodicamente com um tempo de execução definido. A Fig. 3.2 apresenta as propriedades temporais de uma tarefa com ativação periódica.

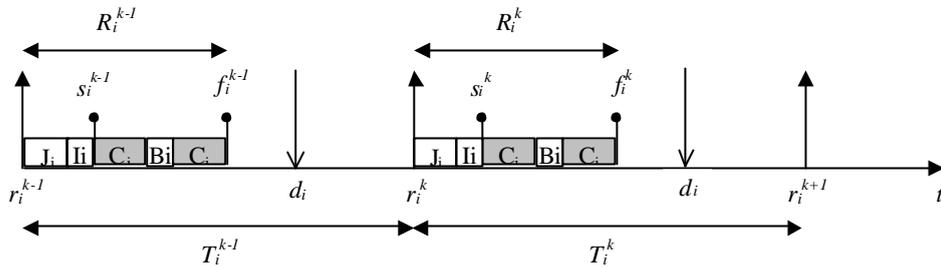


Figura 3.2: Propriedades temporais de uma tarefa periódica de tempo real

Em geral, uma tarefa de tempo real  $t_i$  é caracterizada por diversos parâmetros que influenciam no seu tempo de resposta, descritos na Tabela 3.1.

Tabela 3.1: Parâmetros de uma tarefa periódica de tempo real

Parâmetro	Descrição
$C_i$	Tempo de computação de uma tarefa no pior caso
$T_i$	Período de execução da tarefa
$r_i$	Instante que uma tarefa torna-se pronta para ser executar
$s_i$	Instante em que a tarefa inicia sua execução
$f_i$	Instante em que a tarefa finaliza sua execução
$J_i$	Tempo gasto entre a liberação até o início da execução no pior caso
$I_i$	Interferência sofrida pela preempção de outras tarefas no pior caso
$B_i$	Bloqueio máximo sofrido devido a execução de tarefas de menor prioridade
$R_i$	Tempo de resposta da tarefa no pior caso
$d_i$	Deadline da tarefa

De acordo com o diagrama temporal de uma tarefa executada em função de um algoritmo de escalonamento em tempo real, pode-se verificar que entre o instante de liberação e finalização, ela poderá sofrer diversas interferências, que a princípio podem ser previsíveis.

A mesma denominação pode ser aplicada aos sistemas de comunicação de tempo real, exceto a preempção de quadros (*frames*), visto que se um nó computacional iniciar a transmissão de um quadro, ele próprio deverá concluí-la.

### 3.2.2 Escalonamento de Mensagens em Tempo real

O escalonamento de mensagens de tempo real em redes de comunicação baseia-se na teoria de escalonamento de tarefas em tempo real e sem preempção, para um dado

conjunto de nó computacional que deseja transmitir suas mensagens, concorrendo assim pela posse de um canal de comunicação.

Para que a transmissão de uma mensagem seja realizada dentro de um prazo limite bem definido (*deadline*), é necessário um algoritmo de escalonamento de mensagens sob os nós computacionais e um protocolo de acesso ao meio determinista.

Relativamente à falta de *deadline* na transmissão de mensagens, os sistemas de comunicação de tempo real podem ser classificados quanto:

- *Sistemas de Comunicação de Tempo Real Críticos (Hard real-time communication systems)* – são sistemas de comunicação onde as mensagens devem ser transmitidas antes de um *deadline*, onde a falta deste pode levar a consequências desastrosas, como em sistemas automotivos *X-by-Wire*.
- *Sistemas de Comunicação de Tempo Real Não Críticos (Soft real-time communication systems)* – são sistemas de comunicação onde as mensagens podem até não cumprir o seu *deadline*, podendo o sistema ainda manter sua operação com uma desempenho degradada, como em sistemas distribuídos multimídia.

Para fornecer um requisito de transmissão de tempo real, a camada física e de aplicação obedecem às mesmas funcionalidades do modelo *RM-OSI*. Porém o protocolo de acesso ao meio da subcamada *MAC* deverá ter um caráter determinista e a subcamada *LLC* deverá apresentar um mecanismo de escalonamento de mensagens local também determinista, ou seja, a arquitetura em si das subcamadas não se mostra modificada, o que ocorre com as propriedades dos protocolos das subcamadas de enlace (*LLC*) e de acesso ao meio (*MAC*).

Esses dois requisitos importantes em uma rede de comunicação de tempo real são suficientes para que a transmissão de uma mensagem seja previsível. Com este propósito em relação aos protocolos de acesso ao meio (*MAC*), as redes de comunicação dividem-se em duas classes: as deterministas e não deterministas.

Dentre as redes não deterministas, podemos citar a Ethernet, padrão que utiliza o protocolo de acesso ao meio CSMA/CD, e entre as deterministas temos as redes PROFIBUS, CAN, TDMA, entre outras.

Um protocolo MAC que nos garante determinar o tempo de transmissão bem definido é um dos critérios mais importantes para que seja possível determinar o atraso de controle em cada ciclo.

A arquitetura de uma rede de comunicação de tempo real deverá obedecer ao seguinte modelo mostrado na Fig. 3.3.

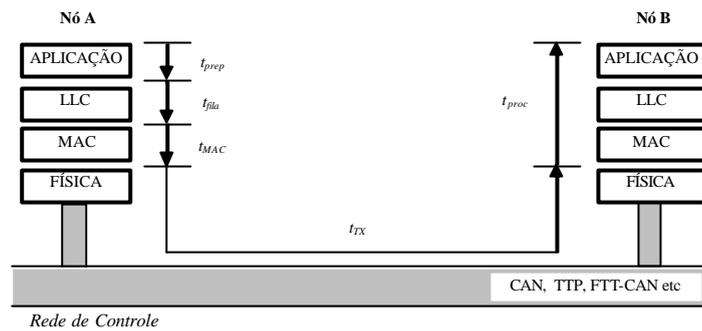


Figura 3.3: Arquitetura de rede de comunicação de dados baseado no modelo em 4 camadas básico.

A latência de transmissão de uma mensagem  $M_m$ , do *Nó computacional A* para o *Nó computacional B* consiste na soma de latências de preempção ( $t_{prep}$ ), fila de mensagens que desejam ser transmitidas em um nó computacional específico ( $t_{fila}$ ) e MAC no emissor ( $t_{MAC}$ ) é o tempo relativo ao escalonamento das mensagens, quando várias mensagens pretendem acessar simultaneamente ao meio de comunicação ( $t_{TX}$ ) e latência de processamento no receptor ( $t_{proc}$ ). Todas estas unidades são dadas em segundos.

Na Eq. (3.1) apresenta-se o atraso total para a transmissão de uma instância  $k$  da mensagem  $m$  sob, uma rede de comunicação.

$$t_m^k = t_{prop,m}^k + t_{fila,m}^k + t_{MAC,m}^k + t_{TX,m}^k + t_{proc,m}^k \quad (3.1)$$

Para o caso de não haver concorrência de mensagens na fila de transmissão e acesso ao meio de comunicação deve-se considerar  $t_{fila}^k$  e  $t_{MAC}^k$  iguais a zero.

Um fluxo de mensagens periódicas de tempo real  $M_m$  pode ser caracterizada pelo comportamento temporal apresentado na Fig. 3.4.

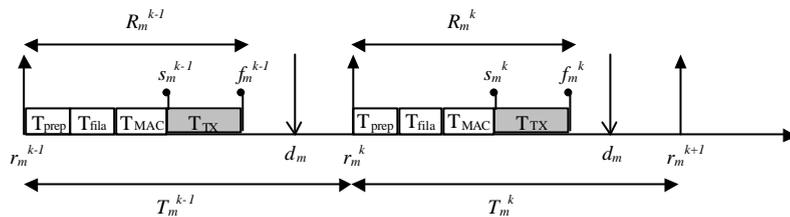


Figura 3.4: Propriedades temporal de uma mensagem periódica de tempo real

Em geral um fluxo de mensagens de tempo real  $M_i$  pode ser caracterizada por diversos parâmetros que influenciam no seu tempo de resposta, descrito na Tabela 3.2.

Tabela 3.2: Parâmetros de mensagem de comunicação de tempo real

Parâmetro	Descrição
$C_m$	Tempo de transmissão de uma mensagem
$T_m$	Período de transmissão da mensagem
$t_{prep}$	Tempo de preparação de mensagem
$t_{fila}$	Atraso na fila de transmissão de mensagem
$t_{MAC}$	Tempo que o nó computacional aguarda para ter posse do meio
$t_{TX}$	Tempo de transmissão da mensagem
$s_m$	Instante em que uma mensagem inicia sua transmissão
$f_m$	Instante em que uma mensagem finaliza sua execução
$R_m$	Tempo de resposta da mensagem
$d_m$	Deadline de uma mensagem

A seguir a problemática de escalonamento de mensagens em tempo real, com análises de escalabilidade será tratada, em função das propriedades temporais de mensagens em redes de comunicação e do paradigma de tempo real.

As redes escolhidas para a discussão possuem mecanismos de onde derivam as condições para que o projetista possa determinar tempo de resposta para mensagens em redes de comunicação em tempo real.

### 3.3 CAN (Controller Area Network)

A rede CAN (Controller Area Network) (ISO 11898, 1993) foi originalmente projetada para ser utilizada em veículos automotivos para resolver os problemas de cablagem que aumentavam com o crescimento do uso de componentes baseados em microprocessadores.

A rede CAN constitui-se num conjunto de especificações ao nível de camada física e controle de acesso ao meio, baseado modelo de referencia OSI (Day and Zimmermann, 1983), tornando-se necessária a implementação de camadas superiores com serviços adicionais para as aplicações veiculares e industriais. De acordo com os requisitos das aplicações automotivas, diversas arquiteturas distribuídas baseado na rede CAN foram apresentadas, dentre as principais podemos citar: VOLCANO (Volcano, 1996), CANKingdom (CANKingdom, 1996) e J1939 (SAE J1939, 2000).

A indústria notou que a rede CAN poderia oferecer os serviços básicos de que as aplicações industriais necessitavam, contanto que camadas adicionais fossem inseridas com serviços específicos. Com o objetivo de padronização dos sistemas Fieldbus, a IEC ( ) e CENELEC ( ) definiram um conjunto de padrões para diferentes tecnologias Fieldbus, gerando respectivamente as normas IS 62026-X e EN 50325 ( ) e ( ) que compreendem diferentes soluções Fieldbus baseadas na tecnologia CAN.

Dentre estas, podemos citar os padrões tais como DeviceNet ( ), SDS ( ) e CANOpen ( ), adequadas para o suporte de Sistemas de Controle via Redes (NCS) de pequena escala em automação industrial.

As variantes da rede CAN para automação industrial possuem um considerável *overhead* relativo às camadas superiores, dificultando o cálculo de tempo de transmissão de mensagens, irrelevante para as aplicações industriais de propósito gerais.

O protocolo CAN implementa um barramento baseado em prioridade, com um protocolo MAC do tipo CSMA/NBA (*Carrier Sense Multiple Access with Non-Destructive Bitwise Arbitration*) ou *Forcing-header*. Neste protocolo, qualquer estação pode acessar o barramento para transmitir uma mensagem, quando ele estiver ocioso. Entretanto, contrariamente às redes com protocolo CSMA, como Ethernet, a resolução da colisão não é destrutiva, no sentido de que sempre uma das mensagens sendo transmitida, será bem sucedida.

Quatro tipos de mensagens podem ser transferidas em uma rede CAN. Duas são usados durante a operação normal da rede CAN: a mensagem *Data-Frame*, que é usada para transferir dados de uma estação para outra e a mensagem *Remote-Frame*, que é usada para requisitar dados de uma estação distante. As outras duas mensagens são usadas para

sinalizar um estado anormal da rede CAN: a mensagem *Error-Frame* sinaliza a existência de um estado de erro e a mensagem *Overload-Frame* sinaliza que uma estação particular não está pronta para transmitir dados.

Os sinais do barramento podem ter dois estados diferentes: bit recessivo e bit dominante (que sempre sobrescreve o bit recessivo). O mecanismo de resolução da colisão funciona como se descreve a seguir: quando o barramento torna-se ocioso, cada estação com mensagem pendente iniciará sua transmissão. Durante a transmissão do campo identificador, se uma estação ao transmitir um bit recessivo e em seguida ler um bit dominante, significa que ocorreu uma colisão com uma mensagem de maior prioridade e, conseqüentemente, esta estação aborta a transmissão da mensagem. A mensagem de maior prioridade sendo transmitida irá prosseguir sem perceber qualquer colisão e, assim, será transmitida com sucesso. Ao se transmitir a mensagem de maior prioridade, ela irá prosseguir sem perceber qualquer colisão é a que tiver o maior valor de bits dominantes no campo identificador. Isto equivale à execução de uma operação lógica AND entre os bits do campo identificador enviados pelas estações concorrentes. Obviamente, cada fluxo de mensagem deverá ser unicamente identificado. A estação que perder a fase de arbitragem terá nova oportunidade de transmissão da sua mensagem no próximo ciclo. A Fig. 3.5 apresenta de forma ilustrativa o processo de arbitragem de um barramento CAN.

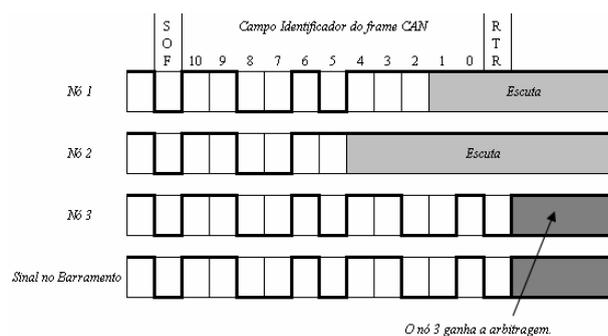


Figura 3.5: Processo de arbitragem no barramento CAN

### 3.3.1 Modelo de Mensagens

Assumindo que uma rede CAN possua  $r$  estações e  $n$  fluxos de mensagens periódicas, sendo cada fluxo definido por:

$$S_m = (C_m, T_m, D_m) \quad (3.2)$$

Um fluxo de mensagem é uma seqüência temporal de mensagens relativas, por exemplo, à leitura remota de uma variável de processo específico. O termo  $S_m$  é caracterizado como um identificador para um fluxo de mensagem;  $C_m$  é o tempo de transmissão da mensagem e  $T_m$  é a periodicidade de transmissão considerada como o intervalo de tempo mínimo entre duas chegadas consecutivas de requisições para a fila de saída do fluxo  $S_m$ . Finalmente,  $D_m$  é a meta temporal relativa de uma mensagem (*deadline*) do fluxo  $S_m$ , que consiste no intervalo de tempo máximo admissível entre o instante em que a requisição da mensagem é colocada na fila de saída e o instante em que cada mensagem é completamente transmitida.

### 3.3.2 Análise de Escalonabilidade de Mensagens

Liu and Layland, 1973 mostraram que o algoritmo RM (*“Rate Monotonic”*) é um algoritmo matematicamente demonstrado como o melhor entre todos os algoritmos de prioridade fixa para tarefas independentes e com período igual ao *deadline*, no sentido de que nenhum outro algoritmo de prioridade fixa pode escalonar um conjunto de tarefas que não possa ser escalonado por RM.

O algoritmo de escalonamento de tarefas RM pode ser estendido ao caso de escalonamento de mensagens, levando-se em conta que a análise sob um conjunto de tarefas sobre uma CPU pode ser estendida ao caso de um conjunto de mensagens sob a rede. Considera-se, ainda, que num ambiente de rede o escalonamento de mensagens é não-preemptivo ao nível de frame.

Um teste de escalonabilidade foi proposto em por Sha and Lechosky, 1987 e descrito a seguir. Para um conjunto de mensagens periódicas e não-preemptivas, com prioridade decrescente são escalonáveis se para todo  $m = 1, \dots, N$ .

$$\frac{C_m}{T_m} + \frac{C_2}{T_2} + \dots + \frac{C_i}{T_i} + \frac{\bar{b}_{l,m}}{T_m} \leq m \cdot \left( 2^{\frac{1}{m}} - 1 \right) \quad (3.3)$$

sendo  $b_{l,i}$  o tempo de bloqueio no pior caso por uma tarefa de prioridade mais baixa ou seja,

$$\bar{b}_{l,m} = \max_{j=i+1,\dots,N} T_j \quad (3.4)$$

### 3.3.3 Análise do Tempo de Resposta no Pior Caso de Mensagens

Em Tindell et al. (1995) os autores apresentaram em detalhes as análises do tempo de resposta de redes CAN. Eles definiram que as mensagens possuíam prioridade fixa para os respectivos fluxos (acesso à rede baseado sobre a prioridade do identificador, assumindo que a cada fluxo de mensagens é atribuído um identificador único na rede) e um modelo de escalonamento não-preemptivo, ou seja, ao se transmitir as mensagens com menor prioridade, estas não serão interrompidas por mensagens de maior prioridade. Considerando tal modelo de escalonamento, adapta-se todas as análises de escalonabilidade existentes para o escalonamento de tarefas (Audsley et al. (1993)) para o caso de escalonamento de mensagens sobre uma rede CAN.

O tempo de resposta no pior caso de uma mensagem enfileirada, medida desde a chegada de uma requisição da mensagem para a fila de saída até o instante em que a mensagem é completamente transmitida, é dado em (3.5):

$$R_m = J_m + w_m + C_m \quad (3.5)$$

O termo  $C_m$  representa o tempo gasto para se transmitir uma mensagem  $m$  fisicamente sobre o barramento. O pacote de uma mensagem CAN básica, contém 47 bits de overhead por mensagem, e um *stuffing* de 5 bits, sendo apenas 34 bits dos 47 bits de overhead que estão sujeitos ao *stuffing*. O termo  $C_m$  é apresentado a seguir.

$$C_m = \left( \left\lfloor \frac{34 + 8 \cdot s_m}{5} \right\rfloor + 47 + 8 \cdot s_m \right) \cdot t_{bit} \quad (3.6)$$

Em (3.6), temos que o termo  $s_m$  indica o tamanho limitado da mensagem  $m$  em bytes, e o termo  $t_{bit}$  é o atraso de propagação, ou seja, o tempo necessário para se

transmitir um bit sobre o barramento, como exemplo, sob um barramento a  $1Mbit/s$ , resulta que  $t_{bit} = 1\mu s$ .

O termo  $w_m$  em (3.5), representa o atraso na fila no pior caso (o maior tempo entre a inserção de uma mensagem na fila de prioridades e o início de sua transmissão) representado por (3.7).

$$w_m = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{w_m + J_j + t_{bit}}{T_j} \right\rceil \cdot C_j \quad (3.7)$$

Sendo que  $hp(m)$  é o conjunto de mensagens no sistema com prioridade maior que a mensagem  $m$ , e  $B_m$  é o tempo de bloqueio no pior caso da mensagem  $m$ , assim representado: (3.8).

$$B_m = \max_{\forall k \in lp(m)} (C_k) \quad (3.8)$$

De (3.8) entende-se que  $lp(m)$  é o conjunto de fluxos de mensagens no sistema com prioridade menor que o fluxo de mensagem  $S_m$ .

Em (3.7), e o termo  $J_j$  corresponde ao *Jitter* de transmissão e  $T_j$  é o período de ativação do conjunto de mensagens com prioridade maior que a mensagem  $m$ .

Como o termo  $w_m$  em (3.7) aparece em ambos os lados da equação, a solução requer, assim, a utilização de uma relação de recorrência para sua solução, conforme apresentado em (3.9).

$$w_m^{n+1} = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{w_m^n + J_j + t_{bit}}{T_j} \right\rceil \cdot C_j \quad (3.9)$$

sendo o termo  $hp(m)$ , o conjunto de mensagens com prioridade maior que a mensagem  $m$ .

Finalmente, o termo  $J_m$  em (3.5), representa o *Jitter* da mensagem (tempo médio de espera de uma mensagem na fila de transmissão antes do escabname nto).

Em redes CAN é possível determinarmos atrasos de transmissão máximos sob a presença de carga através de prioridade sob mensagens.

### 3.4 Token Passing

O protocolo MAC PROFIBUS é baseado no procedimento de passagem do bastão usado por estações-mestres para garantir acesso ao barramento a cada outra, e um procedimento mestre-escravo usado pelas estações-mestres para comunicar com estações-escravos. Neste caso definem-se duas categorias de mensagens: mensagens de alta prioridade e mensagens de baixa prioridade, que utilizam duas filas de saídas independentes.

O procedimento de passagem do bastão PROFIBUS utiliza uma versão simplificada do protocolo Timed-Token (Grow, 1982) para controlar tanto o acesso ao barramento como também o tempo de ciclo do bastão. Este procedimento funciona como a seguir se descreve: Cada estação mestra, depois de receber o bastão, realiza a medição do início do tempo de rotação do bastão. Esta medição expira na próxima chegada do bastão e resulta no tempo de rotação real do bastão ( $T_{RR}$ ). Um valor comum do tempo de rotação-alvo do bastão ( $T_{TR}$ ) deve ser definido, devendo este ser pequeno o suficiente para satisfazer os requisitos de correção de resposta para o sistema global. Quando uma estação recebe o bastão, o temporizador do tempo de retenção do bastão ( $T_{TH}$ ) é dado pelo valor correspondente pela diferença, se positivo, entre os valores de  $T_{TR}$  e de  $T_{RR}$ .

Se, à chegada do bastão este estiver atrasado, ou seja, o tempo de rotação real do bastão ( $T_{RR}$ ) for maior que o maior tempo de rotação alvo do bastão ( $T_{TR}$ ), a estação-mestra pode transferir, no máximo, uma mensagem de alta prioridade. Caso contrário, a estação mestra pode transferir mensagens de alta prioridade enquanto  $T_{TH} > 0$ . As mensagens de baixa prioridade serão transferidas, quando não existirem mais mensagens de alta prioridade e, enquanto  $T_{TH} > 0$ .

Um conceito importante no protocolo PROFIBUS é o ciclo de mensagem. Um ciclo de mensagem consiste numa mensagem de ação do mestre (mensagem de requisição ou envio/requisição) associada a uma mensagem de reconhecimento ou resposta do receptor. O dado de controle pode ser transmitido na mensagem de ação ou na mensagem de resposta.

A seguir é apresentado o procedimento de passagem do bastão da rede PROFIBUS (Tovar, 1999):

```

1.1. /* procedimento de inicializacao */
1.2. Em cada estacao K, (K = 1, 2, ... , n):
1.3.      $T_{TH} \leftarrow 0$ ;
1.4.      $T_{RR} \leftarrow 0$ ;
1.5.     Libera  $T_{RR}$ ;          /* temporizador de crescente */
1.6.
1.7. /* procedimento de execucao */
1.8. Em cada estacao K, (K = 1, 2, ... , n), na chegada do Token, DO:
1.9.      $T_{TH} \leftarrow T_{TR} - T_{RR}$ ;
1.10.     $T_{RR} \leftarrow 0$ ;
1.11.    Libera  $T_{RR}$ ;          /* temporizador de crescente */
1.12.    IF  $T_{TH} > 0$  THEN
1.13.        Libera  $T_{TH}$ ;          /* temporizador decrescente */
1.14.    ENDIF;
1.15. IF Existe mensagens de alta prioridade pendentes
1.16.     Transfira uma mensagem de alta prioridade
1.17. ENDIF;
1.18. WHILE  $T_{TH} > 0$  AND Mensagens de alta prioridade DO
1.19.     Transfira mensagens de alta prioridade
1.20. ENDWHILE;
1.21. WHILE  $T_{TH} > 0$  AND Mensagens de baixa prioridade DO
1.22.     Transfira mensagens de baixa prioridade
1.23. ENDWHILE;

```

Assumindo uma rede com  $r$  estações (estações mestre para o caso PROFIBUS) e  $n$  fluxos de mensagens (fluxos de mensagens de alta prioridade para o caso PROFIBUS) definidas como:

$$S_m^k = (C_m^k, T_m^k, D_m^k) \quad (3.10)$$

Um fluxo de mensagem é uma seqüência temporal de mensagens relativas, por exemplo, à leitura remota de uma variável de processo específica. O termo  $C_m$  é a mais longa duração da mensagem do fluxo  $S_m$ .  $T_m$  é a periodicidade da requisição de fluxos  $S_m$ , considerado como o intervalo de tempo mínimo entre duas chegadas consecutivas de requisições para a fila de saída do fluxo  $S_m$ . Finalmente,  $D_m$  é a meta temporal relativa de uma mensagem; que consiste no intervalo de tempo máximo admissível entre o instante em que a requisição da mensagem é colocada na fila de saída e o instante em que cada mensagem é completamente transmitida.

Adicionalmente, para o caso PROFIBUS considera-se uma mensagem especial (o bastão), que circula ao longo do anel lógico formado pelos mestres. Denota-se a latência do anel lógico (tempo de rotação do bastão, incluindo o atraso de latência do nó computacional e o atraso de propagação do meio, etc) como  $t$ .

Em (Tovar e Vasques, 1999a), os autores sugerem duas abordagens diferentes para garantir o comportamento de tempo real do tráfego síncrono em redes PROFIBUS.

Na primeira abordagem – o Perfil de Tráfego de Baixa Prioridade Não Restrito, o requisito de tempo real para o tráfego de alta prioridade é satisfeito independentemente da carga de tráfego de baixa prioridade, desde que o protocolo garanta que no mínimo uma mensagem de alta prioridade seja transmitida por visita do bastão. Neste caso, o máximo tempo de resposta para a transmissão de uma mensagem do fluxo  $S_m$  é:

$$R_{mi}^k = Q^k + C_m^k = n^k \times T_{cycle}^k + C_{mi}^k \quad (3.11)$$

O termo  $T_{cycle}$  é a periodicidade que o bastão leva para passar por todos os nós computacionais, e  $n$  corresponde ao número de nó computacional sob a rede.

### 3.5 TDMA (Time Division Medium Access)

O protocolo TDMA (*Time Division Medium Access*) citado em Kurose et al., 1984, sendo o TTP - *Time Triggered Protocol*, a sua implementação mais conhecida, fornece um serviço de comunicação de múltiplo acesso e livre de colisão, permitindo que cada estação utilize periodicamente a capacidade total de transmissão ou largura de banda total, do canal de comunicação por alguma quantidade de tempo fixo.

Desta forma, o canal é compartilhado no tempo entre as estações, e o tempo é assim dividido dentro de intervalos de comprimentos fixos denominados por *slots*, como pode ser visto na Fig. 3.6.

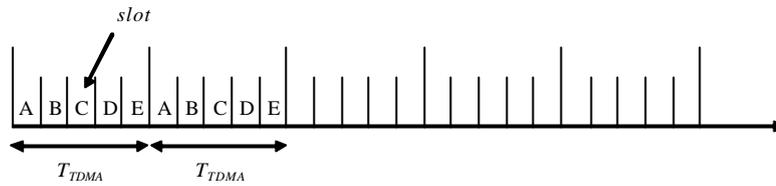


Figura 3.6: Esquema de acesso ao meio do protocolo TDMA

Em uma versão mais simplificada do TDMA, o número de *slots* por frame corresponde ao mesmo número de estações conectadas à rede e, a cada estação é garantida a utilização do canal para a duração de um *slot-time* por frame.

É importante notar que o protocolo TDMA é muito ineficiente. Uma primeira fonte de ineficiência surge, quando o número de estações na rede muda com o tempo, como em um ambiente de rádio na terra com estações móveis.

Uma segunda fonte de ineficiência ocorre quando uma estação não possui mensagem a ser transmitir, desta forma o *slot-time* referente à estação não é utilizado, mesmo que outras estações pudessem utilizá-lo.

Como será mostrado a seguir, apesar destes fatores negativos, o protocolo TDMA é determinista e bastante utilizado em aplicações de segurança crítica tais como freio ABS, air-bag e outras aplicações automotivas.

### 3.5.1 Modelo de Mensagens

Assumindo que uma rede TDMA possua  $r$  estações e  $n$  fluxos de mensagens, sendo cada fluxo definido por:

$$S_m = (C_m, T_m, D_m) \quad (3.12)$$

Um fluxo de mensagem é uma seqüência temporal de mensagens relativas a, por exemplo, leitura remota de uma variável de processo específico. Um fluxo de mensagem  $S_m$  é caracterizado por um único identificador.  $C_m$  é a maior tempo de duração da mensagem do fluxo  $S_m$ .  $T_m$  é a periodicidade de transmissão considerada como o intervalo de tempo mínimo entre duas chegadas consecutivas de requisições para a fila de saída do fluxo  $S_m$ . Finalmente,  $D_m$  é a meta temporal relativa de uma mensagem (deadline) do fluxo  $S_m$ , que consiste no intervalo de tempo máximo admissível entre o instante em que a

requisição da mensagem é colocada na fila de saída e o instante quando cada mensagem é completamente transmitida.

### 3.5.2 Análise do Tempo de Resposta no Pior Caso de Mensagens

A análise de escalabilidade para um simples processador é empregado ao caso de transmissão de mensagens em uma rede LAN, do tipo difusão que utiliza o protocolo TDMA de acesso ao meio, que foi inicialmente abordada por Tindell and Clark, 1994.

Neste trabalho será considerado que o tempo total de transmissão de uma mensagem  $m$  de um *nó computacional* a outro sob a rede é dado por:

$$R_m = C_m + W_m \quad (3.12)$$

sendo que  $C_{m,k}$  corresponde ao tempo necessário para transmitir uma mensagem do adaptador de comunicação-origem para o adaptador de comunicação-destino, e  $W_{m,k}$  é o tempo que uma mensagem gasta para ser totalmente removida da fila de pacotes do adaptador de comunicação.

Para uma dada mensagem  $m$ , transmitida entre um *nó computacional*  $k$  de origem para o *nó computacional*  $j$  de destino, são apresentadas algumas notações importantes para o desenvolvimento da ferramenta matemática necessária para análise de escalabilidade de mensagens no protocolo TDMA.

- $C_m$  - Representa o número de unidades de tempo necessárias para transmitir uma mensagem  $m$ ;
- $P_m$  - É o número de pacotes de tamanho fixo que a mensagem  $m$  é composta;
- $T_m$  - É o período de ativação da mensagem;
- $S_p$  - É o número de pacotes que o processador  $p$  é permitido transmitir no *slot* reservado a ele.

Em cada ciclo  $T_{TDMA}$ , o adaptador de comunicação conectado ao *nó computacional* processador na rede remove  $S_p$  pacotes da fila e começa a transmiti-los por completo. O número de ciclos  $T_{TDMA}$  necessários no pior caso para que todos os pacotes de uma mensagem  $m$  sejam removidos na fila de pacotes para serem transmitidos, é dado por:

$$\left\lceil \frac{P_m + I_m}{S_p} \right\rceil T_{TDMA} \quad (3.13)$$

Sendo que  $I_m$  é a interferência das mensagens de maior prioridade sob a mensagem  $m$  que podem ser inseridos em sua frente (quantidade de pacotes com prioridade maior que), o número de pacotes da mensagem de prioridade mais alta que  $m$ , num instante  $T_m$ , corresponde ao período de ativação da mensagem  $m$ , que pode ser dado por Eq. (3.14).

A interferência ocorre, se num mesmo nó computacional o processador tiver mensagens concorrendo por serem transmitidas.

$$I_m = \sum_{\forall j \in hp(m)} \left\lceil \frac{T_m}{T_j} \right\rceil P_j \quad (3.14)$$

Sendo  $T_j$  o período de ativação das mensagens com prioridade mais alta que  $m$  e  $P_j$  é o número de pacotes das mensagens de prioridade mais alta.

Então a equação que descreve o tempo que uma mensagem  $m$  gasta para ser removida da fila de pacotes é dada por:

$$W_m = \left\lceil \frac{P_m + \sum_{\forall j \in hp(m)} \frac{T_m}{T_j} P_j}{S_p} \right\rceil T_{TDMA} \quad (3.15)$$

Para encontrar o termo  $C_m$  na equação 3.12, determina-se que os  $S_p$  pacotes transmitidos no último *slot* TDMA correspondem ao último pacote da mensagem  $m$ .

O número máximo de pacotes que precisam ser levados da fila no instante  $T_m$  de forma a garantir a transmissão do último pacote de  $m$  é dado por:

$$x = P_m + I_m \quad (3.16)$$

O número de *slots* necessários para transmitir estes pacotes pode ser dado por:

$$s = \left\lceil \frac{x}{S_p} \right\rceil \quad (3.17)$$

Os primeiros  $(s-1)$  slots de transmissão não levaram o último pacote da mensagem  $m$ , por isso o último *slot* irá transportar o último pacote da mensagem  $m$ , que será denotado como  $a$  ésimoo pacote do *slot* final onde  $a$  denota-se por:

$$a = x - (s - 1)S_p \quad (3.18)$$

Sendo que  $a$  corresponde a quantidade dos últimos pacotes a serem transmitidos no último *slot*. O tempo gasto para transmitir todos os pacotes anteriores aos últimos pacotes do último *slot* da mensagem  $m$  é representado por:

$$a_m = (x - a) \cdot S_p \quad (3.19)$$

Sendo que  $a$  é o tempo gasto para transmitir um pacote sobre o barramento de comunicação. Desta forma o tempo gasto para transmitir uma mensagem  $m$  sob o barramento de comunicação é dado por:

$$A_m = a_m + a \cdot S_p \quad (3.20)$$

sendo:

$$C_m = (x - a) \cdot S_p + (x - (s - 1)S_p) \cdot r \quad (3.21)$$

O tempo total para transmissão de uma mensagem  $m$  entre os adaptadores de comunicação origem e destino pode ser dado por:

$$R_m = C_m + W_m$$

$$R_m = (x - a) \cdot S_p + (x - (s - 1)S_p) \cdot r + \left[ \frac{P_m + \sum_{\forall j \in hp(m)} \frac{T_m P_j}{T_j}}{S_p} \right] T_{TDMA} \quad (3.22)$$

### 3.6 Conclusão

Neste capítulo foram apresentadas as propriedades temporais de tarefas em tempo real num sistema computacional e mensagens sob uma rede de comunicação do tipo difusão, que suporta tráfego de mensagens em tempo real. Realizou-se uma análise das propriedades temporais de tarefas e mensagens com características semelhantes, sendo apenas diferenciadas no fator preempção, onde as mensagens não podem ser interrompidas durante sua transmissão.

A escolha e análise de propriedades das redes: CAN, Token-passing e TDMA, foram essenciais para o desenvolvimento deste trabalho cujo objetivo consiste em suportar sistemas de controle via redes. As aplicações de controle fechadas sob uma rede de comunicação requerem um tráfego de mensagens em tempo real, sendo que para as redes apresentadas acima esta análise foi desenvolvida baseada no cálculo do tempo de resposta no pior caso de mensagens.

# 4 Sistemas de Controle via Redes (NCS)

## 4.1 Introdução

Com a grande disponibilidade e o baixo custo da atual tecnologia de redes de comunicação de dados, a comunidade de sistemas de controle notou uma grande utilização destas para interconexão de sensores, atuadores e controladores em sistemas de controle realimentado. Este tipo de implementação em que as malhas de controle são fechadas sob uma rede de comunicação é habitualmente denominada por Sistemas de Controle via Redes (ou NCS, para “*Networked Control Systems*”).

Em torno dos NCS está se desenvolvendo uma nova área de pesquisa multidisciplinar, envolvendo as comunidades de *sistemas de controle*, *sistemas de tempo real* e *redes de comunicação* (Ärzén *et al.*, 1999).

O problema de NCS assemelha-se ao de escalonamento em tempo real e controles integrados (ver um estudo detalhado em Ärzén *et al.*, (1999)) pelo fato de que possuem mensagens que concorrem para serem transmitidas no canal de comunicação. Basicamente, deve-se definir uma rede de comunicação com um protocolo de escalonamento em tempo real para as mensagens concorrentes de forma que os algoritmos de controle tenham suas restrições temporais garantidas (Tovar, 1999). Em conseqüência, é efetuada uma análise de uma arquitetura genérica, envolvendo todos os principais problemas que podem ser encontrados num projeto de diferentes arquiteturas de NCS.

Um dos principais requisitos de um NCS com relação à rede de comunicação, é que esta ofereça um serviço de comunicação de tempo real. Subjacente a este tipo de serviço, deve estar a possibilidade de *suportar fluxos de mensagens periódicas*, de forma a transferir dados periódicos relacionados com o controle, a capacidade em garantir um *tempo de resposta limitado superiormente* para transferência de mensagens entre os *nós computacionais* e a capacidade de garantir um *comportamento temporal previsível* na presença de carga de rede variável por causa do tráfego não relacionado à aplicação de controle.

Existem diferentes redes candidatas ao suporte de *NCS*, devido a fornecerem serviços de comunicação de tempo real adequados. Como possíveis redes candidatas, incluem-se neste caso as redes que possuam um protocolo MAC determinístico, tal como os apresentados no Capítulo 3.

## 4.2 Sistemas de Controle por Computador

Uma malha clássica de controle por computador consiste em três partes principais: o elemento sensor para aquisição de dados, o computador para a execução do algoritmo de controle e o atuador para transmissão do sinal de controle à planta física.

Quando diversas malhas de controle são fechadas sob um computador, os algoritmos de controle tornam-se programas concorrentes executando sobre recursos computacionais compartilhados e escassos, como CPU, memória, e unidades de E/S (ou I/O de *I* para *Input* e *O* para *Output*). Faz-se assim necessário uma política de escalonamento de tarefas, de forma que os diferentes programas tenham uma execução correta e garantida, em termos lógicos e temporais. Para isto, é fundamental a consideração do paradigma de tempo real que contempla os requisitos lógicos e temporais de um conjunto de programas concorrentes (Cervin, 2003).

Considere um algoritmo de controle executados sob um conjunto de recursos computacionais escassos e compartilhados por outros algoritmos. A planta física controlada é amostrada periodicamente em intervalos de tempo  $h$ .

O instante exato de medição da variável de saída e atuação na variável de controle da planta física é denominado por *I* e *O*. O instante da liberação para execução da tarefa de controle é denotado por  $r_i$ .

O atraso entre o instante de medição e atuação na planta física,  $L_{IO}$ , é denominado por *atraso de entrada-saída* (ou *atraso de controle*) e pode ser variável em função da preempção do tempo de execução de um algoritmo de controle  $C_i$  (que é definido a partir do *WCET* – *Worst Case Execution Time*), por tarefas de maior prioridade ou em situações de bloqueio. A variação do *atraso de controle* é denominada como o *jitter de controle*,  $J_{IO}$ .

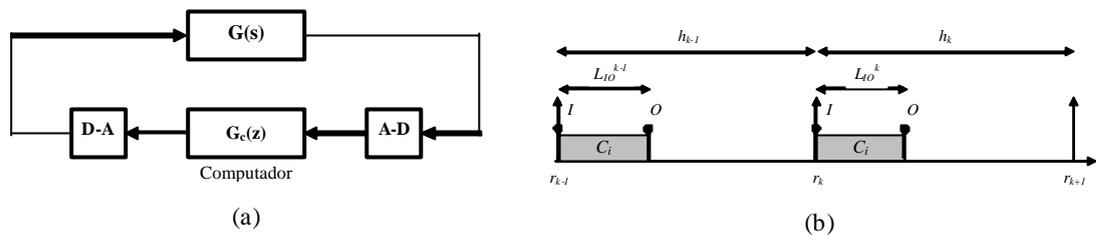


Figura 4.1: (a) Diagrama em blocos de um sistema de controle por computador e (b) Diagrama temporal de um controlador digital.

A Fig. 4.1 apresenta um diagrama em blocos para um sistema de controle por computador e o diagrama temporal para um controlador digital. Note que para este caso, onde existe apenas uma malha de controle, o controlador digital não possui *atraso no instante de amostragem*  $L_S$  e conseqüentemente não possui *jitter de amostragem*  $J_S$  e nem *jitter de entrada-saída*  $J_{IO}$ . Esta idéia nos dá condições de realizarmos projetos de controladores, considerando apenas *período de amostragem*  $h$  e *atraso de entrada-saída*  $L_{IO}$ .

Em (Cervin, 2003) foi apresentado um estudo sobre o impacto do *atraso e jitter do instante de amostragem, entrada-saída e período de amostragem* sob o desempenho do controle (QoC – Qualidade do Controle), como base para o problema de escalonamento em tempo real e controle integrados, apresentando também as soluções alternativas.

Em um sistema de controle por computador o controlador pode ser implementado de duas formas distintas (Cervin *et al.*, 1999), na primeira o projeto é desenvolvido no domínio de tempo contínuo e em seguida realizado a discretização do modelo com a implementação do controlador na forma digital. Na segunda abordagem, todo o projeto é realizado baseado na teoria de controle digital e imediatamente o controlador é implementado na forma digital.

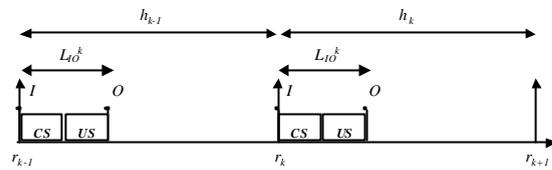
A implementação do algoritmo de controle num computador digital é realizada na forma algorítmica com sua codificação e comportamento temporal, podendo ser visto na Fig. 4.2.

```

t := TempoCorrente;
LOOP
  LeEntrada;
  AlgoritmoControle;
  AtualizaEstado;
  EscreveSaida;
  t := t + h;
  EsperaAte(t);
END

```

(a)



(b)

Figura 4.2: (a) Algoritmo de controle clássico e (b) Diagrama temporal de um controlador digital clássico.

Para este tipo de implementação clássica o algoritmo de controle é composto um parte do código que executa os cálculos para gerar o sinal de saída do controlador *CS* (*Code Segment*) e a parte do código que atualiza os estados do controlador *US* (*Update State*), onde o tempo de execução de ambos compreende no *atraso de entrada-saída*. Existem implementações de sistemas de controle por computador onde a desempenho de controle está diretamente relacionada com o *atraso de entrada-saída*.

O código do controlador é composto em duas partes distintas, *CS* e *US*, onde o instante de atuação na planta física ocorre exatamente após a execução das partes *CS* e *US*, num determinado ciclo de controle. O pedaço de código *US* introduz um *atraso de entrada-saída* considerável que pode vir a comprometer a qualidade de controle da planta física.

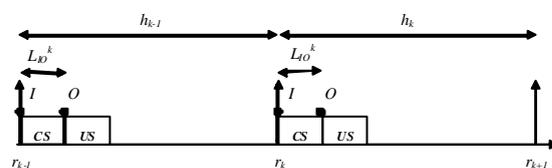
Uma solução alternativa para a redução do *atraso de entrada-saída* pode ser vista na Fig. 4.3. Note que o instante de atuação na planta de controle ocorre exatamente quando o parte de código *CS* termina, reduzindo assim o *atraso de entrada-saída*.

```

t := TempoCorrente;
LOOP
  LeEntrada;
  AlgoritmoControle;
  EscreveSaida;
  AtualizaEstado;
  t := t + h;
  EsperaAte(t);
END

```

(a)



(b)

Figura 4.3: (a) Algoritmo de controle modificado e (b) Diagrama temporal de um controlador digital modificado.

O principal problema de um sistema de controle por computador ao nível de algoritmo de controle consiste em como definir o *período de amostragem*  $h$  e *atraso de entrada-saída*  $L_{IO}$  de forma a garantir um desempenho e estabilidade aceitável. Nesta tese não iremos considerar o *jitter de amostragem* e *entrada-saída* (Cervin, 2003).

Serão apresentados a seguir dois métodos para determinação do *período de amostragem* e *atraso de entrada-saída* de um sistema de controle por computador realimentado, *Regra de Thumb (Rule of Thumb)* e *Critério de Desempenho Quadrático*.

#### 4.2.1 Projeto do Controlador no Domínio de Tempo Contínuo

Considere um servo-sistema do tipo *motor-dc*, onde o objetivo de controle é que a saída  $y(t)$  siga um sinal de referência  $y_{ref}(t)$  o mais próximo possível. O servo-sistema é representado por uma função de transferência e um controlador do tipo PD (proporcional mais derivativo), projetado e implementado na forma de um controlador discretizado. O procedimento de projeto, que pode ser encontrado em literaturas clássicas de sistemas de controle, é descrito brevemente a seguir.

Tabela 4.1. Parâmetros de projeto do controlador PD.

Processo $G(s)$	$w_b$ (rad/s)	Controlador PD	
		$Kd$	$Td$
$G(s) = \frac{1000}{s(s+1)}$	40	1	0,04

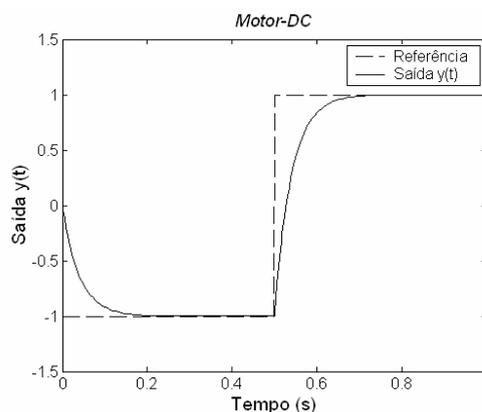


Figura 4.4: Desempenho em malha fechada do motor-dc para um sinal de referência  $y_{ref}(t)$ .

A partir do modelo dinâmico do processo, especifica-se um índice de desempenho do tipo largura de banda  $w_b$ , desejada em malha fechada para o sistema de controle. Desta forma, é possível algebricamente determinar os parâmetros  $Kp$  e  $Td$  do controlador PD, conforme apresentados na Tabela 4.1. A Fig. 4.4 apresenta a resposta  $y(t)$  em malha fechada do *motor-dc* para um sinal de referência  $y_{ref}(t)$ . Note que o sistema foi projetado para satisfazer os requisitos de resposta sub amortecida.

#### 4.2.2 Regra de Thumb

A seleção do *período de amostragem* para um sistema de controle por computador (realimentado) pode ter um desempenho aceitável, se obedecer à *regra de thumb* (Åstrom and Wittenmark, 1997). A regra sugere que o intervalo de amostragem  $h$  deverá ser escolhido tal que:

$$0,2 < w_b \times h < 0,6 \quad (4.1)$$

sendo que  $w_b$  corresponde à largura de banda do sistema em malha fechada. Para os parâmetros de projeto do sistema de controle da Tabela 4.1, a seleção do *período de amostragem* deverá satisfazer a desigualdade  $5 < h < 15$  com  $h$  dado em milisegundos, para que o sistema de controle tenha um desempenho e estabilidades aceitáveis.

Para utilização deste método, é necessário que se conheça o modelo do sistema de controle em malha fechada com sua largura de banda. Dada a sua simplicidade, é largamente utilizado, porém não proporciona informações do efeito para o *atraso de entrada-saída* sobre o desempenho de controle. De forma experimental um desempenho aceitável pode ser garantido para *atraso de entrada-saída*  $L_{IO}$  em torno de 10% do *período de amostragem*  $h$ .

A *regra de thumb*, apesar de ser funcional, não gera informações sob a faixa de *atraso de entrada-saída* ( $L_{IO}$ ) aceitável, apresentando um recurso limitado na implementação de sistemas de controle de tempo real em que as tarefas podem ter tempos de resposta variável e imprevisível.

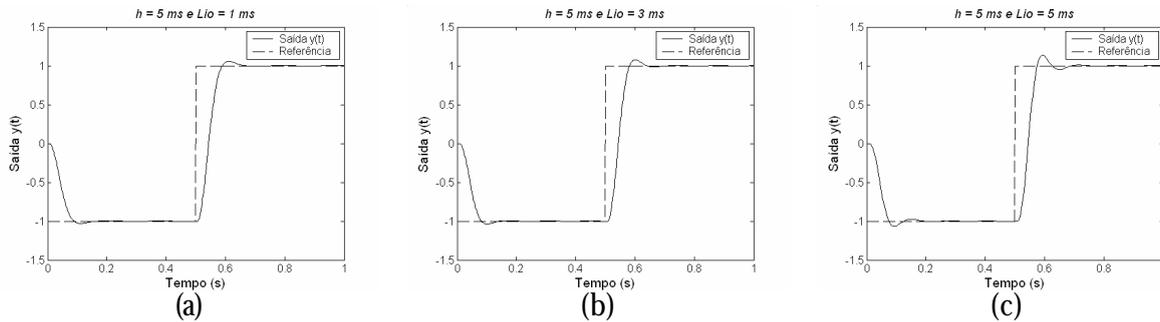


Figura 4.5: (a)  $h = 5$  ms e  $L_{10} = 1$  ms, (b)  $h = 5$  ms e  $L_{10} = 3$  ms, (c)  $h = 5$  ms e  $L_{10} = 5$  ms.

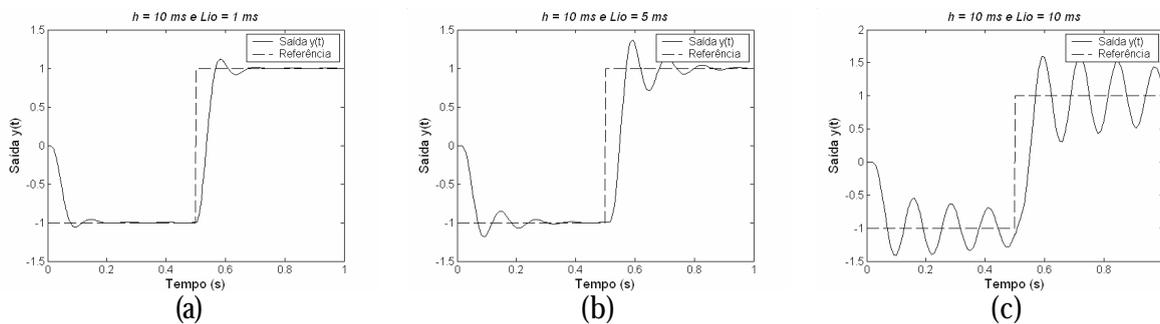


Figura 4.6: (a)  $h = 10$  ms e  $L_{10} = 1$  ms, (b)  $h = 10$  ms e  $L_{10} = 5$  ms, (c)  $h = 10$  ms e  $L_{10} = 10$  ms.

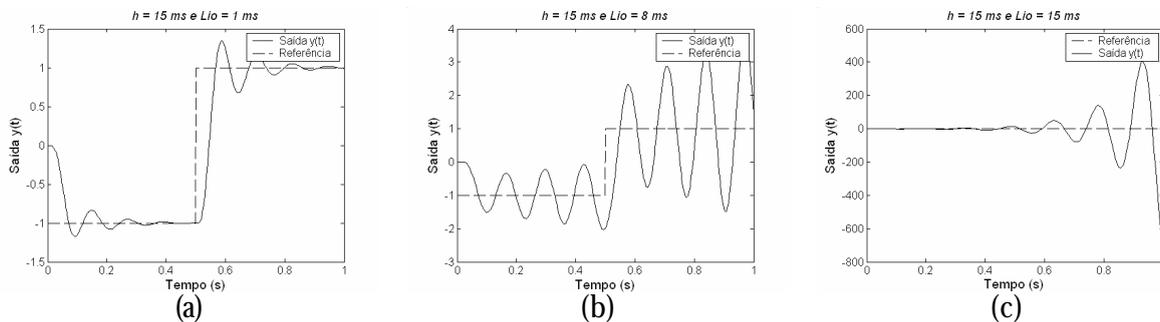


Figura 4.7: (a)  $h = 15$  ms e  $L_{10} = 1$  ms, (b)  $h = 15$  ms e  $L_{10} = 8$  ms, (c)  $h = 15$  ms e  $L_{10} = 15$  ms.

As Fig. 4.5 à 4.7 apresentam as simulações da resposta  $y(t)$  em malha fechada do motor-dc para um sinal de referência  $y_{ref}(t)$  para diferentes *períodos de amostragem* e *atrasos de entrada-saída*. Note que o desempenho e estabilidade são influenciados pela seleção do *período de amostragem* e *atraso de entrada-saída*.

Podemos verificar que, através de simulações, tornam-se exaustivas a análise e a escolha do melhor *período de amostragem* ( $h$ ) e *atraso de entrada-saída* ( $L_{10}$ ) para o projetista do sistema de controle.

Outro fator importante para a seleção do *período de amostragem* e *atraso de entrada-saída* é a utilização da CPU pelo algoritmo de controle, que pode ser assim representado por  $U_{cpu} = L_{IO}/h$  (Vamos considerar que o *atraso de entrada-saída*  $L_{IO}$  é o mesmo tempo de execução do algoritmo de controle  $C_i$  dado pelo seu *WCET*).

Para *período de amostragem* maior, tem-se a vantagem da menor utilização de uma CPU, por exemplo:  $U_{cpu}(Fig.4.5(a)) = 1/5 = 0.2$  ou 20% de utilização da CPU. Porém com uma sensibilidade maior na qualidade do controle para valores consideráveis de latência de entrada-saída. Para período de amostragem menor, a qualidade de controle é consideravelmente melhor, porém a utilização da CPU torna-se maior subutilizando o recurso computacional, por exemplo:  $U_{cpu}(Fig.4.7(a)) = 1/15 = 0.066$  ou 6.7% de utilização da CPU.

O projeto de sistemas de controle por computador requer um compromisso entre qualidade de controle, seleção de período de amostragem e utilização dos recursos computacionais. Para isto, técnicas de projeto de controladores, baseadas numa função-custo da qualidade de controle em função do *período de amostragem* e *atraso de entrada-saída* é apresentado a seguir, quando o projetista pode, através de funções-custo, definir quais valores de *período de amostragem* e *atraso de entrada-saída* para a implementação do controlador.

### 4.2.3 Critério de Desempenho Quadrático

A *regra de thumb* apesar de ser facilmente utilizada, apresenta uma deficiência, quando necessitamos de um projeto para sistema de controle por computador cuja qualidade do desempenho de controle seja fundamental.

O efeito combinado do *período de amostragem* e *atraso de entrada-saída* pode ser facilmente analisado baseado no desempenho do controlador e avaliado a partir de uma função-custo quadrática (Cervin, 2003).

$$J = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T (y^2(t) + u^2(t)) dt \quad (4.2)$$

A função-custo (4.2) para um sistema de controle por computador pode ser avaliada para diferentes *períodos de amostragem* num determinado intervalo de tempo e para uma faixa de *atrasos de entrada-saída* entre 0 a 100% do intervalo de amostragem.

Baseado numa *função-custo de qualidade de controle* (QoC), é possível determinarmos um espectro da qualidade do controle em função do *período de amostragem* e *atraso de entrada-saída*. Uma ferramenta computacional importante para este fim é o Jitterbug (Cervin and Lincoln, 2003), que a partir da planta física e do controlador gera graficamente uma curva que relaciona a qualidade do controle da planta em malha fechada, em função do *período de amostragem* e do *atraso de entrada-saída*.

$$J = f(h, L_{IO}) \quad (4.3)$$

Como exemplo considere um sistema de controle da Tabela 4.1

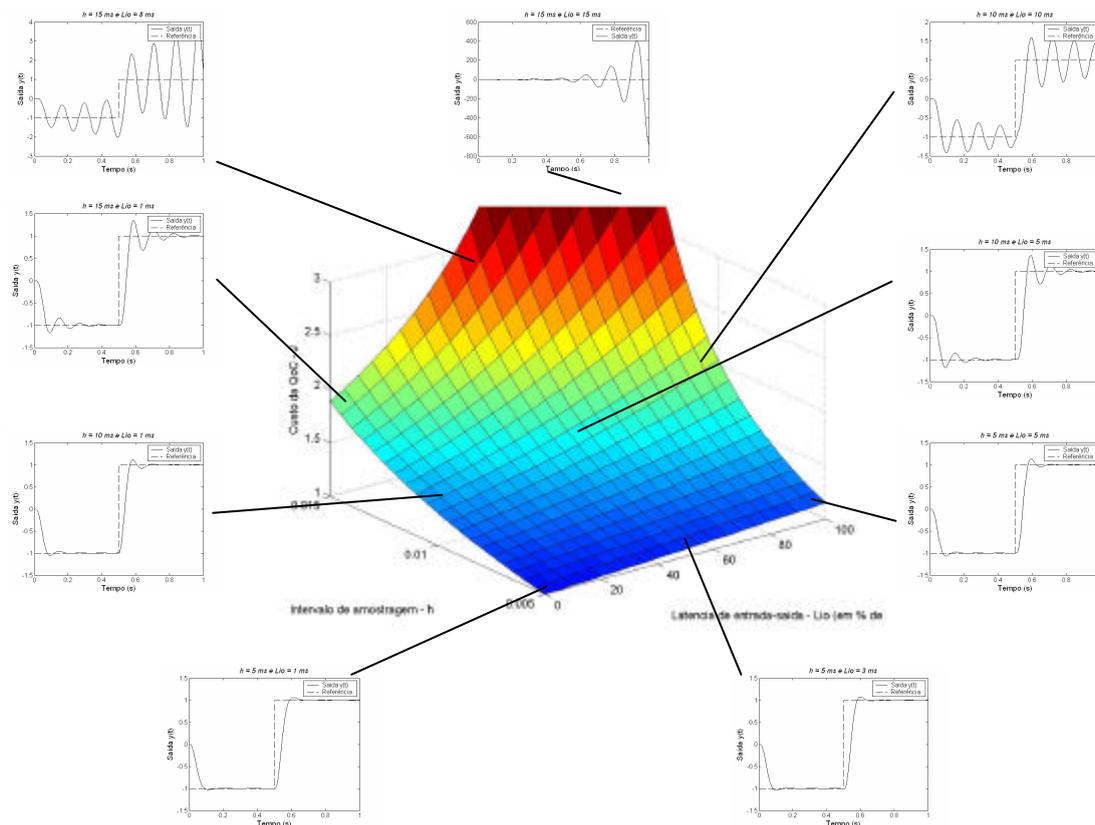


Figura 4.8: O gráfico da função  $J(h_k, L_{IO})$  para o sistema de controle da Tabela 4.1.

A ferramenta computacional *Jitterbug* gera, a partir da seleção de parâmetros internos de entrada do software, um gráfico de desempenho da qualidade da malha de

controle em função do *período de amostragem* e do *atraso de entrada-saída*, conforme pode ser visto na Fig. 4.8.

Para valores de  $h = 15 \text{ ms}$  e  $L_{IO} = 8 \text{ ms}$  ( $L_{IO}$  corresponde a 53% de  $h$ ) no gráfico da Fig. 4.8, percebe-se que a função-custo apresenta um valor no gráfico, utilizando uma cor avermelhada. Significa que o desempenho do sistema de controle por computador para estes valores de  $h$  e  $L_{IO}$  apresenta um desempenho marginalmente estável. Isto pode ser verificado na Fig. 4.7, através dos resultados de simulação.

Vamos considerar agora os valores de  $h = 5 \text{ ms}$  e  $L_{IO} = 3 \text{ ms}$  ( $L_{IO}$  corresponde a 60% de  $h$ ) no gráfico da Fig. 4.8, onde se percebe que a função-custo apresenta um valor que no gráfico aparece agora na cor azulada. Significa que o sistema de controle por computador para estes valores de  $h$  e  $L_{IO}$  apresenta um desempenho estável. Isto pode ser verificado na Fig. 4.5, através dos resultados de simulação.

Podemos notar que o gráfico da Fig. 4.8 nos dá informações suficientes para o projeto de sistemas de controle por computador, apresentando assim o impacto do *período de amostragem* e *atraso de entrada-saída* sob a qualidade do controle.

### 4.3 Sistema de Controle via Redes

Considere agora um sistema de controle por computador onde os dispositivos de campo (sensores, atuadores e controladores) são interconectados por uma rede de comunicação de dados com a vantagem inerente no aumento da descentralização das tarefas de controle e de medição. Este tipo de sistema de controle, em que a malha de controle é fechada através de uma rede de comunicação de dados denomina-se por Sistema de Controle via Redes (ou NCS, de “*Networked Control Systems*”).

Os elementos *sensor*, *controlador* e *atuador* podem ser considerados sistemas computacionais com uma interface de comunicação sob a rede (*nós computacionais* de comunicação). A disposição de *nós computacionais* sob a rede para a formação de um NCS pode ser realizada de três formas distintas, caracterizando assim três possíveis arquiteturas de NCS conforme pode ser visto na Fig. 4.9.

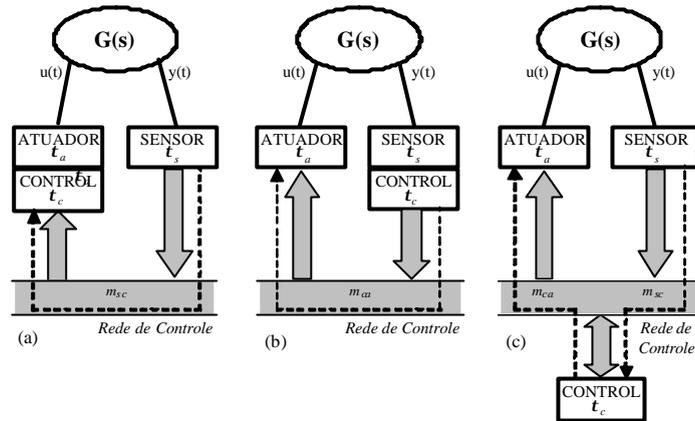


Figura 4.9: Arquiteturas de NCS com a representação de tarefas e mensagens

Dois tipos de atrasos devem ser considerados em *NCS*: os atrasos relacionados com a execução das tarefas sob os *nós computacionais* de processamento e os atrasos relacionados com a transmissão de mensagens sob a rede de comunicação. Conforme apresentado na Fig. 4.9, a diferença das arquiteturas apresentadas está na quantidade de fluxos de mensagens sob a rede de comunicação, nos atrasos e variações de atrasos na execução de tarefas e na transmissão de mensagens. Todos estes aspectos podem influenciar diretamente no desempenho e estabilidade do sistema de controle do *NCS*.

As Fig. 4.9 (a) e (b) apresentam uma arquitetura de *NCS* com dois *nós computacionais* contendo como diferença básica a disposição do controlador e geram apenas um fluxo de mensagem sob a rede de comunicação. A Fig. 4.9 (c) apresenta uma arquitetura de *NCS* com três *nós computacionais* e gera dois fluxos de mensagens sob a rede de comunicação.

Conforme as arquiteturas de *NCS* apresentadas na Fig. 4.9, as unidades (*nós computacionais*) de processamento são caracterizadas por tarefas do tipo medição  $t_s$ , execução do algoritmo de controle  $t_c$  e atuação  $t_a$ , as quais necessitam de um determinado tempo de processamento da CPU para serem executadas. O tempo de execução das tarefas é dado a partir do *Tempo de Execução no Pior Caso* (ou *WCET*, de “*Worst Case Execution Time*”) e denotados por:

$C_{t_s}$  = WCET para a tarefa de medição  $t_s$

$C_{t_c}$  = WCET para a tarefa de execução do algoritmo de controle  $t_c$

$C_{t_a}$  = WCET para a tarefa de atuação  $t_a$

A rede de comunicação de dados possui uma largura de banda dada em *bps* (bits por segundo) e transmite mensagens do tipo *sensor-controlador*  $m_{sc}$  e *controlador-atuador*  $m_{ca}$ . O tempo de transmissão de uma mensagem é dado por:

$C_{m_{sc}}$  = Tempo de transmissão da mensagem  $m_{sc}$  do *nó computacional* sensor para o *nó computacional* controlador

$C_{m_{ca}}$  = Tempo de transmissão da mensagem  $m_{ca}$  do *nó computacional* controlador para o *nó computacional* atuador

#### 4.3.1 Caracterização das Tarefas

A tarefa sensor  $t_s$  deverá ser uma tarefa com *ativação periódica* (“*time-triggered*”), que amostra a saída do processo  $y(t)$  e envia o valor de amostragem ao *nó computacional* controlador. O atraso de processamento do *nó computacional* sensor, que inclui a conversão *A/D* (análogo-digital) e a fila para transmissão de mensagens, é dada por:  $C_{t_s}$ .

A tarefa controladora  $t_c$  deverá ser uma tarefa com *ativação pela ocorrência de um evento significativo* (“*event-triggered*”), ativada pela chegada do valor de sensor relacionado. O atraso de processamento desta tarefa é dado por:  $C_{t_c}$ , e inclui ambos os atrasos: a) atraso de despacho por interrupção e b) atraso de processamento do algoritmo de controle.

A tarefa atuador  $t_a$  deverá ser também uma tarefa com *ativação pela ocorrência de um evento significativo* que envia um sinal de correção à entrada do processo  $u(t)$ , ativada pela chegada ao *nó computacional* controlador do valor de atuação relacionado. O atraso de processamento desta tarefa é dado por:  $C_{t_a}$ , e inclui o despacho de interrupções e os atrasos de conversão *D/A* (digital-analógico).

Nota-se que os *nós computacionais* de processamento executam tarefas independentes e sem concorrência, considerando assim que para as arquiteturas de *NCS* da Fig. 4.9 o tempo de computação das tarefas seja constante.

### 4.3.2 Caracterização das Mensagens

Os atrasos de comunicação incluem os atrasos de acesso ao meio de comunicação (escalonamento de mensagens) e da transmissão de mensagens. No entanto, enquanto o atraso de transmissão de mensagem é aproximadamente constante, o atraso de acesso ao meio é altamente variável por depender da carga na fila de transmissão sob a rede, num determinado instante e, também, sob o protocolo de acesso ao meio utilizado.

A rede de comunicação é um recurso compartilhado no qual o *nó computacional* de processamento desejam transmitir mensagens. Quando existir apenas um *nó computacional* transmitindo um fluxo de mensagens sob a rede, o tempo de transmissão desta é dado por  $C_{m_i}$ .

Considere agora a situação onde vários *nós computacionais* sob a rede de comunicação desejamos transmitir mensagens. Para esta situação pode-se determinar o *tempo de transmissão no pior caso para mensagens* (ou *WCRT*, de “*Worst Case Resposta Time*”), que é dado por  $R_{m_i}$ .

A implementação de um sistema de controle que utiliza uma rede de comunicação para interconectar sensores, controladores e atuadores requer que a rede de comunicação ofereça um *serviço de tempo real* na transmissão de mensagens, ou seja:

a) capacidade de *suportar fluxos de mensagens periódicas*, de forma a transferir dados periódicos relacionados com o controle (por exemplo, dados de leitura periódica de um grupo de sensores);

b) capacidade de garantir um *tempo de resposta com limite superior* para transferência de mensagens entre os *nós computacionais*, de forma a respeitar os atrasos máximos de controle relacionados às malhas de controle;

c) e, acima de tudo, capacidade de garantir um *comportamento temporal previsível* na presença de carga de rede variável por causa do tráfego distinto à aplicação de controle (tais como: alarmes, fluxos de vídeo relacionados com a vigilância de instalações, etc).

Em adição, um problema recorrente, quando se utiliza uma rede de comunicação para suportar aplicações de controle, é a presença de *jitter*, ou seja, a variabilidade do intervalo de tempo entre transferências consecutivas de mensagens pertencentes a um mesmo fluxo periódico.

Por exemplo, considere o caso de um sensor que, periodicamente, requisita a transferência do valor do seu estado, esta não será imediatamente executada, pois as mensagens necessitam serem escalonadas para transmissão em um recurso compartilhado (o meio de comunicação). Como consequência, em alguns ciclos a mensagem do sensor será transferida mais cedo no período do ciclo e em outros ciclos ela será transferida mais tarde, gerando assim uma variação no atraso de controle denominado por *jitter de controle (jitter de entrada-saída)*. O serviço de tempo real fornecido pela rede de controle deverá garantir que, apesar desta variabilidade, as mensagens do sensor serão sempre transferidas, antes das respectivas metas temporais (“*deadlines*”).

O problema do *jitter de entrada-saída* pode ser mais acentuado quando a rede de comunicação é do tipo *event-triggered* (CAN e Ethernet) ou os nós computacionais de processamento são compartilhados entre múltiplas malhas de controle. Neste caso, uma requisição particular de um sensor para transferência de dados, pode ser imediatamente efetuada, caso não haja mais nenhuma requisição pendente, ou pode ser atrasada no tempo, caso existam requisições múltiplas pendentes para serem efetuadas no mesmo *nó computacional*. A mesma condição pode ser verificada para outros *nós computacionais* que possuam tarefas compartilhadas.

Para este caso, no qual existem problemas de concorrência quer entre tarefas (nos *nós computacionais*), quer entre mensagens (na rede de comunicação), a utilização de um kernel de tempo real e de uma rede de comunicação que forneça um serviço de tempo real não é suficiente para garantir uma latência de execução e comunicação bem definidas e, conseqüentemente, manter o desempenho e a estabilidade nas malhas de controle. Isso se

deve ao fato de que as mensagens induzem atrasos no período de amostragem, entrada-saída e *jitter* nas malhas de controle.

A variação dos atrasos de comunicação impõe uma variação de tempo para o atraso de execução do controle, que é definido como *jitter de controle*, que tem uma forte influência sobre a estabilidade e desempenho de um *NCS*.

Dois abordagens opcionais podem ser usadas para caracterizar tais atrasos de comunicação:

- a) ou, considera-se a análise do máximo tempo de resposta, que garante que as mensagens de controle nunca serão atrasadas mais que um atraso superior específico.
- b) ou, assume-se que os resultados de simulação são suficientes para garantir as propriedades de estabilidade e desempenho do *NCS*.

### **4.3.3 Caracterização Temporal das Tarefas e Mensagens**

Em um sistema de controle por computador, o controlador irá amostrar os sinais de entrada a cada intervalo de tempo  $h$ , lendo diretamente o valor de sensor através de um conversor *A/D*. Após a execução do algoritmo de controle, o controlador irá atualizar cada atuador, escrevendo diretamente o valor relacionado a um conversor *D/A*. O atraso de execução do controle (atraso de entrada-saída) é dado pelo tempo gasto entre o instante de medição e o instante de atuação.

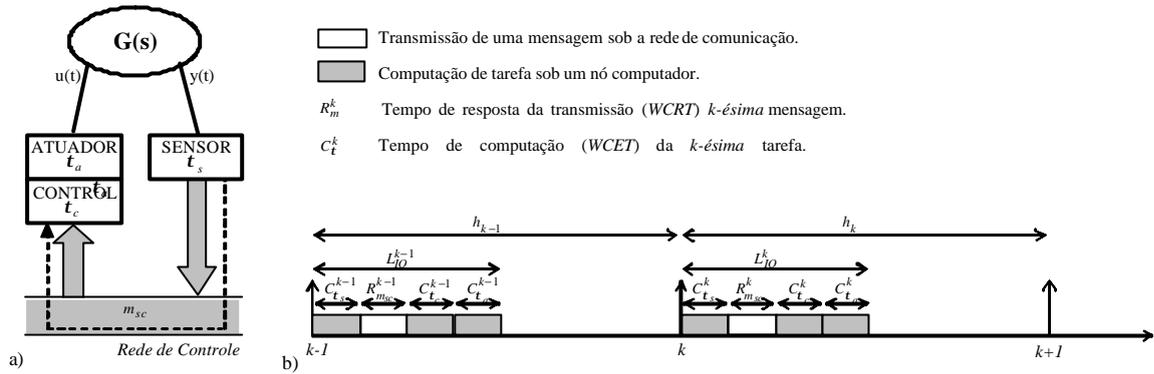


Figura 4.10: (a) Arquitetura de NCS com nó computacional integrando controlador e atuador; (b)

Diagrama temporal das tarefas e mensagens.

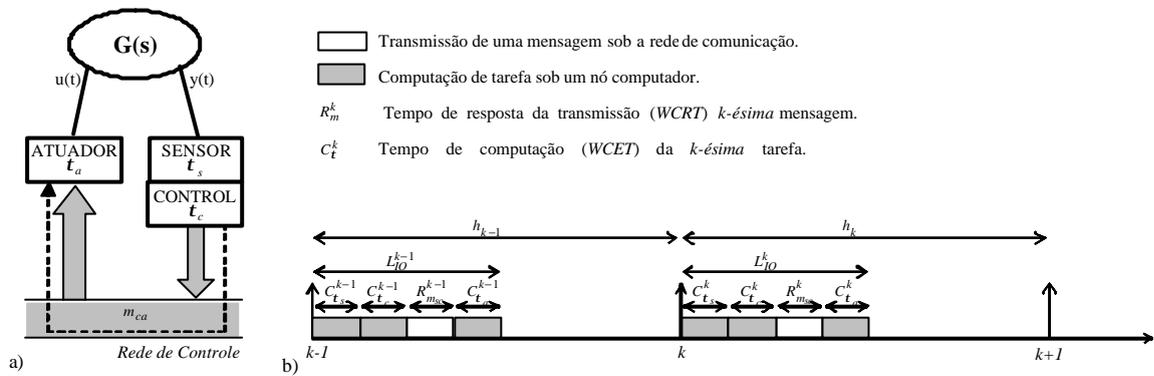


Figura 4.11: (a) Arquitetura de NCS com nó computacional integrando sensor e controlador; (b) Diagrama

temporal das tarefas e mensagens.

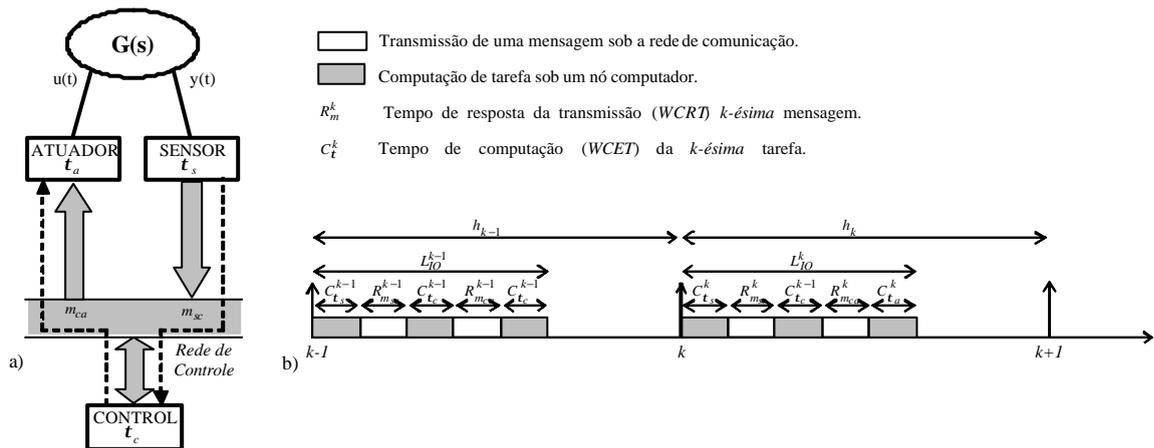


Figura 4.12: (a) Arquitetura de NCS com nós computacionais sensor, controlador e atuador; (b) Diagrama

temporal das tarefas e mensagens.

## 4.4 Estudo de Caso

Será apresentado um estudo de caso do impacto da rede de comunicação em diferentes condições de carga sob o desempenho e estabilidade de um *NCS*.

Considere uma rede de comunicação em tempo real composta por 7 *nós computacionais* de processamento. Os *nós computacionais* 1 e 2 pertencem a um *NCS* com arquitetura da Fig. 4.9 (a) e parâmetros do processo físico descrito na Tabela 4.2. Os *nós computacionais* 3 à 6 geram mensagens periódicas sob a rede e o *nó computacional* 7 gera uma carga de mensagens extra em função da utilização da largura de banda da rede desejada, que corresponde a 80%.

Considere também um servo-sistema do tipo *motor-dc*, em que o objetivo de controle é que a saída  $y(t)$  siga um sinal de referência  $y_{ref}(t)$  o mais próximo possível. O servo-sistema é representado por uma função de transferência e um controlador do tipo PD (proporcional mais derivativo), projetado e implementado na forma de um controlador discreto no tempo. Os parâmetros de projeto do controlador são apresentados na Tabela 4.2.

Tabela 4.2. Parâmetros de projeto do controlador PD

Processo $G(s)$	$w_b$ (rad/s)	Controlador PD		$h_k$ (ms)
		$Kd$	$Td$	
$G(s) = \frac{1000}{s(0,1s + 1)}$	50	3,52	0,021	1,5

A Fig. 4.13 apresenta a resposta  $y(t)$  a um degrau de entrada em malha fechada do *motor-dc*, com requisitos de resposta subamortecido.

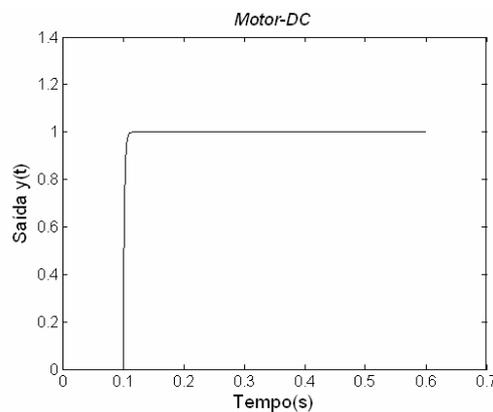


Figura 4.13: Desempenho em malha fechada do motor-dc para a resposta a um degrau

A Fig. 4.14 apresenta uma configuração de rede composta por um *NCS*, nós computacionais que geram mensagens periódicas e carga extra sob a rede de comunicação.

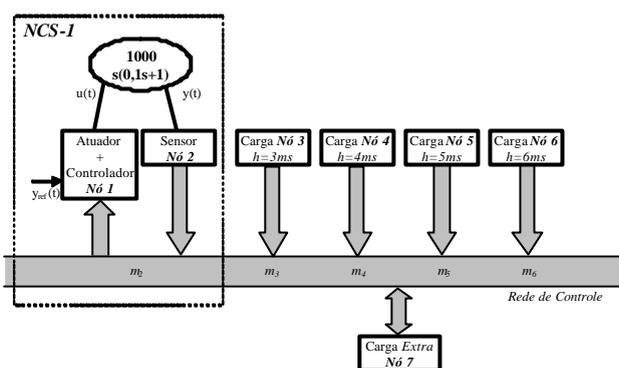


Figura 4.14: Arquitetura de rede com *NCS*, carga de mensagem periódica e carga extra.

Os fluxos de mensagens  $m_3, m_4, m_5$  e  $m_6$ , são periódicos e representam uma carga adicional sob a rede. O *nó computacional 2* lê a variável de saída  $y(t)$  da planta física e gera uma mensagem  $m_2$  contendo o valor da variável de saída que é enviada ao *nó computacional 1* que, ao recebê-lo executa o algoritmo de controle e gera um sinal de correção a ser enviado a variável de controle  $u(t)$  da planta física.

O período de amostragem da planta física é  $h = 1,5ms$  o atraso de entrada-saída dado por:

$$L_{IO} = C_{t_s} + R_{m_1} + C_{t_c} + C_{t_a} \quad (4.4)$$

Vamos considerar que o tempo de computação das tarefas de medição, execução do algoritmo de controle e atuação sejam pequenos, porém não desprezíveis,  $C_{t_s} = C_{t_c} = C_{t_a} = 100 \text{ ms}$ .

Os *nós computacionais 3 à 6* possuem tarefas ativadas periodicamente com  $h_2 = 3ms, h_3 = 4ms, h_4 = 5ms$  e  $h_5 = 6ms$  e a cada ativação geram uma mensagem sob a rede. O tempo de computação das tarefas é de  $C_{t_3} = C_{t_4} = C_{t_5} = C_{t_6} = 100 \text{ ms}$ . O *nó computacional 7* gera uma carga de mensagens sob a rede baseada na percentagem da largura de banda do canal de comunicação (80%).

As propriedades temporais das mensagens periódicas sob a rede são dadas na Tabela 4.3.

Tabela 4.3: Características dos fluxos de mensagens periódicas

<i>Mensagens</i>	$T_i$ (ms)	$p_i$	<i>Ativação</i>
$m_2$	1,5	1	Timed-driven
$m_3$	3	2	Timed-driven
$m_4$	4	3	Timed-driven
$m_5$	5	4	Timed-driven
$m_6$	6	5	Timed-driven

Para o conjunto de mensagens dados acima serão analisadas as seguintes redes de comunicação em tempo real que podem suportar respectivamente: *fluxo de mensagens periódico*, *tempo de resposta com limite superior* e *comportamento temporal previsível*: CAN, Token-Passing e TDMA. O estudo de caso específico será realizado para as três redes da seguinte forma:

- Taxa de transmissão: 500 Kbps e 1 Mbps
- Rede de comunicação com baixa e alta utilização da largura de banda (sem e com carga extra)

A partir de propriedades das redes de comunicação de tempo real, serão analisados os impactos destas sob a qualidade do controle (QoC) de um NCS.

#### 4.4.1 NCS-I sobre a Rede CAN

Na rede CAN a atribuição de prioridade aos diferentes fluxos de mensagens é baseada no algoritmo de prioridade fixa e não preemptivo *RM* (*Rate Monotonic*), as mensagens com menor período de ativação têm maior prioridade. Será considerado que um frame CAN tenha um comprimento máximo de até 131 bits (147 bits para o CAN 2.0) e pode levar até 64 bits de informação. O cálculo de tempo resposta  $R_i$  em redes CAN pode ser realizado a partir de Eq. (3.3) e (3.4) e visto na Tabela 4.4. Vamos ainda considerar que todos os *frames* gerados sejam completos.

Tabela 4.4: Características das mensagens sob a rede CAN

Rede CAN											
Mensagens	$T_i$ (h)	500 Kbps					1 Mbps				
		$U_i(\%)$	Sem carga		Com carga		$U_i(\%)$	Sem carga		Com carga	
			$C_i(ms)$	$R_i(ms)$	$C_i(ms)$	$R_i(ms)$		$C_i(ms)$	$R_i(ms)$	$C_i(ms)$	$R_i(ms)$
m <sub>2</sub>	1,5	1,7	0,262	0,524	0,262	0,524	0,85	0,131	0,262	0,131	0,262
m <sub>3</sub>	2	1,3	0,262	0,786	0,262	0,786	0,65	0,131	0,393	0,131	0,393
m <sub>4</sub>	3	0,9	0,262	1,048	0,262	1,048	0,45	0,131	0,524	0,131	0,524
m <sub>5</sub>	4	0,7	0,262	1,031	0,262	1,031	0,35	0,131	0,655	0,131	0,655
m <sub>6</sub>	5	0,5	0,262	1,572	0,262	1,572	0,25	0,131	0,786	0,131	0,786
Carga		80	-	-	-	-	80	-	-	-	-

### Cálculo de Atrasos para o NCS-1

O desempenho e estabilidade de um NCS-1 podem ser verificados a partir do *atraso de entrada-saída* e *período de amostragem*, conforme apresentados nos gráficos do espectro de qualidade do controle. A Tabela 4.5 apresenta os parâmetros relativos aos *atrasos de entrada-saída*.

Tabela 4.5: Atraso de entrada-saída do NCS-1 a partir dos atrasos de execução de tarefas e transmissão de mensagens

$h$ (ms)	500 Kbps				1 Mbps			
	$C_{t_s}$	$R_{m_{sc}}$	$C_{t_s} + C_{t_a}$	$L_{IO}$	$C_{t_s}$	$R_{m_{sc}}$	$C_{t_s} + C_{t_a}$	$L_{IO}$
1,5	0,100	0,524	0,200	<b>0,824</b>	0,100	0,262	0,200	<b>0,562</b>

O critério de desempenho quadrático pode ser utilizado para avaliar o custo da qualidade de controle. Pode-se verificar que a taxa de transmissão de uma rede de comunicação está diretamente relacionada ao *atraso de entrada-saída*, desde que os tempos de computação das tarefas sejam desconsiderados.

### Simulações

A Fig. 4.15 (a) e (b) apresenta o gráfico relativo à resposta do sistema de controle para a uma rede CAN a 500 Kbps, sem carga extra e com carga extra.

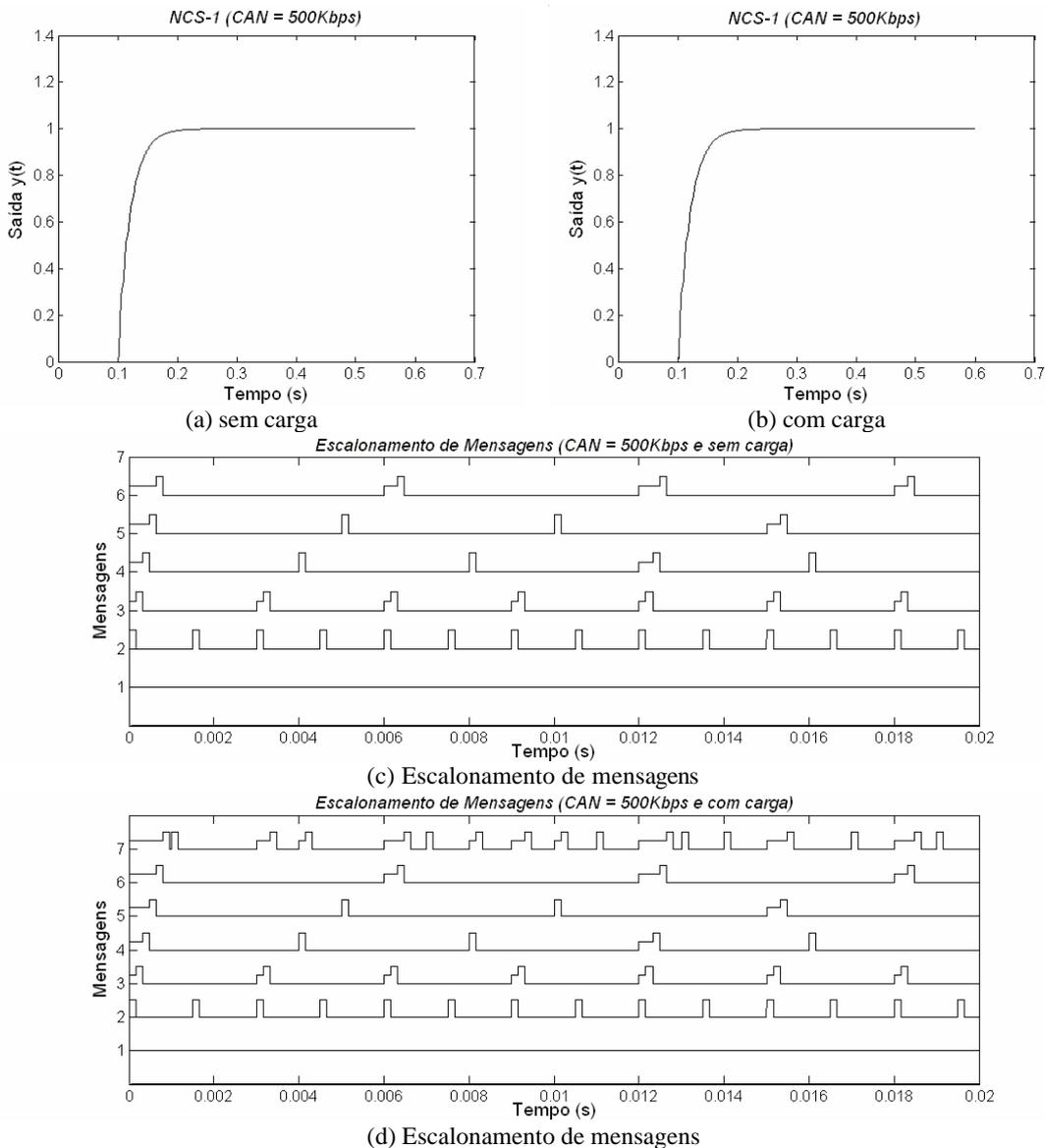


Figura 4.15: Resposta do NCS-1 e escalonamento de mensagens sob uma rede CAN a 500 Kbps.

Pode-se verificar que o desempenho de resposta do *NCS-1* para ambos os casos não sofrerá degradação, isto porque o fluxo de mensagens relacionado com a malha de controle ter maior prioridade sob as outras mensagens, conforme pode ser visto em Fig. 4.15 (c) e (d).

Na Fig. 4.15 (d) a interferência de mensagens de carga extra sob o fluxo de mensagens da malha de controle e o bloqueio de mensagens de baixa prioridade deve ser considerada no teste de escalonabilidade; porém, mesmo sob estas condições, o

comportamento do tempo de resposta das instâncias da mensagem gerada pelo *nó computacional 2* são os mesmos, conforme Fig. 4.15 (c). Isto é uma característica importante da rede CAN, pois para um determinado fluxo de mensagens é possível determinar o *WCRT* e estes se mantêm para quaisquer condições de carga sob a rede.

A mesma análise pode ser realizada para uma rede CAN a 1 Mbps. A Fig. 4.16 (a) e (b) apresenta os gráficos relativos à resposta do sistema *NCS-1* e Fig. 4.16 (c) e (d) escalonamento de mensagens.

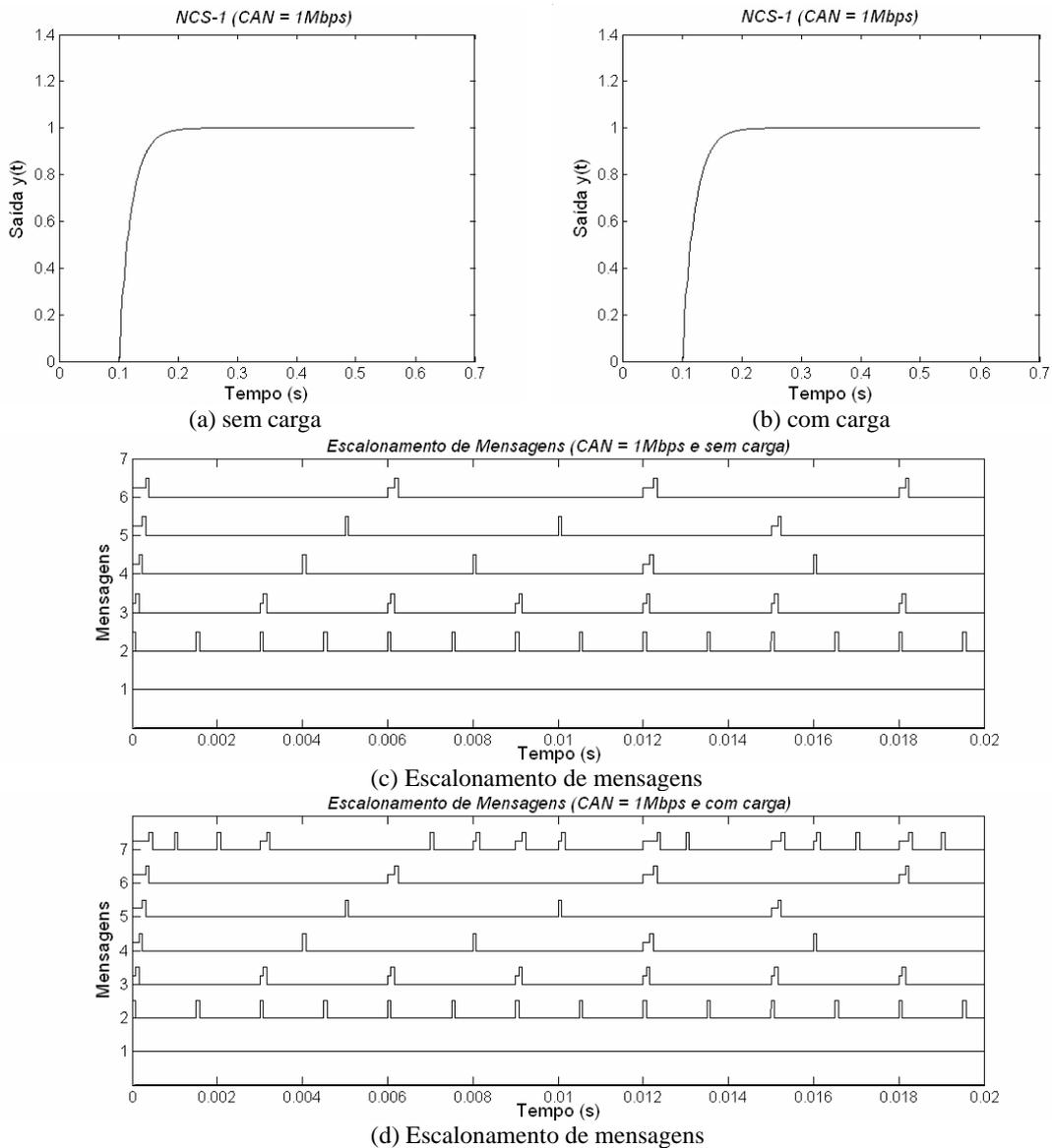


Figura 4.16: Resposta do NCS-1 e escalonamento de mensagens sob uma rede CAN a 1 Mbps.

Note, na Fig. 3.13 que o desempenho da resposta da malha de controle é melhorado, em função da redução do atraso de entrada-saída, que está diretamente relacionada com a taxa de transmissão da rede. A melhoria do desempenho se deve a um aumento da taxa de transmissão da rede CAN em análise.

Para malhas de controle insensíveis ao *jitter*, a rede CAN é considerada satisfatória para suportar *NCS*, porque a análise e projeto do sistema de controle podem ser realizados em função do *WCRT* das mensagens sob a rede CAN.

#### 4.4.2 *NCS-I* sobre a Rede Token-Passing

Na rede Token-passing, os *nós computacionais* recebem uma prioridade para acessar o meio de transmissão e com isto as mensagens a serem transmitidas, além de ter de ser escalonadas numa fila interna, deverão aguardar o exato momento em que o bastão estiver no *nó computacional* dando condições a este para transmitir uma mensagem pendente.

O comprimento máximo de um *frame* Token-passing é considerado por 152 bits e pode levar uma mensagem de até 64 bits como na rede CAN. O cálculo do tempo de resposta em redes Token-passing  $R_i$  pode ser realizado a partir de Eq. (3.10) e (3.11) e visto na Tabela 4.6. Vamos ainda considerar que todos os *frames* gerados sejam completos.

Tabela 4.6: Características das mensagens na rede Token-passing

Rede Token-passing											
Mensagens	$T_i$ (h)	500 Kbps						1 Mbps			
		$U_i(\%)$	Sem carga		Com carga		$U_i(\%)$	Sem carga		Com carga	
			$C_f(ms)$	$R_f(ms)$	$C_f(ms)$	$R_f(ms)$		$C_f(ms)$	$R_f(ms)$	$C_f(ms)$	$R_f(ms)$
$m_2$	1,5	2,0	0,304	1,904	0,304	1,904	1,0	0,152	0,952	0,152	0,952
$m_3$	2	1,5	0,304	1,904	0,304	1,904	0,75	0,152	0,952	0,152	0,952
$m_4$	3	1,0	0,304	1,904	0,304	1,904	0,5	0,152	0,952	0,152	0,952
$m_5$	4	0,8	0,304	1,904	0,304	1,904	0,4	0,152	0,952	0,152	0,952
$m_6$	5	0,6	0,304	1,904	0,304	1,904	0,3	0,152	0,952	0,152	0,952
Carga		80	-	-	-	-	80	-	-	-	-

#### Cálculo de Atrasos para o *NCS-I*

O desempenho e estabilidade de um *NCS* podem ser verificados, a partir do *atraso de entrada-saída* e *período de amostragem* conforme apresentados nos gráficos do

espectro de qualidade do controle. A Tabela 4.7 apresenta os parâmetros relativos aos *atrasos de entrada-saída*.

Tabela 4.7: Atraso de entrada-saída do NCS-1 a partir dos atrasos de execução tarefas e transmissão de mensagens

$h$ (ms)	500 Kbps				1 Mbps			
	$C_{t_s}$	$R_{m_{sc}}$	$C_{t_s} + C_{t_a}$	$L_{IO}$	$C_{t_s}$	$R_{m_{sc}}$	$C_{t_s} + C_{t_a}$	$L_{IO}$
1,5	0,100	1,904	0,200	<b>2,204</b>	0,100	0,952	0,200	<b>1,262</b>

### Simulações

A Fig. 4.17 (a) e (b) apresenta o gráfico relativo à resposta do sistema de controle para uma rede Token-passing a 500 Kbps, sem ou com carga extra adicional de mensagens.

O desempenho da resposta de controle para o *NCS-1* não pode ser visto em 4.17 (a) e (b). Porém os tempos de resposta das mensagens podem ser verificados em Fig. 4.17 (c) e (d).

Na Fig. 4.17 (c) o bastão irá visitar apenas *5 nós computacionais* sob a rede e na Fig. 4.17 (d) pode-se notar que o bastão irá visitar um *nó computacional* a mais, que corresponde ao gerador de mensagem com carga extra, que conseqüentemente fará com que o atraso de entrada-saída seja elevado. A partir de um gráfico do custo de qualidade de controle é possível verificarmos o impacto dos atrasos das mensagens sob a qualidade do controle.

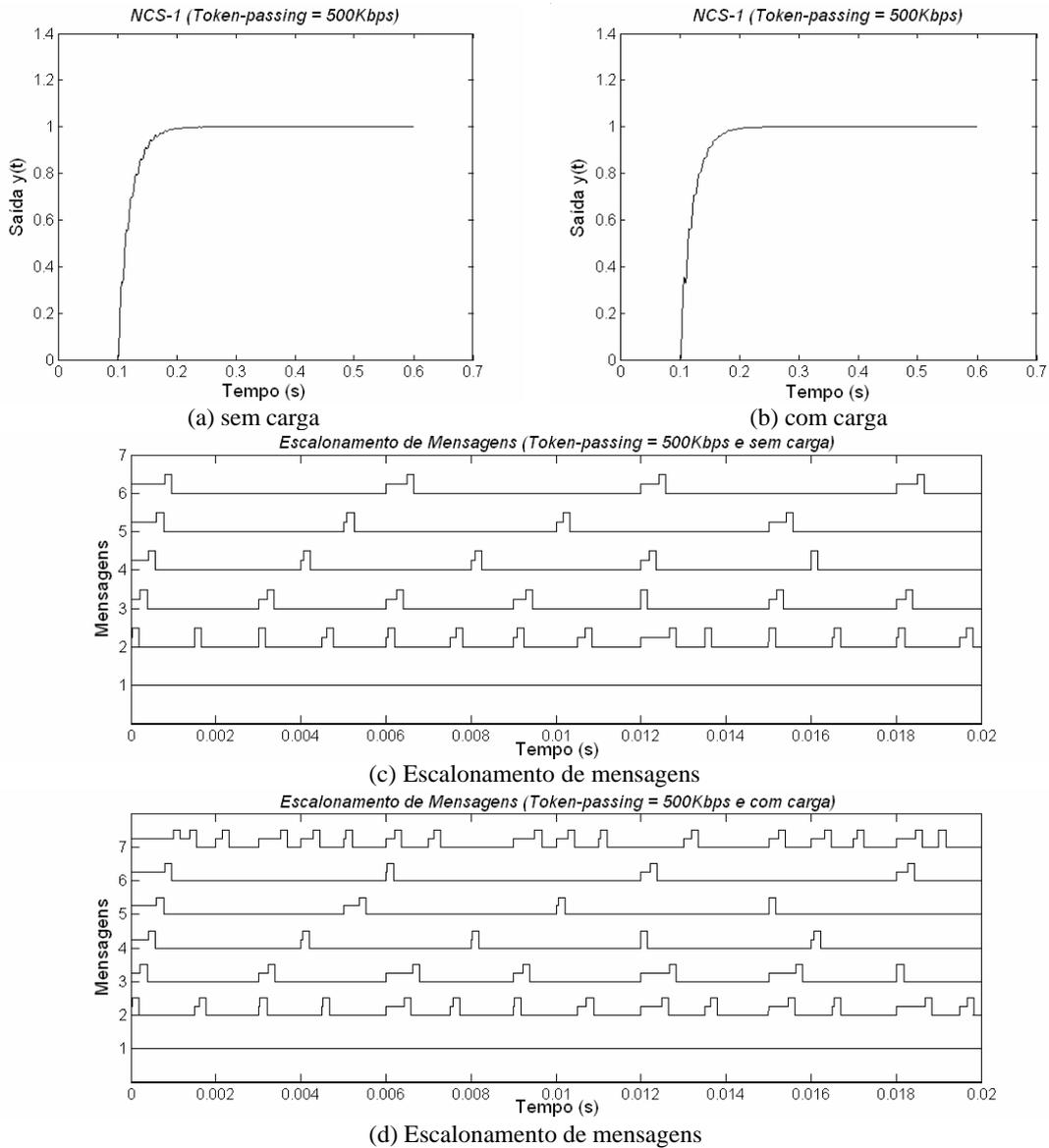


Figura 4.17: Resposta do NCS-1 e escalonamento de mensagens sob uma rede Token-passing a 500 Kbps

Ao comparar-se a rede Token-passing com as mesmas condições da rede CAN na Fig. 4.15 (b), conclui-se que esta última é a mais adequada para suportar aplicações de controle baseado no tempo de resposta para mensagens (*WCRT*), mesmo na presença de carga-extra, porque o mecanismo de prioridade é definido sob as mensagens.

Uma característica importante da rede Token-passing é que se um *nó computacional* estiver de posse do bastão, ao iniciar a transmissão de uma mensagem, esta será realizada totalmente.

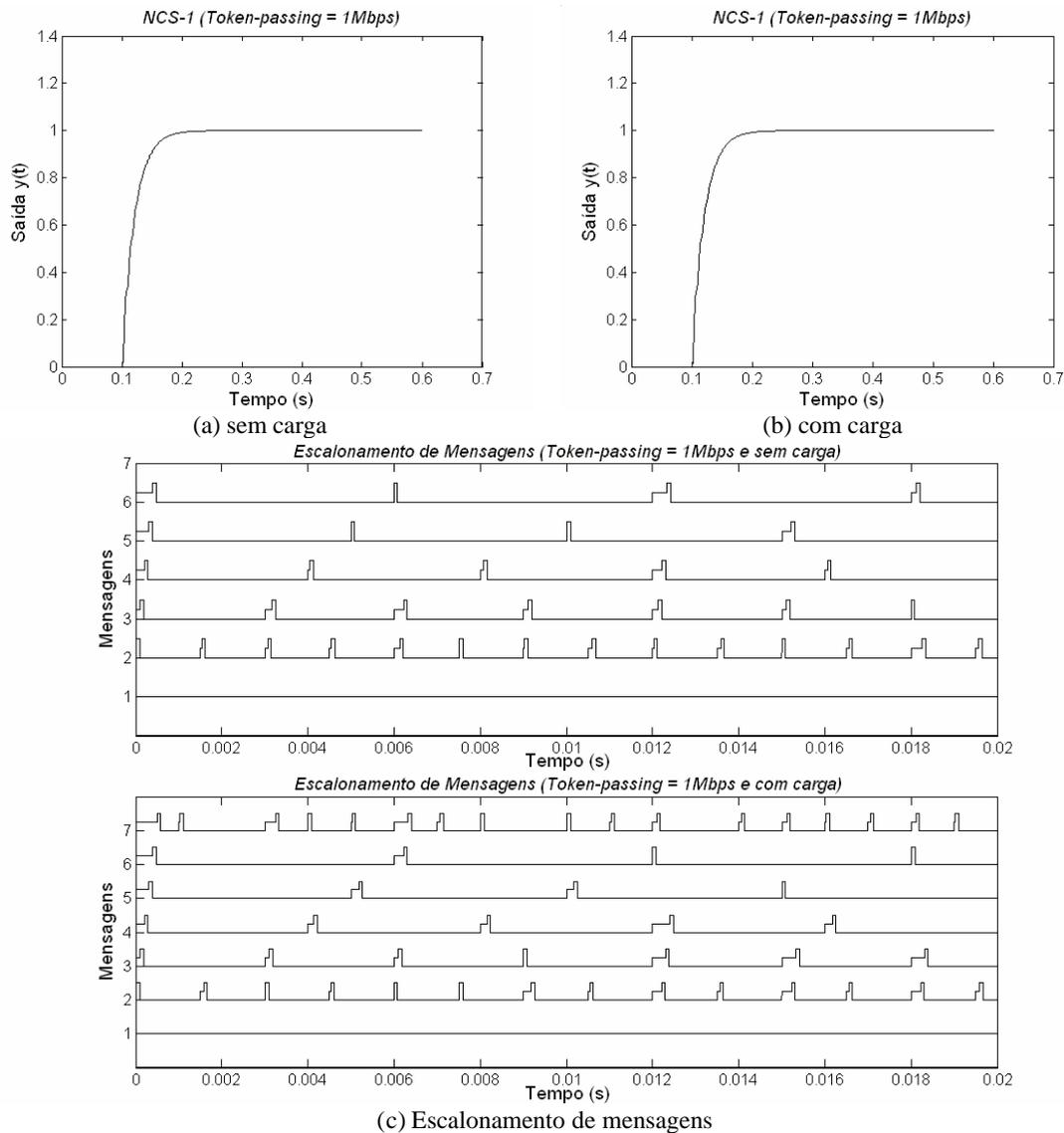


Figura 4.18: Resposta do NCS-1 e escalonamento de mensagens sob uma rede Token-passing a 1 Mbps

Para a rede Token-passing a 1 Mbps, o *atraso de entrada-saída* do NCS-1 é reduzido, fazendo com que o desempenho e estabilidade do sistema de controle sejam melhorados.

O desempenho e estabilidade para um NCS-1 sob a rede CAN é insensível a variações de carga sob a rede de comunicação, em função da prioridade sob mensagens. Sob a rede Token-passing o desempenho e estabilidade do NCS-1, podem ser comprometidos em função da quantidade de *nós computacionais* e quantidade de

mensagens sob a rede de comunicação, porque estes influem diretamente no tempo de resposta das mensagens no pior caso (*WCRT*), de um *NCS*.

#### 4.4.3 *NCS-1* sobre a Rede TDMA

Na rede TDMA, os *nós computacionais* têm acesso ao canal de comunicação em intervalos de tempo definidos ciclicamente escalonados no tempo. As mensagens a serem transmitidas, além de ter de ser escalonadas numa fila interna, deverão aguardar pelo *slot-time* relativo ao *nó computacional*.

O formato de um *frame* com comprimento máximo de até 85 bits do protocolo TTP (*Time-Trigger Protocol*), com uma capacidade de até 8 bytes do campo de dados. O cálculo de tempo resposta em redes TDMA  $R_i$  pode ser realizado, a partir de Eq. (3.10) e (3.11) e visto na Tabela 4.8.

Desta forma, podemos verificar que na Fig. 3.16, a instabilidade das malhas de controle se deve a um atraso de controle ocorrido, em função da mensagem do nó computacional sensor exigir uma quantidade maior de *slot-time*. A instabilidade é verificada para o sistema com e sem carga-extra.

Tabela 4.8: Características das mensagens na rede TDMA

Rede TDMA											
Mensagens	$T_i$ ( $h_k$ )	500 Kbps						1 Mbps			
		$U_i(\%)$	Sem carga		Com carga		$U_i(\%)$	Sem carga		Com carga	
			$C_i(ms)$	$R_i(ms)$	$C_i(ms)$	$R_i(ms)$		$C_i(ms)$	$R_i(ms)$	$C_i(ms)$	$R_i(ms)$
$m_2$	1,5	1,10	0,17	0,85	0,17	0,85	0,55	0,085	0,425	0,085	0,425
$m_3$	2	0,85	0,17	0,85	0,17	0,85	0,40	0,085	0,425	0,085	0,425
$m_4$	3	0,57	0,17	0,85	0,17	0,85	0,28	0,085	0,425	0,085	0,425
$m_5$	4	0,42	0,17	0,85	0,17	0,85	0,21	0,085	0,425	0,085	0,425
$m_6$	5	0,34	0,17	0,85	0,17	0,85	0,17	0,085	0,425	0,085	0,425
Carga		80	-	-	-	-	80	-	-	-	-

#### Cálculo de Atrasos para o TDMA

O desempenho e estabilidade de um *NCS* podem ser verificados, a partir do *atraso de entrada-saída* e *período de amostragem* conforme apresentados nos gráficos do espectro de qualidade do controle. A Tabela 4.9 apresenta os parâmetros relativos aos *atrasos de entrada-saída*.

Tabela 4.9: Atraso de entrada-saída do NCS a partir dos atrasos de tarefas e mensagens

$h_k$ (ms)	500 Kbps				1 Mbps			
	$C_{t_s}$	$R_{m_{sc}}$	$C_{t_s} + C_{t_a}$	$L_{IO}$	$C_{t_s}$	$R_{m_{sc}}$	$C_{t_s} + C_{t_a}$	$L_{IO}$
1,5	0,100	0,850	0,200	<b>1,150</b>	0,100	0,425	0,200	<b>0,725</b>

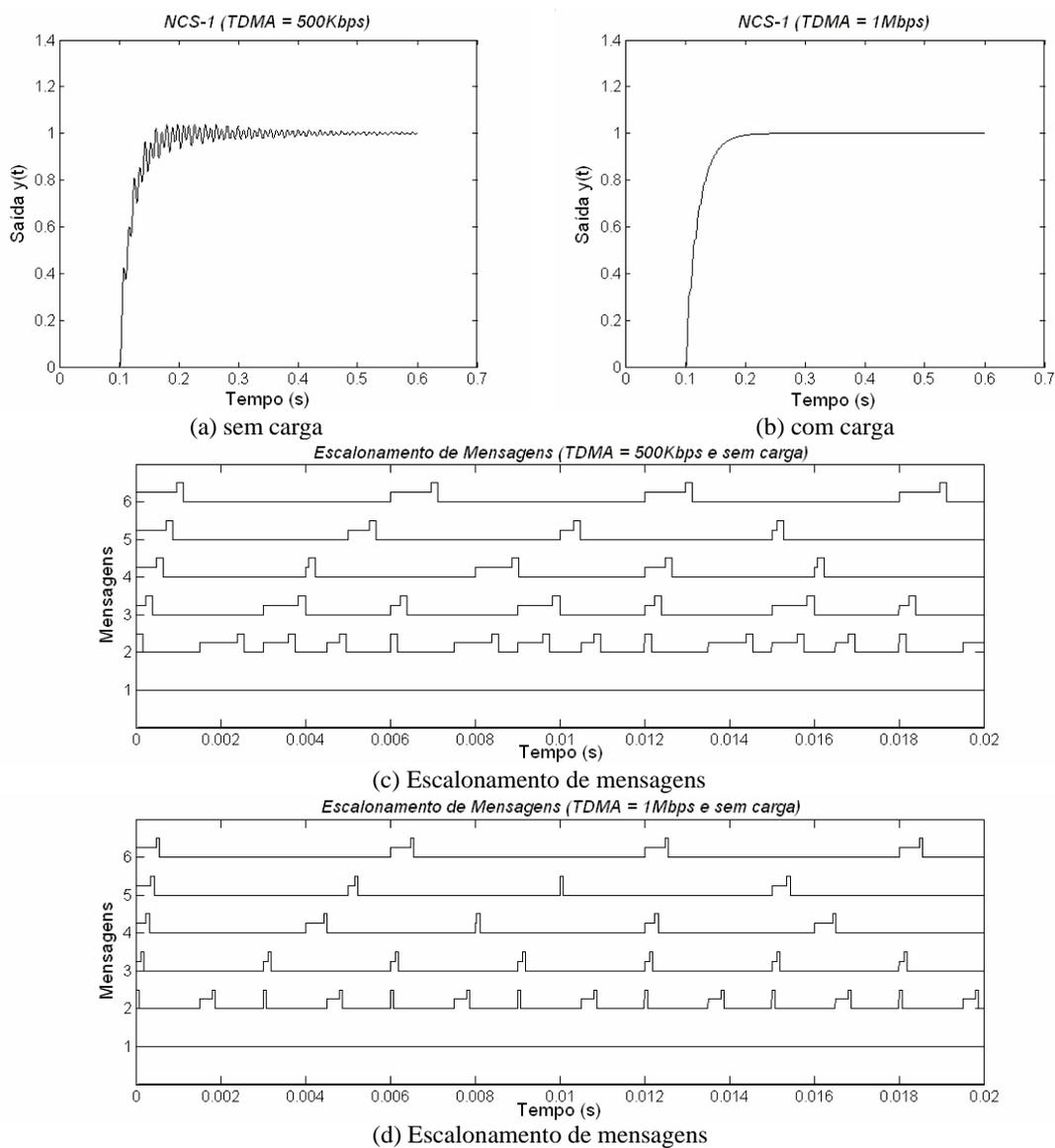


Figura 4.19: Resposta do NCS-1 e escalonamento de mensagens sob uma rede TDMA a 500Kbps e 1Mbps, sem carga extra e apenas cinco nós computacionais de acesso ao meio.

A Fig. 4.19 (a) apresenta o desempenho de controle oscilatório inferior que a Fig. 4.19 (b) para uma resposta a degrau, considerando apenas 5 *slot-time*. Isto se deve ao fato

de que o WCRT das mensagens para o *NCS-1*, conforme demonstrado na Fig. 4.19 (c) ser maior que na Fig. 4.19 (d).

A Fig. 4.19 apresenta a situação de uma rede TDMA a 500Kbps e 1Mbps, sem carga extra e com apenas 5 nós computacionais sob a rede de comunicação.

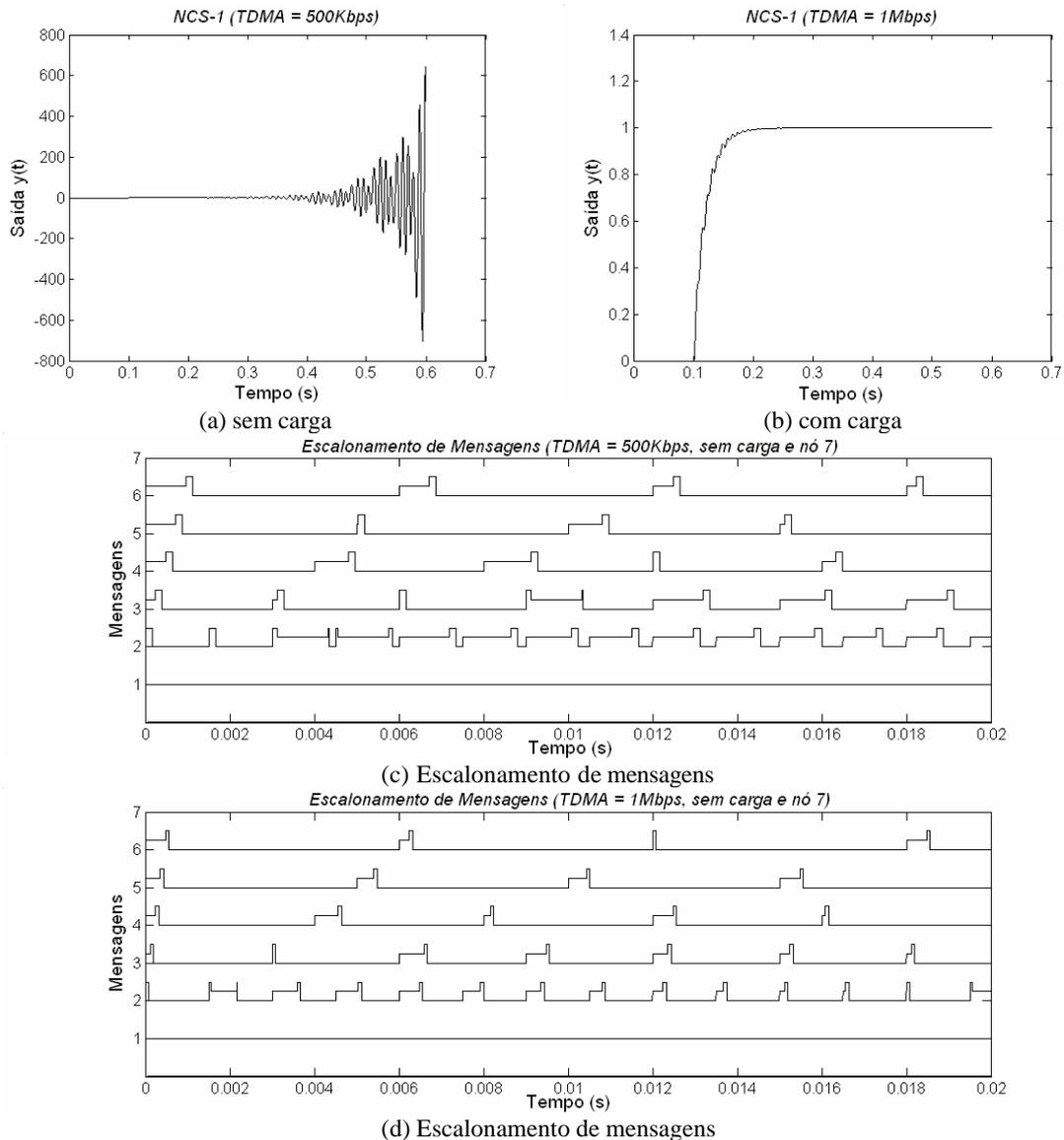


Figura 4.20: Resposta do NCS e escalonamento de mensagens sob uma rede TDMA a 500Kbps e 1Mbps, com carga extra e apenas cinco nós computacionais de acesso.

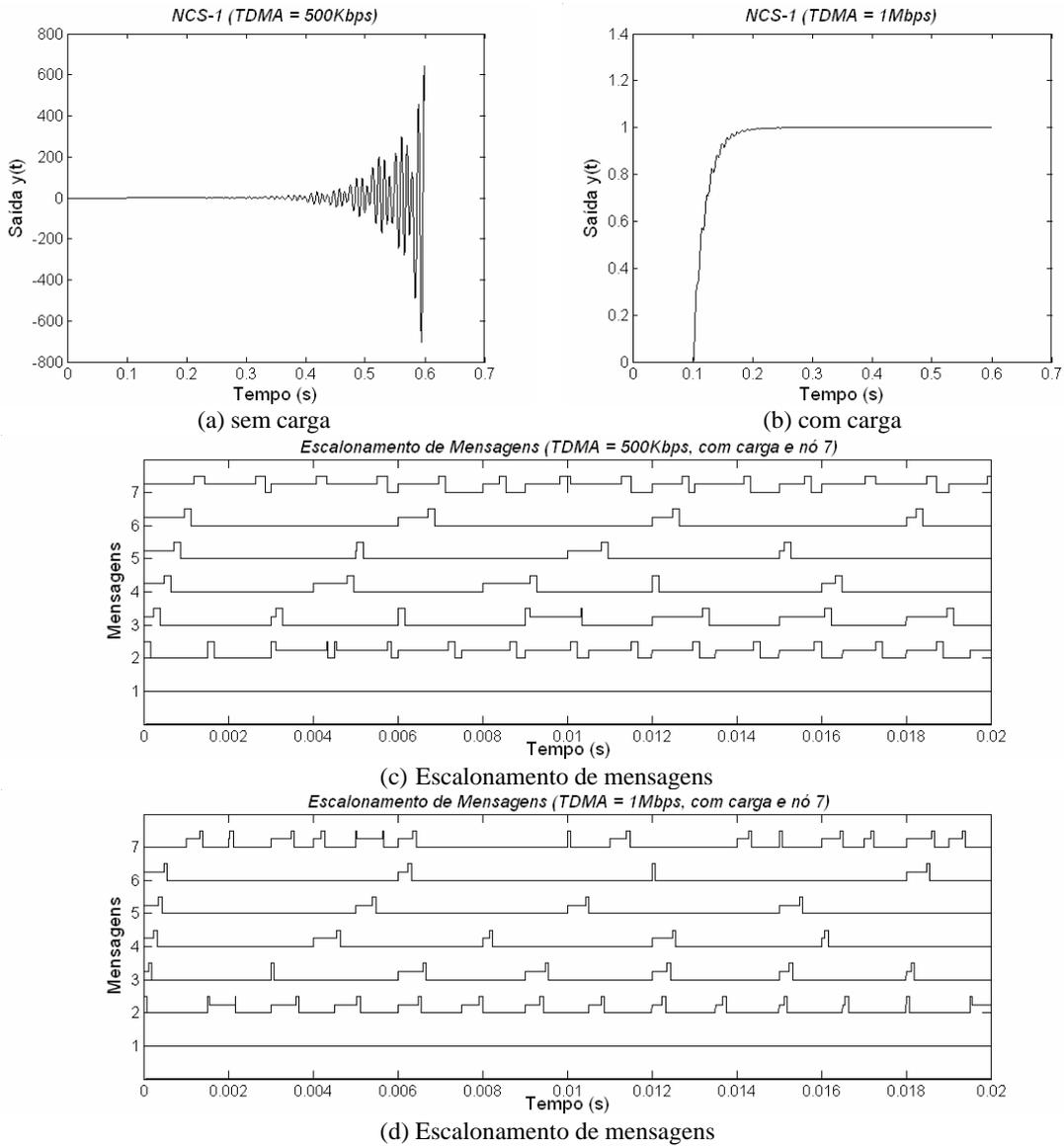


Figura 4.21: Resposta do NCS e escalonamento de mensagens sob uma rede TDMA a 500Kbps e 1Mbps, com carga extra e seis nós computacionais de acesso ao meio.

Nas redes de passagem de bastão, quando um *nó computacional* está de posse de um bastão, este realiza a transmissão da mensagem totalmente. Já nas redes do tipo TDMA, um *nó computacional* só poderá transmitir uma mensagem, quando for alocado o seu *slot-time*, e o tempo de transmissão da mensagem for inferior ao *slot-time*, mas para mensagens que necessitam de um tempo de transmissão maior que o *slot-time* designado

ao *nó computacional*, esta necessitará de uma quantidade maior de *slot-time* dos ciclos subsequentes.

A Fig. 4.20 apresenta a situação de uma rede TDMA a 500Kbps e 1Mbps com carga extra e com apenas 5 *nós computacionais* sob a rede de comunicação. Nas figuras 4.20 e 4.21 pode-se verificar que o desempenho de controle são idênticos, porém os *WCRT* são diferentes em função da carga sob a rede, taxa de transmissão e números de nós computacionais sob a rede de comunicação. A Fig. 4.21 apresenta a situação de uma rede TDMA a 500Kbps e 1Mbps, com carga extra e 6 *nós computacionais* sob a rede.

## 4.5 Conclusão

Em sistemas de controle por computador, afirmou-se que numa implementação real o sistema computacional pode induzir atrasos e variação de atrasos que influenciarão diretamente no desempenho e estabilidade do sistema de controle. Relativo à seleção do *período de amostragem* e *atraso de entrada-saída* foram apresentados dois métodos, a *regra de thumb* e o *critério de desempenho quadrático*.

Um *NCS* consiste em um sistema de controle por computador realimentado onde a malha de controle é fechada sob uma rede de comunicação, quando foram estudadas três arquiteturas de *NCS* para sistemas do tipo *SISO* (do inglês: "*Single Input and Single Output*"), e o comportamento temporal das tarefas e mensagens para um *NCS*.

Desenvolveu-se um estudo de caso com o projeto do controlador para o motor-dc e inserido na forma de *NCS* em uma rede de comunicação com nós computacionais adicionais que geram uma carga de mensagens sob a rede. Avaliou-se o impacto da comunicação sob o desempenho e estabilidade do *NCS* com as redes de comunicação em tempo real CAN, TDMA, e Token-passing.

Constatou-se que a rede CAN é a mais adequada para suportar aplicações de controle com carga extra variável sob as mesmas condições de taxa de transmissão e carga adicional também submetida à rede de comunicação. Notou-se ainda que a taxa de transmissão tem uma relação direta com o atraso *de entrada-saída*, sendo assim um fator determinante para a garantia de desempenho e estabilidade *NCS*.

# 5 Projeto de *NCS* Baseado no Tempo de Resposta de Mensagens

## 5.1 Introdução

Em aplicações de sistema de controle via redes, uma rede de comunicação de dados em tempo real induz atraso e variação de atraso na transmissão de mensagens que diretamente comprometem o *período de amostragem* e *atraso de entrada-saída* nas malhas de controle.

O projeto de um *NCS* pode ser realizado baseado no cálculo de *WCRT* (*Worst Case Response Time*) de mensagens, quando uma malha de controle não é sensível a *variação de entrada-saída* (*jitter de entrada-saída*). Pretende-se apresentar um método de projeto para *NCS* baseado no cálculo de *WCRT* para mensagens.

Em sistemas de *NCS* sensíveis ao *jitter* de entrada-saída, a realização do projeto em função do *WCRT* não é suficiente para garantir o desempenho e estabilidade, porque as várias instâncias de uma mensagem do *NCS* possuem tempo de respostas diferentes. Uma abordagem de projeto que considere em separado os tempos de resposta das diversas instâncias de uma mensagem seria mais adequado.

Com base na teoria de controle por computador, o modelo de controle discreto no tempo padrão e métodos para análise e projeto para *NCS*, serão apresentados através de exemplos ilustrativos. Primeiro será descrito o método de controle tradicional com atraso constante; em seguida, uma abordagem de projeto baseado no controlador discreto *LQG* (*Linear Quadratic Gaussian*) e, finalmente, demonstraremos um exemplo de projeto para *NCS*.

## 5.2 Método de Controle Tradicional com Atraso Constante

Nesta seção, será apresentado um modelo de controle discreto no tempo, a partir de modelos de sistemas dinâmicos contínuos no tempo com atraso de tempo constante, que será a base para a formulação do método de projeto para NCS a ser proposto a seguir.

Considere um processo modelado por equações de espaço de estado contínuo e invariante no tempo, com um *atraso de controle (atraso de entrada-saída)*  $L_{IO}$  descrito pelas equações em (5.1) como descrito em Åstrom and Wittnemark, 1997 e assumindo  $L_{IO} < h$ .

$$\begin{aligned}\frac{dx(t)}{dt} &= Ax(t) + Bu(t - L_{IO}) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\tag{5.1}$$

As transformações do modelo contínuo no tempo em (5.1) para um modelo discreto no tempo são dados por (5.2), (5.3) e (5.4).

$$\Phi(h) = e^{Ah}\tag{5.2}$$

$$\Gamma_0(h, L_{IO}) = \int_0^{h-L_{IO}} e^{As} ds B\tag{5.3}$$

$$\Gamma_1(h, L_{IO}) = e^{A(h-L_{IO})} \int_0^{L_{IO}} e^{As} ds B\tag{5.4}$$

O sistema (5.1) na forma discretizada do tempo é dado por (5.5) e (5.6).

$$x(kh + h) = \Phi(h)x(kh) + \Gamma_0(h, L_{IO})u(kh) + \Gamma_1(h, L_{IO})u(kh - h)\tag{5.5}$$

$$y(kh) = Cx(kh) + Du(kh)\tag{5.6}$$

As matrizes  $\Phi(h)$ ,  $\Gamma_0(h, L_{IO})$  e  $\Gamma_1(h, L_{IO})$  são obtidas a partir de (5.1) para cada execução em malha fechada de cada *período de amostragem*  $h$  e *atraso de entrada-saída*  $L_{IO}$ , sendo que o controlador por realimentação de estado  $-L(h, L_{IO})$  é obtido para cada execução em malha fechada, usando um método de projeto do controlador de tempo discreto clássico tal como alocação de pólos ou otimização.

$$u(kh) = -L(h, L_{IO})x(kh)\tag{5.7}$$

O sistema em malha fechada dado por (5.5), (5.6) e (5.7) depende do *período de amostragem* e *atraso de entrada-saída*.

### 5.3 Projeto de NCS Considerando o WCRT de Mensagens

Em sistemas de controle via redes (NCS) é possível derivar a mesma análise do método de projeto do controle tradicional com atraso constante. O projeto do controlador é realizado a partir de uma especificação de controle, sendo o *período de amostragem* ( $h$ ) e *atraso de entrada-saída* ( $L_{IO}$ ) definidos como parâmetros de tempo do controlador de interesse para análise e projeto.

Considere o projeto de um controlador em tempo discreto com parâmetros de *período de amostragem* ( $h$ ) e *atraso de entrada-saída* ( $L_{IO}$ ). A partir do tempo de resposta das mensagens no pior caso (WCRT - *Worst Case Response Time*) sob a rede de comunicação, é possível determinar o *atraso de entrada-saída* para um NCS, conforme apresentado no Capítulo 3.

Utiliza-se como exemplo de projeto um controlador do tipo LQG (*Linear Quadratic Gaussian*) para NCS. A Fig. 5.1 apresenta uma arquitetura de controle LQG para um sistema na forma de espaço de estados e função de transferência.

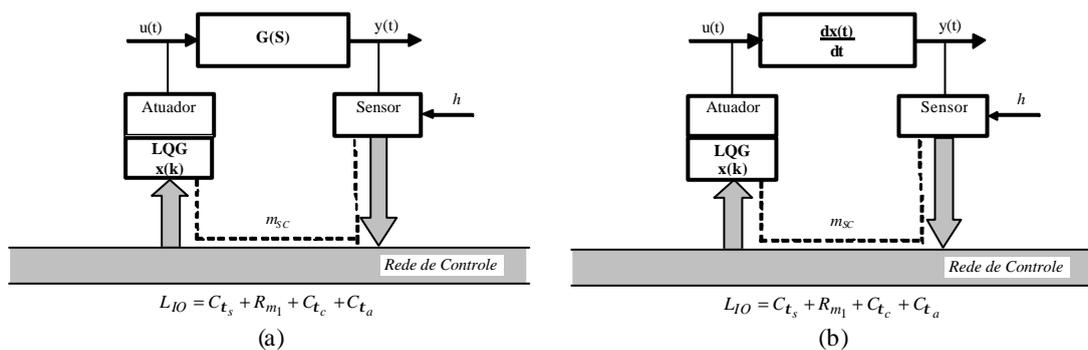


Figura 5.1: Arquiteturas de NCS com implementação de controlador LQG.

O método de projeto do controlador LQG para sistemas de controle por computador, pode ser visto em (Astrom and Witnemark, 1997).

A seguir, apresenta-se o projeto de um controlador *LQG* clássico, para uma planta física contínua no tempo com uma função-custo de tempo contínuo, assumindo um período de amostragem constante  $h$  e atraso de entrada-saída  $L_{IO}$ . Este projeto realiza-se a partir do modelo do sistema físico na forma de espaço de estados ou função de transferência.

Um sistema na forma de espaço de estados é descrito por (5.8). Sendo  $L_{IO}$  é o atraso de entrada-saída,  $v(t)$  é um processo de ruído branco com média zero e covariância  $R_1$ , e  $e(t)$  corresponde a um processo de ruído branco com media zero e covariância  $R_2$ . Os processos de ruído  $v(t)$  e  $e(t)$  assumem-se independentes. Os instantes de amostragem são dados por  $t_k = kh$ .

$$\begin{aligned}\frac{dx(t)}{dt} &= Ax(t) + Bu(t - L_{IO}) + v(t) \\ y(t_k) &= Cx(t_k) + e(t_k)\end{aligned}\quad (5.8)$$

A função-custo a ser minimizada para o controlador é especificada em (5.9), onde o termo  $Q$  é uma matriz semi-definida positiva.

$$J = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \begin{pmatrix} x(t) \\ u(t) \end{pmatrix}^T Q \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} dt \quad (5.9)$$

O sistema na forma de função de transferência é descrito por (5.10), com  $v(t)$  é um ruído branco com média zero e covariância  $R_1$ , e  $e(t)$  corresponde a um ruído branco com media zero e covariância  $R_2$ .

$$\begin{aligned}y^0(t) &= G(p)(u(t - L_{IO}) + v(t)) \\ y(t_k) &= y^0(t_k) + e(t_k)\end{aligned}\quad (5.10)$$

A função-custo a ser minimizada para o controlador é especificada em (5.11), onde  $Q$  é uma matriz semi-definida positiva.

$$J = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \begin{pmatrix} y^0(t) \\ u(t) \end{pmatrix}^T Q \begin{pmatrix} y^0(t) \\ u(t) \end{pmatrix} dt \quad (5.11)$$

O controlador resultante do projeto LQG tem a seguinte forma.

$$\begin{aligned}u(k) &= -L\hat{x}_e(k|k) \\ \hat{x}_e(k|k) &= \hat{x}_e(k|k-1) + K_f(y(k) - C_e\hat{x}_e(k|k-1)) \\ \hat{x}_e(k+1|k) &= \Phi_e\hat{x}_e(k|k-1) + \Gamma_e u(k) + K(y(k) - C_e\hat{x}_e(k|k-1))\end{aligned}\quad (5.12)$$

sendo:  $\hat{x}_c(k) = \begin{pmatrix} \hat{x}(k) \\ u(k-1) \end{pmatrix}$

## 5.4 Exemplo 5.1

Como exemplo de aplicação do projeto *LQG* para *NCS*, será considerado o problema de controle do pêndulo invertido apresentado por (Cervin, 2003), na forma de escalonamento em tempo real e controle integrado.

Considere um conjunto de três pêndulos invertidos descritos por uma função de transferência linear,

$$G(s) = \frac{w_0^2}{s^2 - w_0^2} \quad (5.13)$$

sendo  $w_0$  a frequência natural de cada pêndulo, dadas respectivamente por  $w_0 = 9.9, 7.0, 5.7$  rad/s. Considera-se que o processo é perturbado por um ruído branco de tempo contínuo com uma variância unitária e que a medição é perturbada por um ruído de tempo discreto com variância unitária. O controlador *LQG* é projetado para uma especificação de controle dada que, a partir de uma condição inicial  $x(0)$ , um sinal de controle  $u(t)$  seja realizado para que o pêndulo retorne a uma posição estável e minimize a seguinte função-custo:

$$J = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \begin{pmatrix} y(t) \\ u(t) \end{pmatrix}^T Q \begin{pmatrix} y(t) \\ u(t) \end{pmatrix} dt \quad (5.14)$$

Uma arquitetura de *NCS* é apresentada com os respectivos fluxos de mensagens sob a rede.

### 5.4.1 Arquitetura de *NCS*'s

A arquitetura de *NCS* assumida para os três pêndulos consiste em um *nó computacional* sensor e outro *nó computacional* controlador *LQG* e atuador, consistindo em um fluxo de mensagens gerado na rede de controle. A Fig. 5.2 apresenta a arquitetura em rede para os pêndulos.

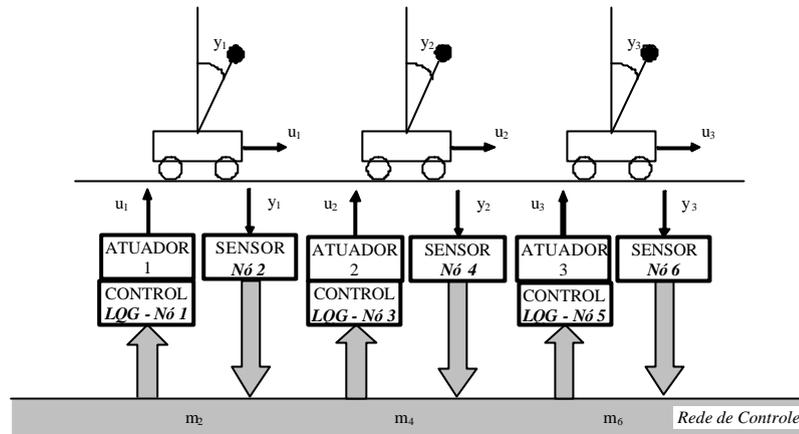


Figura 5.2: Configuração de três pêndulos invertidos na forma de NCS.

Os fluxos de mensagens  $m_2$ ,  $m_4$ , e  $m_6$ , são periódicos com as propriedades temporais das mensagens de controle, que podem ser observados na Tabela 5.1, assumindo que uma rede CAN a 250Kbps será usada com atribuição de prioridades baseada no algoritmo RM (*Rate Monotonic*).

Tabela 5.1. Cálculo do tempo de resposta na rede CAN.

Rede CAN - 250Kbit/s					
Mensagens	Propriedades Temporais do Fluxo de Mensagens				Tempo de Resposta WCRT (ms)
	C (ms)	J (ms)	T (ms)	D (ms)	
$m_2$	0.524	0	15	15	1.048
$m_4$	0.524	0	20	20	1.572
$m_6$	0.524	0	25	25	2.096

O período de amostragem dos pêndulos invertidos é dado respectivamente por:  $h = 15ms, 20ms$  e  $25ms$  e o atraso de entrada-saída a partir da equação (5.15).

$$L_{io} = C_{t_s} + R_{m_i} + C_{t_c} + C_{t_a} \quad (5.15)$$

Vamos considerar que o tempo de computação das tarefas de medição, execução do algoritmo de controle e atuação sejam pequenos, mas não desprezíveis para os cálculos do atraso de entrada-saída,  $C_{t_s} = C_{t_c} = C_{t_a} = 1ms$ .

O atraso de entrada-saída para os NCS's relativo aos pêndulos invertidos são dados a seguir:

$$NCS_1 \rightarrow L_{IO} = C_{t_s} + R_{m_1} + C_{t_c} + C_{t_a} \cong 4ms \quad (5.16)$$

$$NCS_2 \rightarrow L_{IO} = C_{t_s} + R_{m_1} + C_{t_c} + C_{t_a} \cong 4.5ms \quad (5.17)$$

$$NCS_3 \rightarrow L_{IO} = C_{t_s} + R_{m_1} + C_{t_c} + C_{t_a} \cong 5ms \quad (5.18)$$

#### 5.4.2 Projeto do Controlador LQG

O projeto do controlador *LQG* é realizado baseado na ferramenta computacional *Jitterbug* (Lincoln, 2001) no *MALTB/Simulink*®, usando o comando *lqgdesign.m* e, ainda, considerando o *período de amostragem* e *atraso de entrada-saída* de cada pêndulo invertido. O *atraso de entrada-saída* é determinado a partir do tempo de computação das tarefas de medição, controle e atuação e cálculo de tempo de resposta no pior caso para (*WCRT*) o fluxo de mensagens do *NCS*.

Os controladores *LQG* para os *NCS's* são implementados na forma de espaço de estados discreto no tempo, com os parâmetros temporais dos algoritmos dados por:  $h = 15, 20$  e  $25$  ms e  $L_{IO} = 4.0, 4.5$  e  $5.0$  ms, e. As Eq. (4.19) – (4.22) apresentam na forma de espaço de estados os controladores *LQG*.

$$\begin{aligned} x(h_{k+1}) &= Ax(h_k) + Bu(h_k) \\ y(h_k) &= Cx(h_k) + Du(h_k) \end{aligned} \quad (4.19)$$

$$A_1 = \begin{bmatrix} 0.3661 & -0.8496 & 0.003161 \\ 0.3684 & -0.8054 & 0.002475 \\ -29.66 & 65.78 & -0.2225 \end{bmatrix} \quad B_1 = \begin{bmatrix} 0.2598 \\ 1.129 \\ -57.7 \end{bmatrix} \quad (4.20)$$

$$C_1 = [-29.26 \quad 65.78 \quad -0.2225] \quad D_1 = [-57.7]$$

$$A_2 = \begin{bmatrix} 0.3708 & -1.266 & 0.00355 \\ 0.2419 & -0.7704 & 0.001879 \\ -20.57 & 67.79 & -0.1777 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0.3979 \\ 1.104 \\ -59.36 \end{bmatrix} \quad (4.21)$$

$$C_2 = [-20.57 \quad 67.79 \quad -0.1777] \quad D_2 = [-59.36]$$

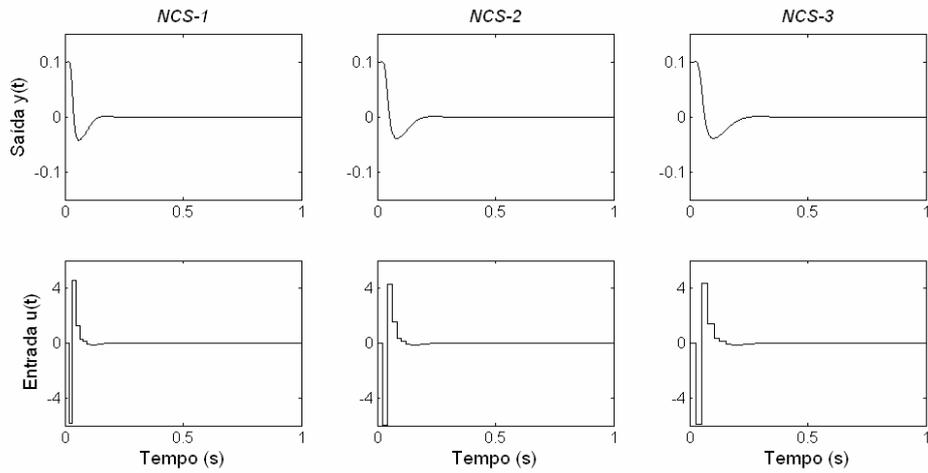
$$\begin{aligned}
A_3 &= \begin{bmatrix} 0.3467 & -0.9548 & 0.003691 \\ 0.2953 & -0.7587 & 0.002588 \\ -16.55 & 43.94 & -0.1595 \end{bmatrix} & B_3 &= \begin{bmatrix} 0.3909 \\ 1.646 \\ -58.23 \end{bmatrix} \\
C_3 &= [-16.55 \quad 43.94 \quad -0.1595] & D_3 &= [-58.23]
\end{aligned} \tag{4.22}$$

### 5.4.3 Simulações

A Fig. 5.3 apresenta os resultados de simulação realizados sob a ferramenta computacional *TrueTime* no MALTAB/Simulink<sup>®</sup> (Henriksson and Cervin, 2000). Na Fig. 5.3 (a), o desempenho dos *NCS* é satisfatório para as especificações de controle desejadas. A partir de uma condição inicial, as saídas  $y(t)$  dos três pêndulos retornam a posição de origem vertical através de um sinal de correção  $u(t)$ .

Na Fig. 5.3 (b) pode-se ver o escalonamento de mensagens sob a rede CAN, onde o tempo de resposta para as mensagens são menores que os especificados na Tabela 5.1. Para os *NCS-2* e *NCS-3*, nota-se que o tempo de resposta das instâncias do fluxo de mensagens correspondente não são constantes, conseqüentemente, o atraso de entrada-saída também não é constante. A partir deste pressuposto, pode-se concluir que os *NCS-2* e *NCS-3* não são sensíveis à *variação de atrasos de entrada-saída (jitter de entrada-saída)* e o método de projeto para *NCS* proposto satisfaz as especificações de projeto.

(a)



(b)

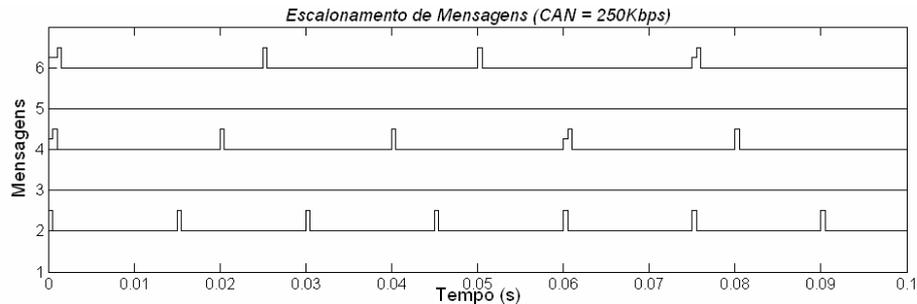


Figura 5.3: Simulação no TrueTime: (a) desempenho de controle e (b) escalonamento de mensagens.

## 5.5 Conclusão

Apresentou-se neste capítulo um método de projeto para *NCS* a partir do tempo de resposta (*WCRT*) de mensagens, considerando que a planta de controle não seja sensível a *variação de atrasos de entrada-saída (jitter de entrada-saída)*.

A partir do método de controle por computador tradicional com atraso de tempo constante aplicado a um *NCS*, pode-se derivar um método de projeto do controlador, a partir do *período de amostragem e atraso de entrada-saída*. Apresentou-se um exemplo de projeto para *NCS* baseado no controlador *LQG* sob uma rede CAN com resultados satisfatórios.

# 6 Projeto de NCS Baseado na Margem de Jitter

## 6.1 Introdução

O conceito de *margem de fase* e *margem de ganho* (Franklin *et al.*, 2002) é utilizado para análise e projeto de sistemas de controle, com capacidade de descrever a sensibilidade de uma malha de controle com relação às incertezas paramétricas de uma planta física. Em (Cervin *et al.*, 2004) foi apresentada uma abordagem de projeto para sistemas de controle de tempo real, considerando a *margem de jitter* de saída do sistema.

Um exemplo de aplicação baseado na teoria de escalonamento e controle integrados (Cervin, 2003), foi apresentado na forma de um sistema de controle em tempo real no qual três plantas de físicas foram fechadas sob um único sistema computacional, dado que suas tarefas de controle eram escalonadas a partir de um algoritmo de escalonamento do tipo RM ou EDF (Liu and Layland, 1973). Um algoritmo para seleção do período de amostragem foi proposto para otimizar o desempenho global do sistema e ao mesmo tempo garantir-lhe a estabilidade.

Em sistemas de controle via redes os elementos sensor, controlador e atuador trocam informações entre si através de mensagens sob uma rede de comunicação, que gera atrasos na transmissão e, conseqüentemente, induz atrasos de controle, o que pode levar o sistema à perda de desempenho e instabilidade.

A abordagem de *margem de jitter* para tratar o problema de *Sistemas de Controle via Rede* (NCS: “*Networked Control Systems*”) (Santos *et al.*, 2004) é proposta neste trabalho, em que o controlador pode ser implementado como uma tarefa em um nó computacional computacional (um kernel de tempo real com uma tarefa única concorrente pelos recursos do computador), porém iremos assumir que os tempos de computação dos algoritmos de controle sejam constantes, gerando outro tipo de incerteza – a impropriedade no escalonamento de mensagens sob a rede de comunicação. A

variabilidade no tempo de transmissão das instâncias de um fluxo periódico de mensagens, em função do escalonamento, gera atrasos e variação de atrasos (*jitter*) de entrada-saída nas malhas de controle consideradas da mesma forma, quando realizadas na forma de um sistema de controle em tempo real. O *jitter de entrada-saída* pode causar uma degradação do desempenho de controle e até mesmo levar a planta física a uma instabilidade.

Baseado no teorema de estabilidade em malhas de controle com atrasos de entrada-saída variantes no tempo (Kao and Lincoln, 2004) e sob o conceito de *margem de jitter* (Cervin *et al.*, 2004) para tarefas de controle, a *margem de jitter* combinado com a teoria de escalonamento em tempo real para transmissão de mensagens é aplicada ao caso de NCS, de forma a garantir o desempenho e estabilidade da planta física.

## 6.2 Margem de Atraso e Margem de Jitter

Sistemas de controle por computador são projetados, considerando amostragem periódica da saída e um atraso computacional igual a zero (Åstrom and Wittenmark, 1997). Numa implementação real, isto não se mostra factível, porque um sistema computacional pode induzir atrasos computacionais consideráveis, introduzindo atrasos ou *jitter* de entrada-saída na planta de controle.

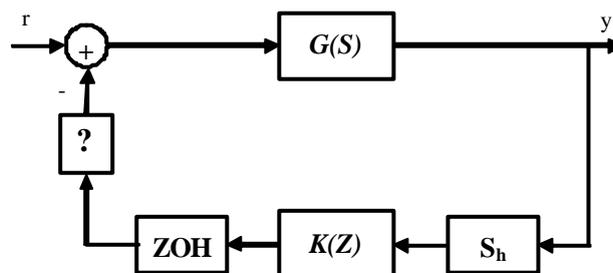


Figura 6.1: Sistema de controle por computador com uma planta contínua no tempo  $P(S)$ , amostrador periódico  $S_h$ , controlador discreto no tempo  $K(Z)$ , segurador de ordem-zero e atraso variante no tempo ? .

A Fig. 6.1 apresenta um modelo na forma de diagrama em blocos de uma malha de controle por computador em que a planta física é descrita por um sistema contínuo linear denotado por uma função de transferência do tipo  $G(S)$ . A saída da planta física é amostrada em intervalos de tempo constante  $h$ . O controlador é descrito por um sistema

discreto linear também descrito por uma função de transferência  $K(Z)$  com um segurador de ordem zero (*sample-hold*), onde existe um atraso variante no tempo  $\tau$  antes do sinal de controle ser aplicado na entrada da planta física.

Considerando um sistema de controle por computador com a arquitetura da Fig. 6.1, (Kao and Lincoln, 2004) formularam um teorema apenas suficiente, mas que garante a estabilidade da malha de controle para quaisquer tipos de atrasos, em um dado intervalo de tempo, incluindo atrasos constante, periódico e aleatório.

### TEOREMA 6.1 – ESTABILIDADE SOB JITTER DE SAÍDA

O sistema em malha fechada da Fig. 6.1 é estável para quaisquer atrasos variante no tempo  $\tau \in [0, Nh]$ , onde  $N > 0$  seja um número real, se

$$\left| \frac{G_{alias}(\mathbf{w})K(e^{i\mathbf{w}})}{1+G_{ZOH}(e^{i\mathbf{w}})K(e^{i\mathbf{w}})} \right| < \frac{1}{\tilde{N}|e^{i\mathbf{w}}-1|}, \quad \forall \mathbf{w} \in [0, \mathbf{p}] \quad (6.1)$$

sendo que  $\tilde{N} = \sqrt{\lfloor N \rfloor^2 + 2\lfloor N \rfloor g + g}$  e  $g = N - \lfloor N \rfloor$ ,  $G_{ZOH}(Z)$  é a discretização do segurador de ordem zero de  $P(S)$ , e

$$G_{alias}(\mathbf{w}) = \sqrt{\sum_{k=-\infty}^{\infty} \left| G\left(i(\mathbf{w} + 2\mathbf{p}k)\frac{1}{h}\right) \right|^2} \quad (6.2)$$

*Demonstração:* Veja em (Kao and Lincoln, 2004).

Em uma implementação real, o algoritmo de controle é implementado num sistema computacional na forma de tarefa de controle (algoritmo) com ativação periódica em instantes de tempo,  $h$ . A Fig. 6.2 apresenta um diagrama de tempo com as propriedades temporal de uma tarefa de controle.

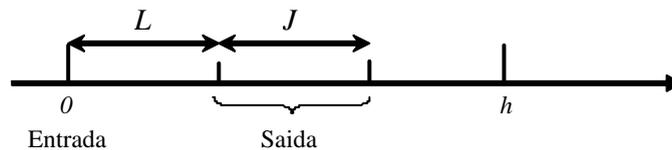


Figura 6.2: O atraso de entrada-saída é composto pelo atraso constante (L) e jitter (J).

O atraso de entrada-saída é o tempo gasto entre a medição da saída, execução do algoritmo de controle e atuação na planta física, podendo ser dividido em duas partes: uma parte constante,  $L = 0$ , e uma parte variante no tempo (o *jitter*),  $J = 0$ . O atraso mínimo de entrada-saída pode ser experimentado por  $L$  e atraso máximo de entrada-saída dado por  $L + J$ .

A seguir, serão apresentadas as definições de *margem de atraso* e *margem de jitter*, a serem utilizadas nas seções seguintes, podendo ser entendidas com maiores detalhes em (Cervin *et al.*, 2004).

#### DEFINIÇÃO 6.1 – MARGEM DE ATRASO

Para um sistema de controle por computador da Fig. 6.1, a *margem de atraso* é definida pelo maior número inteiro  $L_m$ ; onde a estabilidade do sistema em malha fechada seja garantida assumindo um *atraso constante*  $\tau = L_m$ .

Em sistemas de controle contínuo no tempo, a *margem de atraso* pode ser computada como:

$$L_m = \frac{\phi_m}{\omega_c} \quad (6.3)$$

onde  $\phi_m$  é a *margem de fase* e  $\omega_c$  é a *freqüência de cruzamento* do sistema em malha fechada.

#### DEFINIÇÃO 6.2 – MARGEM DE JITTER

Para um sistema de controle por computador da Fig. 6.1, a *margem de jitter* é definida pelo maior número  $J_m(L)$ , em que a estabilidade do sistema de controle em malha fechada seja garantida assumindo um *atraso variante no tempo*  $\tau \in [L, L + J_m(L)]$ . Para aplicação do Teorema 6.1 ao sistema de controle da Fig. 6.1, deve-se considerar a planta atrasada no tempo na forma de  $G(s)e^{-sL}$  e  $N = J/h$ .

### 6.2.1 Análise de Desempenho e Estabilidade

A estabilidade de um sistema de controle em malha fechada é garantida se, a partir do conhecimento do tempo de execução do algoritmo de controle, conhecermos o *atraso constante*  $L$  e um *jitter*  $J$ , com a seguinte desigualdade satisfeita:

$$J_m(L) > J \quad (6.4)$$

#### DEFINIÇÃO 6.3 – MARGEM DE ATRASO PARA SISTEMAS COM ATRASO E JITTER

Para um sistema de controle por computador da Fig. 6.1, onde se observa um *atraso constante*  $L$  e um *jitter*  $J$ , a *margem de atraso* é definida pelo maior número  $L_m$  em que a estabilidade do sistema em malha fechada seja garantida para qualquer atraso variante no tempo  $\Delta \in [L + L_m, L + L_m + J]$ .

Em relação à função *margem de jitter*  $J_m(L)$ , a *margem de atraso* é dada pelo menor valor de  $L_m$  que satisfaça:

$$J_m(L + L_m) = J \quad (6.5)$$

#### DEFINIÇÃO 6.4 – MARGEM DE FASE APARENTE

Para um sistemas de controle por computador da Fig. 6.1, com um *atraso constante*  $L$  e um *jitter*  $J$ , a *margem de atraso aparente* é definida pelo maior número  $\hat{J}_m$  para que a estabilidade seja garantida em qualquer atraso variante no tempo  $\Delta \in [L + \hat{J}_m / \omega_c, L + \hat{J}_m / \omega_c + J]$  a qual  $\omega_c$  é a *frequência de cruzamento* do sistema, considerando um *atraso constante*  $L$ .

De forma análoga, em relação à função *margem de jitter*  $J_m(L)$ , a *margem de fase aparente* é dada pelo menor valor de  $\hat{J}_m$  que satisfaça:

$$J_m(L + \hat{J}_m / \omega_c) = J \quad (6.6)$$

### 6.2.2 Atribuição de Deadline

A partir de um deadline crítico real, pode-se garantir a estabilidade em malha fechada do sistema de controle, dado que a partir de um limite inferior sob o *atraso constante*  $L$ , garante-se a estabilidade assumindo um deadline relativo:

$$D = L + J_m(L) \quad (6.7)$$

Estendendo a atribuição de deadline que garanta certa margem de fase aparente para o sistema alvo, dado um limite inferior sob um *atraso constante*  $L_m$  e uma *margem de fase aparente* desejada  $\hat{\mathbf{j}}_m < \mathbf{w}_c(L_m - L)$ , pode-se garantir um nível de desempenho atribuindo um deadline na forma de:

$$D = L + J_m(L\hat{\mathbf{j}}_m / \mathbf{w}_c) \quad (6.8)$$

## 6.3 Margem de Jitter Aplicado a Sistemas de Controle via Redes

Considere uma arquitetura de *NCS*, conforme demonstrado na Fig. 6.3 (a), com dois nós computacionais sob a rede e um fluxo de mensagens entre o nó computacional controlador e atuador. Em nosso caso específico, iremos analisar o impacto do escalonamento de mensagens sob o desempenho e estabilidade da malha de controle.

Consideramos ainda que as tarefas de medição  $\mathbf{t}_s$  e atuação  $\mathbf{t}_a$ , com tempo de computação respectivamente dados por  $C_{\mathbf{t}_s}$  e  $C_{\mathbf{t}_a}$  a partir do *WCET* (*Worst Case Execution Time*) de tarefas sejam considerados nulos. O tempo de computação da tarefa de controle  $\mathbf{t}_c$  é constante, dado por  $C_{\mathbf{t}_c}$ , e o tempo de transmissão da mensagem  $m_{ca}$  (mensagem enviada do nó computacional controlador/sensor para o nó computacional atuador) é dado por  $C_{m_{ca}}$ , que é em função da carga de mensagens e algoritmo de escalonamento sob a rede (Tovar, 1999).

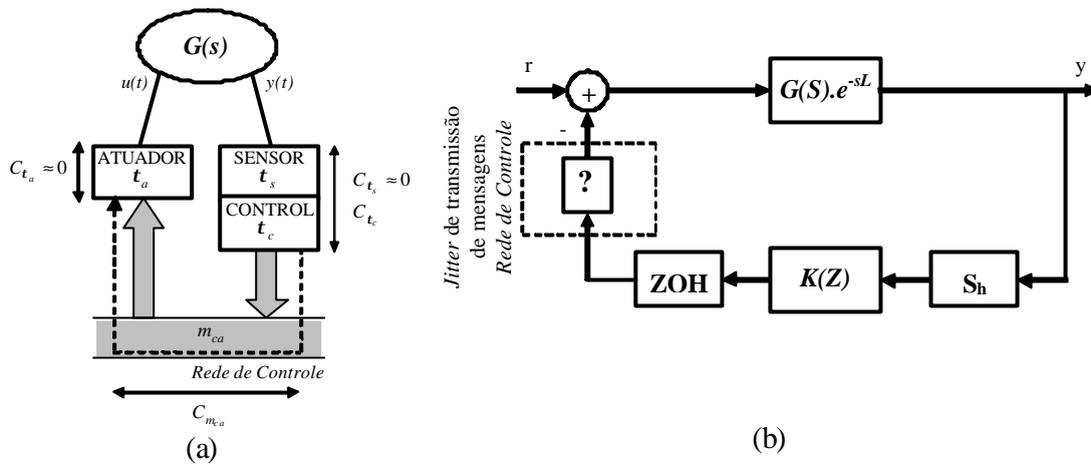


Figura 6.3: (a) Arquitetura de um NCS, (b) Diagrama em blocos do NCS.

A Fig. 6.3 (b) apresenta um modelo em diagrama de blocos para uma malha de controle na forma de NCS, onde a planta física é descrita por um sistema linear contínuo e com um atraso no tempo  $G(S).e^{-sL}$ , e dado que  $N = J/h$ . O termo  $L$  representa o atraso na execução de tarefas e transmissão de mensagens em um NCS. A saída da planta física é amostrada em intervalos de tempo constante  $h$ . O controlador é descrito por um sistema discreto no tempo e linear  $K(Z)$  e um segurador de ordem zero, em que existe um atraso variante no tempo  $?$  antes do sinal de controle ser aplicado na entrada da planta física.

O diagrama temporal do atraso de entrada-saída referente a um NCS é apresentado nas Fig. 6.4 e 6.5 e pode também ser dividido em duas partes: o atraso de entrada-saída constante  $L = 0$  que compreende o tempo de execução do algoritmo de controle mais o tempo de transmissão da mensagem quando não existem mensagens a serem transmitidas sob a rede,  $L = C_{t_c} + C_{m_{ca}}$ .

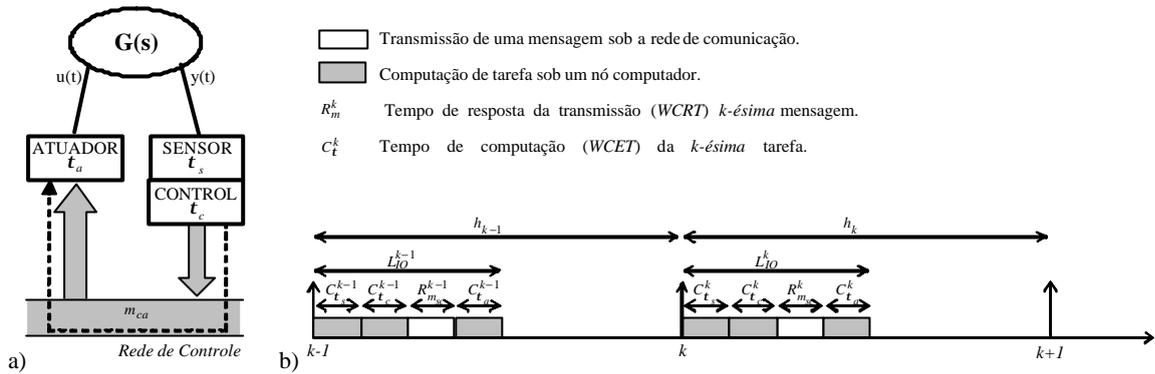


Figura 6.4: (a) Arquitetura de NCS; (b) Diagrama temporal das tarefas e mensagens.

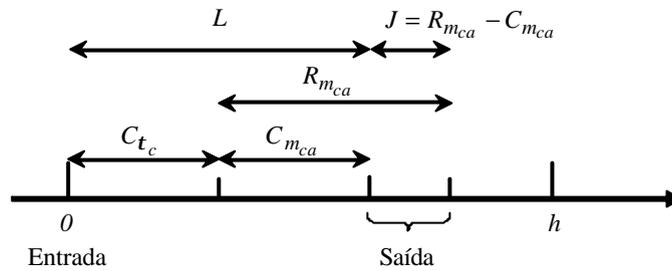


Figura 6.5: O atraso de entrada-saída pode ser dividido na forma de atraso constante ( $L$ ) e um jitter ( $J$ ).

A parte variante no tempo (o *jitter*),  $J = 0$ , que compreende a diferença do tempo de transmissão de uma instância da mensagem para o *pior-caso* e *melhor-caso*,  $J = R_{m_{ca}} - C_{m_{ca}}$ .

## 6.4 Análise do *Jitter* em Mensagens

Para definir o *atraso constante* e o *jitter* em um *NCS*, deve-se considerar que cada *NCS* seja composto por tarefas com tempo de execução constantes, executadas em nós computacionais com capacidade computacional, isolados e sem concorrência de recursos e mensagens que compartilham a largura de banda de um sistema de comunicação.

A partir deste pressuposto, pode-se aplicar a análise de estabilidade e desempenho apresentada na seção 6.2. As Eq. 6.9 e 6.10 apresentam a relação do *atraso constante* e *jitter* em um *NCS* composto por tarefas e mensagens.

$$L_i = C_{t_c,i} + C_{m_{ca},i} \quad (6.9)$$

$$J_i = R_{m_{ca},i} - C_{m_{ca},i} \quad (6.10)$$

Aplica-se a análise do tempo de resposta em tarefas (Audsley *et al.*, 1993), e mensagens (Tindell *et al.*, 1995).

A seguir uma breve discussão acerca da análise do tempo de resposta em redes CAN, no qual o escalonamento é realizado sob atribuição de prioridade fixa sob mensagens.

#### 6.4.1 Modelo de Mensagens

Assume-se que a rede CAN possua  $r$  estações e  $n$  fluxos de mensagens definidas por:

$$S_m = (C_m, T_m, D_m) \quad (6.11)$$

Um fluxo de mensagem é uma seqüência temporal de mensagens relativas a, por exemplo, à leitura remota de uma variável de processo específico. Um fluxo de mensagem  $S_m$  é caracterizado por um único identificador.  $C_m$  é o tempo de transmissão da mensagem do fluxo  $S_m$ .  $T_m$  é a periodicidade da requisição de fluxos  $S_m$ , considerado como o intervalo de tempo entre duas chegadas consecutivas de requisições para a fila de saída do fluxo  $S_m$ . Finalmente,  $D_m$  é a meta temporal relativa de uma mensagem (deadline), que consiste no intervalo de tempo máximo admissível entre o instante em que a requisição da mensagem é colocada na fila de saída e o instante em que cada mensagem é completamente transmitida.

#### 6.4.2 Análise do Tempo de Resposta no Pior-Caso

Em (Tindell *et al.*, 1995), os autores apresentaram em detalhes a análise do tempo de resposta das mensagens em redes CAN. Definiu-se que as mensagens possuem atribuição de prioridade fixa para os respectivos fluxos e um modelo de escalonamento não-preemptivo. Considerando tal modelo de escalonamento, adaptou-se toda a análise de escalonabilidade sob um conjunto de tarefas, desenvolvida por (Audsley *et al.*, 1993).

O tempo de resposta no pior caso de uma mensagem enfileirada,  $R_m$  medida desde a chegada de uma requisição da mensagem para a fila de saída até o instante em que a mensagem é completamente transmitida, é dado pela Eq. (6.12):

$$R_m = J_m + w_m + C_m \quad (6.12)$$

O termo  $C_m$  representa o tempo gasto para se transmitir uma mensagem  $m$  fisicamente sobre o barramento, o termo  $w_m$  representa o atraso na fila de transmissão no pior caso. Finalmente, o termo  $J_m$ , representa o *jitter* para inserção da mensagem na fila de transmissão.

Para um NCS baseado em rede CAN, o atraso de controle máximo está diretamente relacionado com o cálculo do máximo tempo de resposta da mensagem.

O pacote de uma mensagem CAN básica, contém 47 bits de overhead por mensagem, e um *bitstuffing* de 5 bits, sendo que apenas 34 bits dos 47 bits de overhead que estão sujeitos ao *stuffing*. O termo  $C_m$  é apresentado a seguir.

$$C_m = \left( \left\lfloor \frac{34 + 8 \cdot s_m}{5} \right\rfloor + 47 + 8 \cdot s_m \right) \cdot t_{bit} \quad (6.13)$$

Em (6.13), temos que o termo  $s_m$  indica o tamanho limitado da mensagem  $m$  em bytes, e o termo  $t_{bit}$  é o atraso de propagação, ou seja, o tempo necessário para se transmitir um bit sobre o barramento (como exemplo podemos afirmar que, sob um barramento a  $1Mbit/s$ , tem-se tem-se que  $t_{bit} = 1\mu s$ ).

O termo  $w_m$  em (6.12), representa o atraso na fila no pior caso (O maior tempo entre a inserção de uma mensagem na fila de prioridades e o início de sua transmissão) e dado por (6.14).

$$w_m = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{w_m + J_j + t_{bit}}{T_j} \right\rceil \cdot C_j \quad (6.14)$$

Onde o termo  $hp(m)$  é o conjunto de mensagens no sistema com prioridade maior que a mensagem  $m$ , e  $B_m$  é o tempo de bloqueio no pior caso da mensagem  $m$ , e é dado por (6.15).

$$B_m = \max_{\forall k \in lp(m)} (C_k) \quad (6.15)$$

De (6.15) pode-se afirmar que  $lp(m)$  é o conjunto de mensagens no sistema com prioridade menor que a mensagem  $m$ . Em redes CAN é possível determinarmos atrasos de transmissão máximos sob a presença de carga através de prioridade sob mensagens.

### 6.4.3 Análise do Tempo de Resposta no Melhor-Caso

Sob escalonamento de prioridade fixa para tarefas, a análise do tempo de resposta no melhor caso considerando  $D = T$  foi desenvolvida por (Redell and Sanfridson, 2002), e podendo ser facilmente estendida ao caso de escalonamento de mensagens em redes CAN. O tempo de resposta no melhor caso de uma mensagem  $m$  pode ser dado pela Eq. (6.16).

$$w_m^b = B_m^b + \sum_{\forall j \in lp(m)} \left[ \frac{w_m^b + J_j + t_{bit}}{T_j} - 1 \right] \cdot C_j^b \quad (6.16)$$

O termo  $B_m^b$  é o tempo de bloqueio no melhor caso da mensagem  $m$ , representado por (6.17).

$$B_m^b = \min_{\forall k \in lp(m)} (C_k) \quad (6.17)$$

De (6.17) entende-se que  $lp(m)$  é o conjunto de mensagens no sistema com prioridade menor que a mensagem  $m$ .

## 6.5 Exemplo de Projeto

Vamos apresentar como o conceito de *margem de jitter* pode ser aplicado no projeto de NCS. Baseado no procedimento de projeto integrado de controle-escalonamento apresentado por (Cervin *et al.*, 2004), considere um conjunto de  $n$  plantas e controladores contínuos no tempo, denotados respectivamente por  $P(s)$  e  $K(s)$ , onde o atraso ( $L$ ) e *jitter* ( $J$ ).

O procedimento de projeto é dado a seguir:

1. Inicialize atribuindo períodos de amostragem  $h$  inicial aos controladores.
2. Ajuste os períodos de amostragem de tal forma que as mensagens sejam escalonadas pela política de escalonamento da rede CAN ( $RM$ ).

3. Discretize os controladores, usando o período de amostragem atribuído, tendo assim um conjunto de controladores discretos no tempo  $K(z)$ .
4. Para cada NCS, compute o tempo de resposta do algoritmo onde se tem tarefa e mensagem para cada NCS. O tempo de resposta no pior-caso,  $R_{task,i}^b + R_{msg,i}$  e o tempo de resposta no melhor-caso  $R_{task,i}^b + R_{msg,i}^b$ .
5. Para cada algoritmo de controle compute a margem de *jitter* baseado no Teorema 1 e a margem de fase aparente  $\hat{f}_{m,i}$ , considerando um atraso constante  $L_i = R_{task,i}^b + R_{msg,i}^b$  e o *jitter*  $J_i = R_{msg,i} - R_{msg,i}^b$ .
6. Para cada algoritmo de controle, calcule a degradação de desempenho  $r_i = \hat{f}_{m,i} / \hat{J}_{m,i}$ . Calcule também o valor médio  $\bar{r} = \sum r_i / n$ .
7. Para cada algoritmo de controle, ajuste o período de acordo com  $h_i = h_i + kh_i(r_i - \bar{r}) / \bar{r}_i$  onde  $k < 1$  é um parâmetro de ganho.
8. Repita o passo 2 até que tenha um desempenho aceitável para os NCS's. O critério de parada define-se pela soma da diferença de desempenho,  $\sum |r_i - \bar{r}|$  não seja crescente.

Considere agora três plantas físicas e os respectivos controladores contínuos no tempo dados por (Carvin *et al.*, 2004) :

$$\begin{aligned}
 P_1(s) &= \frac{8 \cdot 10^5}{s(s+1000)} & K_1(s) &= \frac{4.88 \cdot 10^4 (s + 2 \cdot 10^5)(s + 1295)}{(s + 5000)(s^2 + 7.325 \cdot 10^4 s + 2.573 \cdot 10^9)} \\
 P_2(s) &= \frac{4 \cdot 10^4}{(s-200)(s+200)} & K_2(s) &= \frac{2.57 \cdot 10^4 (s + 2 \cdot 10^5)(s + 259.1)}{(s + 3000)(s^2 + 1.645 \cdot 10^4 s + 1.35 \cdot 10^8)} \\
 P_3(s) &= \frac{5 \cdot 10^7}{s(s^2 + 100s + 2.5 \cdot 10^5)} & K_3(s) &= \frac{478(s + 2 \cdot 10^5)(s^2 + 160.6s + 1.655 \cdot 10^5)}{(s + 2740)(s + 1000)(s^2 + 2494s + 7.109 \cdot 10^6)}
 \end{aligned} \tag{6.18}$$

As plantas físicas são controladas pelos controladores, através de uma realimentação de saída, e possuem uma largura de banda  $\omega_b$  e margem de fase  $f_m$ , conforme Tabela 6.1.

Tabela 6.1: Largura de banda e margem de fase das malhas de controle.

Malha de Controle	$\omega_b$	$f_m$
$P_1(s), K_1(s)$	960 rad/s	74.1°
$P_2(s), K_2(s)$	599 rad/s	49.5°
$P_3(s), K_3(s)$	179 rad/s	69.7°

A resposta dos sistemas de controle contínuo no tempo e em malha fechada é apresentado na Fig. 6.6.

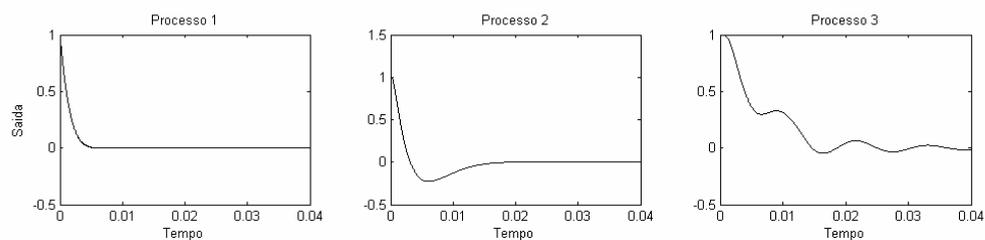


Figura 6.6: Resposta das malhas de controle contínuo no tempo.

Para implantação do algoritmo de controle num sistema computacional, os períodos de amostragem são escolhidos pela *regra de thumb*  $0,2 = \omega_b \cdot h = 0,6$ .

A arquitetura de NCS assumida para as três plantas físicas é denotada por um nó computacional sensor/controlador e outro nó computacional atuador, consistindo em um fluxo de mensagens gerado na rede de controle, conforme Fig. 6.7. Os NCS's são suportados por uma rede CAN de 1 Mbps (Esta taxa de transmissão não é padrão, então tem-se um valor hipotético para efeito de simulação).

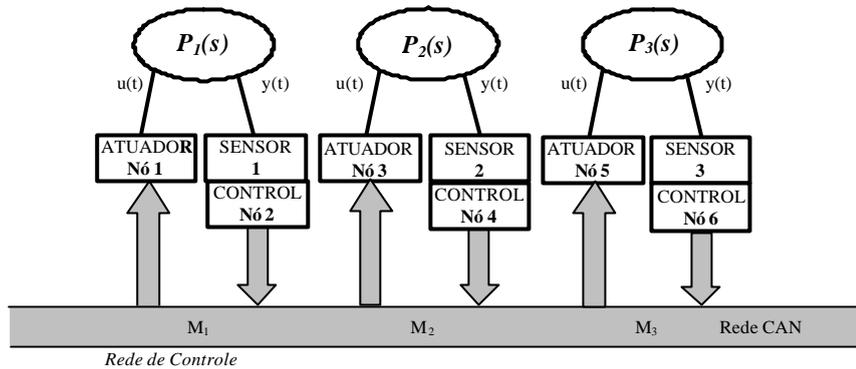


Figura 6.7: Configuração de NCS's sob uma rede CAN.

O fluxo de mensagens é periódico e gerado pelos nós computacionais sensor, 2, 4 e 6, respectivamente para os nós computacionais 1, 3 e 5. As propriedades temporais das mensagens de controle são dadas na Tabela 6.2. A atribuição de prioridades aos diferentes fluxo de mensagens é baseada no algoritmo *RM (Rate Monotonic)*.

Tabela 6.2. Características dos fluxos de mensagens – Rede CAN (1Mbps).

Mensagens	$C_m$ (ms)	$T_m$ (ms)	$J_m$ (ms)	$U_m$ (%)	$WCRT$ (ms)	Ativação
$m_1$	0.130	0.400	0.020	32.50	0.280	Timed-driven
$m_2$	0.130	0.560	0.020	23.21	0.410	Timed-driven
$m_3$	0.130	2.600	0.020	5.00	0.570	Timed-driven

O termo  $C_m$  indica o tempo de transmissão de um frame completo, que transporta a mensagem;  $T_m$  é o período de ativação do frame (corresponde ao período de amostragem da malha com uma defasagem de ativação  $h_k$ );  $J_m$  é o *jitter* de transmissão de liberação da mensagem;  $U_m$  indica a utilização da largura de banda da respectiva mensagem sob a rede e  $WCRT$  compreende o tempo de resposta no pior caso de uma mensagem.

Baseado no trabalho de (Cervin *et al.*, 2004), de controle-escalonamento integrados considera-se também um ganho de adaptação  $k = 0.2$ . Na Tabela 6.3 apresenta-se os valores referentes a primeira e décima interação. Note a partir dos parâmetros da Tabela 6.3 (a) que na primeira interação apenas a margem de fase aparente do *NCS-3* é negativa, indicando assim a instabilidade do sistema e podendo ser confirmado através de resultados de simulação através da Fig. 6.8 (a). Os *NCS-1* e *NCS-2* possuem desempenho e estabilidade aceitável baseado na *regra de thumb*.

Com base no algoritmo proposto, nota-se que os parâmetros da Tabela 6.3 (b) para a décima iteração, convergem com os períodos de amostragem ajustados a um valor, de forma que o desempenho e estabilidade para os *NCS's* sejam aceitáveis. A Fig. 6.8 (b) mostra através de simulação que os *NCS's* apresentam bom desempenho e estabilidade na décima iteração.

Tabela 6.3: Resultados do projeto de *NCS* sob uma rede CAN: a) primeira iteração, b) décima iteração.

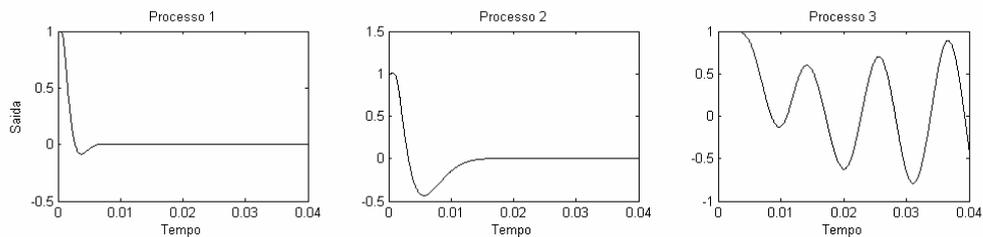
(a)

NCS	$h$	$C_{t_c} + R_{m_{ca}}$	$C_{t_c} + C_{m_{ca}}$	$J$	$J_m(C_{t_c} + C_{m_{ca}})$	$\hat{\mathbf{j}}_m$	$\hat{\mathbf{j}}_m / \mathbf{j}_m$
<i>NCS-1</i>	0.40	0.29	0.16	0.13	1.04	55.7°	0.75
<i>NCS-2</i>	0.56	0.41	0.16	0.25	1.17	32.1°	0.65
<i>NCS-3</i>	2.60	0.57	0.16	0.52	0.04	-10.7	-0.15

(b)

NCS	$h$	$C_{t_c} + R_{m_{ca}}$	$C_{t_c} + C_{m_{ca}}$	$J$	$J_m(C_{t_c} + C_{m_{ca}})$	$\hat{\mathbf{j}}_m$	$\hat{\mathbf{j}}_m / \mathbf{j}_m$
<i>NCS-1</i>	0.55	0.29	0.16	0.13	1.01	51.1	0.70
<i>NCS-2</i>	0.60	0.41	0.16	0.50	1.09	30.0	0.61
<i>NCS-3</i>	1.80	0.67	0.16	0.75	1.15	34.5	0.50

(a)



(b)

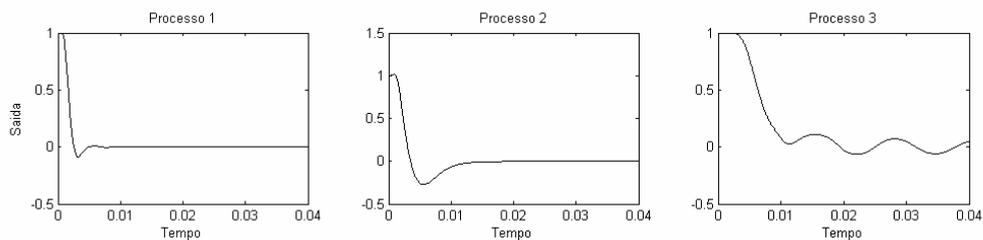


Figura 6.8: Respostas dos NCS's sob a rede CAN: a) primeira iteração, b) décima iteração.

As simulações foram realizadas com a ferramenta computacional TrueTime (Henriksson *et al.*, 2002).

## 6.6 Conclusão

Neste capítulo apresentou-se a aplicação da *margem de jitter* no projeto de *NCS*. Baseado no conceito de *margem de jitter* apresentado em (Cervin *et al.*, 2004) com a sua aplicação ao projeto de sistemas de controle em tempo real, estendeu-se ao caso de sistemas de controle via redes.

Baseado no projeto integrado de controle-escalonamento integrado, algoritmos de escalonamento de tempo real introduzem *jitter* de saída na malhas de controle, o qual através de teste de escalonabilidade pode calcular o *jitter* de execução das tarefas de controle. Uma extensão ao caso de *NCS* foi realizada, considerando que as mensagens sob a rede induzem *jitter* de saída nas plantas físicas. Nós utilizamos o mesmo procedimento de projeto de sistemas de controle em tempo real para *NCS* sob uma rede CAN do artigo original, onde pudemos verificar os resultados através de simulações na ferramenta computacional TrueTime.

# 7 Conclusão e Trabalhos Futuros

## 7.1 Conclusão Geral

Nesta tese foram apresentadas metodologias de projeto para sistemas de controle via redes, considerando as propriedades temporais das redes de comunicação. Realizou-se a análise das propriedades de um sistema de comunicação do tipo difusão, que suporta tráfego de mensagens em tempo real, onde se destaca o fato em determinar o cálculo de resposta no pior caso de mensagens (*WCRT*), que está intrinsecamente relacionada com o atraso de entrada-saída em uma malha de controle fechada sob um sistema de comunicação.

A escolha e análise das propriedades das redes definidas: CAN, Token-passing e TDMA foram essenciais para o desenvolvimento deste trabalho, cujo objetivo consiste em suportar sistemas de controle via redes. As aplicações de controle fechadas sob uma rede de comunicação requerem um tráfego de mensagens em tempo real; sendo que, para as redes apresentadas acima, desenvolveu-se esta análise baseada no cálculo do tempo de resposta no pior caso de mensagens. Realizou-se também uma análise do impacto da comunicação sob desempenho e estabilidade em malhas de controle, neste sentido, torna-se imprescindível conhecer as propriedades temporais das redes de comunicação, conforme apresentado anteriormente.

Desenvolveu-se uma metodologia de projeto para *NCS* no qual o controlador considera o *período de amostragem* e *atraso de entrada-saída* baseado na análise do tempo de resposta de mensagens (*WCRT*) de um fluxo de mensagens relacionado ao *NCS*. Verificou-se que esta abordagem é aceitável para sistemas de controle insensíveis a variação de atrasos de entrada-saída (*jitter*). Para malhas de controle sensíveis a variação de atrasos de amostragem e entrada-saída, a abordagem de projeto baseada no *WCRT* de mensagens torna-se inviável, pois o desempenho e estabilidade podem não ser garantidos em tempo de projeto. Para satisfazer uma classe de sistemas sensíveis ao *jitter*, apresentou-

se um modelo de projeto para *NCS*, baseado na análise de estabilidade, considerando a *margem de atraso* e a *margem de jitter* de sistema de controle em tempo real.

## 7.2 Sugestões para Trabalhos Futuros

***Projeto de NCS Considerando outras Redes de Comunicação.*** Apresentaram-se técnicas de análise e projeto em redes CAN. É de fundamental importância a aplicação destas técnicas em outras redes de comunicação em tempo real para avaliação de desempenho e estabilidade de *NCS*.

***Projeto Integrado de NCS.*** Uma abordagem para a solução integrada de *NCS* que vise otimizar o escalonamento de mensagens e desempenho de controle facilitaria o trabalho do projetista de sistemas de controle, deixando transparente assim à parte de redes e escalonamento de mensagens.

***Abordagem de Escalonamento de Sistemas Distribuídos para NCS.*** As técnicas de análise de escalabilidade de sistemas distribuídos podem ser eficientemente aplicadas ao projeto de *NCS*, onde os algoritmos de controle são distribuídos e uma rede de comunicação em tempo será utilizada.

***Análise e Projeto de NCS Considerando o atraso e variação de atrasos.*** Foram apresentadas técnicas de análise e projeto de *NCS* considerando *período de amostragem* e *atraso de entrada-saída*. A análise apurada de *NCS* com atraso e *jitter* de amostragem, atuação mostra-se faz necessário e interessante para plantas físicas sensíveis a estes parâmetros.

***Aplicação das Técnicas de Projeto em NCS.*** Um estudo de caso de técnicas de projeto de *NCS* aplicado a um sistema de *Mesa-XY* e um sistema automotivos *X-by-Wire* está sendo desenvolvido, onde resultados preliminares foram avançados e não apresentados nesta tese.

## 8 Referências

- Agrawal, G., Chen, B., Zhao, W. and Davari, S. (1997). Guaranteeing Synchronous Message Deadlines with Timed Token Protocol. In *Proceedings of Distributed Computing System*, pp.468-475.
- Almutairi, N. B., Chow, M.-Y. And Tipsuwan, Y. (2001). Network-based controlled DC motor with fuzzy compensation. *The 27th annual conference of the IEEE industrial electronics society (IECON 01)*, Vol. 3 (pp.1844-1849), Denver, CO.
- Alves, M., Tovar, E. and Vasques, F. (2000). Ethernet Goes Real-Time: A Survey on Research and Technological Developments. *Technical Report – HURRAY-TR-2K01*, Polytechnic Institute of Porto, School of Engineering (ISEP-IPP).
- Arvind, K., Ramamitham, K. and Stankovic, J. A. (1991). A Local Area Network Architecture for Communication in Distributed Real-Time Systems". *The Journal of Real-Time Systems*, pp. 115-147, Kluwer Academic Publishers, 1991.
- Årzén, K.-E., Bernhardsson, B., Eker, J., Cervin, A., Nilsson, K., Persson, P. and Sha, L. (1999). Integrated Control and Scheduling. *Report ISRN LUTFD2/TFRT--7586--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.*
- Årzén, K-E., Cervin, A., Eker, J. and Sha, L. (2000). An Introduction to Control and Scheduling Co-Design. *39<sup>th</sup> IEEE Conference on Decision and Control*, Sydney, Austrália, December 12-15.
- Åstrom, K.J. and Wittenmark, B. (1997). *Computer-Controlled Systems. Theory and Design. Third Edition.* Prentice Hall. ISBN 0-13-314899-8.

- Audsley, N., Burns, A., Richardson, M., Tindell, K and Wellings, A. (1993). Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling. In *Software Engineering Journal*, Vol. 8, No. 5 - pp. 285-292.
- Belschner, R. et al. (2001). Flexray Requirements Specification (draft) , version 1.9.7, Flexray Consortium; <http://www.flexray.group.com>.
- Bestavros, A. (1997). LOAD PROFILING – A Methodology for Scheduling Real-Time Tasks in a Distributed System. In *Proceedings of ICDCS'97: The International conference on Distributed Computing Systems*. Baltimore, Maryland.
- Boustany, N., Folkerts, M., Rao, K., Ray, A., Troxel, L. & Zhang, Z. (1992). Um simulation based methodology for analyzing network-based intelligent vehicle control systems. *Proceedings of the Intelligent Vehicles Symposium*. Detroit, MI, (pp. 138-143).
- Branicky, M. S., Phillips, S. M. and Zhang, W. (2000). Stability of networked control systems: Explicit analysis of delay. In *Proc. Amer. Control Conf.*, Chicago, IL., pp. 2352-2357.
- Burns, A. (1991). Scheduling Hard Real-Time Systems. In *Software Engineering Journal*, Special Issue on Real-Time Systems, May 1991, pp.116-128.
- Bushnell, L. G. (2001). Networks and Control. In *IEEE Control Systems Magazine*, February 2001, pp. 22-23.
- CANKingdom specification, ver. 3.01, CanKingdom Int'l, Port Huenemi, Calif., 1996; <http://www.cankingdom.org>.
- Carreiro, F. B., Fonseca, K. V. O. and Miranda, C. S. (2000). Uma Ferramenta para Análise de Garantia de Comunicação Tempo real em Redes padrões de Comunicação Industrial (FIELD BUS)”, *II Workshop de Sistemas em Tempo Real*, Florianópolis, Brasil.

- Cervin, A. (1999). Improved Scheduling of Control Tasks. In *Proceedings of 11th Euromicro Conference on Real-Time Systems*, York, England, June.
- Cervin, A. (2003). Integrated Control and Real-Time Scheduling. PhD Thesis ISRN LUTFD2/TFRT-1065—SE. Department of Automatic Control, Lund Institute of Technology, Sweden.
- Cervin, A. and Lincoln, B. (2003). Jitterbug 1.1 – Reference manual. Technical Report ISRN LUTFD2/TFRT—7604—SE. Department of Automatic Control, Lund Institute of Technology, Sweden.
- Cervin, A., Henriksson, D., Lincoln, B. and Arzén, K. E. (2003). Analysis and Simulation of Controller Timing. In *Control Systems Magazine*.
- Cervin, A., Lincoln, B., Eker, J. and Cervin, A., Arzen, K-E. and Buttazzo, G. (2004). The Jitter Margin and Its Application in the Design of Real-Time Control Systems. *Proceedings of the 11<sup>th</sup> International Conference on Real-Time and Embedded Computing Systems Applications (RTCSA)*, Goteborg, Sweden, August.
- Chen, B. and Zhao, W. (1992). Properties of the timed Token Protocol. *Technical Report – 92-038 – Texas A&M University*.
- Chen, B., Agrawal, G. and Zhao, W. (1992). Optimal Synchronous Capacity Allocation for Hard Real-Time Communications with the Timed Token Protocol. In *Proceedings of the 13<sup>th</sup> IEEE Real-Time Systems Symposium*, pp.198-207.
- Day, J. and Zimmermann, H. (1993). The OSI Reference Model. In *Proceedings of the IEEE*, Vol. 79, n° 12, pp. 1334-1340.
- Dorf, R.C. and Bishop, R.H. (1995). *Modern Control Systems*. 7th edition, Addison-Wesley, ISBN 0-201-84559-8.

- Eker, J. and Cervin, A. (1999). A Matlab Toolbox for Real-Time and Control Systems Co-Design. *Proceedings of the 6<sup>th</sup> International Conference on Real-Time Computing Systems and Applications*, Hong Kong, China, December.
- EN 50170 (1996). General Purpose Field Communication System. Volume 1 – P-NET, Volume 2 - PROFIBUS, Volume 3 – WorldFIP.
- Fascioni, L. C. (1996). Avaliação de Arquiteturas de Comunicação Distribuídas para Interligação de Sensores/Atuadores em um Helicóptero não Tripulado. Dissertação de Mestrado, LCMI-UFSC.
- Fonseca, K. V. O. (1995). Schedulability Analysis of a Fibre Channel Communication System”, Technical Report – Electrical Engineering Department – Carnegie Mellon University.
- Fonseca, K. V. O. (1997). Uma Metodologia de Configuração do Suporte de Comunicação de Sistemas Tempo real Crítico. Tese de Doutorado, LCMI-UFSC.
- Fonseca, K. V. O. and Farines, J. M. (1993). Uma Análise das Diversas Propostas de Atendimento dos Requisitos de Comunicação para Sistemas Tempo real em Sistemas de Manufatura. *Anais do Simpósio Brasileiro de Redes de Comunicação*, Campinas-SP Brasil.
- Fonseca, K. V. O. and Farines, J. M. (1995). Um Estudo Comparativo de Técnicas de Análise de Escalonabilidade em um Sistema de Comunicação em Tempo Real. *Anais do Simpósio Brasileiro de Redes de Comunicação*, Fortaleza-CE Brasil.
- Franklin, G., Powell, D. And Emami-Naeini. (2002). *Feedback Control of Dynamic Systems*, 4th edition, Prentice Hall.
- Grow, R. (1982). A Timed Token Protocol for Local Area Networks. In *Proc. of Electro'82*, Token Access Protocols, Paper 17/3.
- Gyron. (1992). Projeto HELIX: Definição e Descrição Geral. *Documento Interno*.

- Gyron. (1996). HELIX Software Reference Manual. *Documento Interno*.
- Haffner et al. (2000). Branch and Bound Algorithm for Transmission System Expansion Planning Using a Transportation Model. *IEEE Proceedings on Generation, Transmission and Distribution*, EUA, 147(3):149-156.
- Halevi, Y. and Ray, A. (1988). Integrated Communication and Control Systems: Part I-analysis. *Journal of Dynamic Systems, Measurement and Control*, 110, 367-373.
- Handaoui, M. and Ramanathan, P. (1992). Selection of Timed-Token MAC Protocol Parameters to Guarantee Message Deadlines”, Technical Report ECE-92-10, Dept. of ECE, Univ. of Wisconsin-Madison.
- Handaoui, M. and Ramanathan, P. (1994). Selection of Timed-Token MAC Protocol Parameters to Guarantee Message Deadlines. *IEEE/ACM Transactions on Networks*.
- Henriksson, D. and Cervin, A. (2003). TrueTime 1.1 – Reference manual. Technical Report ISRN LUTFD2/TFRT—7605—SE. Department of Automatic Control, Lund Institute of Technology, Sweden.
- Henriksson, D., Cervin, A. and Årzen, K-E. (2002). TrueTime: Simulation of Control Loops Under Shared Computer Resources. In *Proceedings of the 15<sup>th</sup> IFAC World Congress on Automatic Control*. Barcelona, Spain.
- Hermant, J. F., Le Lann, G. and Rivierre, N. (1995). A General Approach to Real-Time Message Scheduling over Distributed Broadcast Channels. *INRIA/IEEE Conference on Emerging Technologies and Factory Automation – ETFA’94*, 10-13 October, Paris, France.
- Hong, S. H. (1995). Scheduling algorithm of data sampling times in the integrated communication and control systems. *IEEE Transactions on Control Systems Technology*, 3(2), 225-230.

- ISO (1983). Information Processing Systems – Open Systems Interconnection – Basic Reference Model. ISO / TC97 / SC16.
- ISO 11898 (1993). Road Vehicle - Interchange of Digital Information – Controller Area Network (CAN) for High Speed Communication, Int'l Organization for Standardization, Geneva.
- Joseph, M. and Pandya, P. (1986). Finding Response Times in a Real-Time System. In *The Computer Journal*, Vol. 29, n° 5, pp. 390-395.
- Kao, C.-Y, and B. Lincoln (2004). Simple stability criteria for systems with time-varying delays. *Automatica*. To appear in September 2004.
- Koeptz, H. (1994). A Solution to an Automotive Control System Benchmark. In *Proceedings 15<sup>th</sup> IEEE Real-Time Systems Symposium*, San Juan, Puerto Rico, Los Alamitos, CA, USA: IEEE Computer Society Press.
- Koeptz, H. (1998). A Comparison of CAN and TTP. In *Proceedings 15<sup>th</sup> IFAC Workshop on Distributed Computer Control Systems*, Como, Italy, Oxford, UK: Pergamon.
- Koeptz, H. and Grunsteidl, G (1994). TTP – A Protocol for Fault-Tolerant Real-Time Systems. *IEEE Computer*, Vol. 27, No. 1, pp. 14-23.
- Kondraske, G. V., Volz, R. A., Johnson, D. H., Tesar, D., Trinkle, J. C. & Price, C. R. (1993). Network-based infrastructure for distributed remote operations and robotics research. *IEEE Transactions on Robotics and Automation*, 9(5), 702-704.
- Kumar, P. R. (2001). New technological vistas for systems and control: The example of wireless networks. In *IEEE Control Systems Magazine*, pp. 24-37.
- Kurose, J. F., Shwartz, M. and Yemini, Y. (1984). Multiple-Access Protocols and Time-Constrained Communication. In *Computer Surveys*, 16(1):43-70, March.

- Kwoon, S. K., Shin, K. G. and Zheng, Q. (1999). Statistical Real-Time Communication over Ethernet for Manufacturing Automation Systems. In *Proc. IEEE Real-Time Technology and Applications Symposium (RTAS'1999)*, pp. 192-202.
- Lawrenz, W. (1997). *CAN System Engineering: From Theory to Practical Applications*. New York: Springer-Verlag.
- Le Lann, G. (1995). "On Real-Time and Non Real-Time Distributed Computing". Workshop on Distributed Algorithms, Springer Verlag.
- Le Lann, G.; Rivierre, N. (1993). Real-Time Communications over Broadcast Networks: the CSMA-DCR and the DOD-CSMA-CD Protocols. *Rapport de Recherche n° 1863*, INRIA, França, March 1993.
- Lehoczky, J., Sha, L. and Ding, Y. (1989). The Rate Monotonic Scheduling Algorithm: Exact Characterization and Average Case Behaviour. In *Proceedings of the 10<sup>th</sup> IEEE Real-Time Systems Symposium*, pp. 166-171.
- Leite, J. R. E. : "O Modelo de Referência para a Interconexão de Sistema Abertos". *Telebrás*, pp. 11-15, Junho 1985.
- Leung, J. and Whitehead, J. (1982). On the Complexity of fixed-priority Scheduling of Periodic Real Time Tasks. In *Desempenho Evaluation*, Vo. 22, No. 4, pp. 237-250.
- Lian, F., Moyne, J. and Tilbury, D. (2002). Network Design Consideration for Distributed Control Systems. *IEEE Trans. on Control Systems Technology*, Vol.10, No.2, March.
- Lian, F.-L. (2001). Analysis, Design, Modeling and Control of Networked Control Systems, PhD Thesis, University of Michigan
- Lian, F.-L., Moyne, J. R. and Tilbury, D. M. (2001). Analysis and modeling of networked control systems: MIMO case with multiple time delays. In *Proceedings of 2001 American Control Conference*.

- Lian, F.-L., Moyne, J. R. and Tilbury, D. M. (2001). Desempenho evaluation of control networks: Ethernet, ControlNet and DeviceNet. *IEEE Control Systems Magazines*, 21(1):66-83.
- Lian, F.-L., Moyne, J. R. and Tilbury, D. M. (2001). Time delay modeling and sample time selction for networked control systems. In *Proceedings of the ASME Dynamic Systems and Control Division*.
- Liu, C. L. and Layland, J. W. (1973). Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *Journal of ACM*. Vol. 20, No. 1, pp. 46-61. January.
- Lonn, H. (1999). *Synchronization and Communication Results in Safety-Critical Real-time Systems*. PhD thesis, Chalmers University of Thecnology.
- Luck, R. And Ray, A. (1990). An observer-based compensator for distributed delays. *Automatic* 26(5), 903-908.
- Malcon, N. and Zhao, W. (19??). Advances in Hard Real-Time Communication with Local Area Networks.
- Malcon, N. and Zhao, W. (1994). The Timed Token Protocol for Real-Time Communications. *IEEE Computer*, pp.35-41.
- Malcon, N. and Zhao, W. (1995). Hard Real-Time Communications in a Multiple Access Protocols. *Real-Time Systems*, pp. 35-77.
- MAP/TOP. (1987). Manufacturing Automation Protocol / Technical Office Protocol, Version 3.0. Users Group of SME.
- Marti, P., Fohler, G., Ramamritham, K. and Fuertes, J.M. (2001). Jitter Compensation in Real-Time Control Systems. *IEEE Real-Time Systems Symposium*, London, UK, December.

- Marti, P., Fuertes, J.M. and Fohler, G. (2001). An Integrated Approach to Real-Time Distributed Control Systems Over Fieldbus. *Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation*. Antibes Juan-les-pins. France. October.
- Marti, P., Fuertes, J.M. and Fohler, G. (2001). Sampling Jitter Compensation in Real-Time Control Systems. *Work-in-progress Session, 13th Euromicro Conference on Real-Time Systems*, Delft, The Netherlands, June.
- Marti, P., Fuertes, J.M. and Fohler, G. (2002). A Control DesempenhoMetric for Real-Time Constraints. *Work-in-progress Session, 14th Euromicro Conference on Real-Time Systems*, Viena, Austria, June.
- Marti, P., Villa, R., Fuertes, J.M. and Fohler, G. (2000). Real Time Scheduling Methods Requirements in Distributed Control Systems. *Proceedings of the 5th IFAC Workshop on Real-Time Programming*, Palma de Mallorca, Spain, May, pp. 101-108.
- Marti, P., Villa, R., Fuertes, J.M. and Fohler, G. (2001). On Real-Time Control Task Schedulability. *Proceedings of European Control Conference*, Porto, Portugal, September 2001, pp. 2227-2232.
- Marti, P., Villa, R., Fuertes, J.M. and Fohler, G. (2001). Stability of On-line Compensation in Real-Time Scheduled Control Tasks. *IFAC Conference on New Technologies for Computer Control*, Hong Kong, November.
- Nilsson, J. (1998). *Real-Time Control Systems with Delays*, PhD Thesis, Lund Institute of Technology, Lund, Sweden.
- Norden, S., Balaji, S., Manimaran, G., Silva, C. and Murthy, R. (1999). Deterministic Protocols for Real-Time Communication in Multiple Access Networks. *Computer Communications*, vol.22, pp. 128-136, Elsevier Press.

- Ogata, K. (1997). *Modern Control Engineering*, 3<sup>rd</sup> ed. Englewood Cliffs, NJ Prentice-Hall cop. ISBN 0-13-261389-1.
- Oliveira, R. S., Farines, J. M. and Fraga, J. S. (2000). *Sistemas de Tempo Real*. Escola de Computação 2000, IME-USP-SBC.
- Overstreet, J. W. & Tzes, A. (1999). An Internet-Based Real-Time Control Engineering Laboratory. *IEEE Control Systems Magazine*, 19(5), 19-34.
- Ozguner, U., Goktas, H., Chan, H., Winkelman, J., Liubakka, M. & Krotolica, R. (1992). Automotive Suspension Control Through a Computer Communication Network. *Proceedings of the 1992 IEEE Conference on Control Applications*. Vol. 2, pp. 859-900, Dayton, OH.
- Pinho, L. M. (2001). A Framework for the Transparent Replication of Real-Time Application. PhD Thesis, FEUP – Univ. Porto.
- Pinho, L. M. and Vasques, F. (1999). Replica Management in Real-Time Ada 95 Applications. *Ada Letters*, vol XIX, n<sup>o</sup>2, pp. 21-27, ACM Press.
- Pinho, L. M. and Vasques, F. (1999). To Ada or Not To Ada: Adaing vs. Javaing in Real-Time Systems. *ACM Ada Letters*, Vol. XIX, N. 4, pp 37-43.
- Pinho, L. M. and Vasques, F. (2001). Atomic Multicast for Reliable CAN Communication. *XIX Simpósio Brasileiro de Redes de Computadores*, Florianópolis, Brasil.
- Pleinavaux, P. (1990). Architecture de Communication pour le Temps Reel Strict. Ph.D. Thesis n<sup>o</sup> 892, Ecole Polytechnique Federale de Lausanne.
- Ramamritham, K. and Stankovic, J. (1994). Scheduling Algorithms and Operating Systems Support for Real-Time Systems. In *Proceedings of the IEEE*, Vol. 82, No. 1, pp. 55-67.

- Ray, A. (1987). Performance Evaluation of Medium Access Control Protocols for Distributed Digital Avionics. *Journal of Dynamic Systems, Measurement and Control*, 109, 370-377.
- Ray, A. and Halevi, Y. (1988). Integrated Communication and Control Systems: Part II- design considerations. *Journal of Dynamic Systems, Measurement and Control*, 109, 370-377.
- Redell, O. and Sanfrindson. (2002). Exact Best-Case Response Time Analysis of Fixed Priority Scheduled Tasks. In *Proceedings of the 14<sup>th</sup> Euromicro Conference on Real-Time Systems*. Vienna, August.
- Rockwell (1997). DeviceNet Product Overview. *Publication DN-2.5*, Rockwell.
- Rosa, P., Vasques, F. and Valette, R. (1994). The Network Transparency Concept in Fieldbus Based Distributed Systems”, In *Proceedings of the International Symposium on Industrial Electronics (ISIE'94)*, Santiago, Chile, pp. 247-251.
- Ryu, M., Hong, S. and Saksena, M. (1997). Streamlining real-time controller design: From Desempenho specifications to end-to-end timing constraints. In *Proceedings of the Third IEEE Real-Time Technology and Applications Symposium*, Montreal, Canada, pp.91-99.
- SAE J1939 (2000). Recommended Practice for Truck and Bus Control and Communication Network, Soc. of Automotive Engineers, Warrendale, Pa. 2000; <http://www.sae.org/products/j1939.htm>.
- SAE. (1994). SAE J2056 Class C Application Requirement Considerations. In *SAE Handbook*, Vol. 2 Warrendale, PA, USA: Society of Automotive Engineers, 1994, pp. 23.366-23-272.
- Santos, M. M. D., Stemmer, M. R. e Vasques, F. (2002). Análise de Escalonabilidade de Mensagens em uma Rede CAN Aplicada a uma Aeronave não Tripulada – O

- Helicóptero HELIX, IV Workshop de Tempo Real (WTR'02), Búzios, Brazil, pp. 29-37, 2002.
- Santos, M. M. D., Vasques, F. and Stemmer, M. R. (2004). O Impacto da Comunicação em Sistemas de Controle via Redes CAN. In *Proceedings of XV Brazilian Automatic Congress*. Gramado, Brazil.
- Santos, M. M. D.; Ataíde, F. H.; Vasques, F.; Stemmer, M. R. (2003): "Compensação de Atrasos em Sistemas de Controle via Redes - Rede CAN." In *Proceedings of IEEE Seminary in 1st Power Systems, Automation and Control*. Juiz de Fora, Brazil.
- Santos, M. M. D.; Mariano Junior, A. P.; Renó computacional, D. (2004): "The Real-Time Systems Simulator Using the LOGO Language." In *Proceedings of IEEE World Congress on Engineering and Technology Education*. Guarujá, Brazil.
- Santos, M. M. D.; Renó computacional, D.; Mariano Junior, A. P. (2003): "Análise de Escalonabilidade do Kernel de Tempo Real S.Ha.R.K." In *Proceedings of III Congresso Brasileiro de Computação*. Itajaí, Brazil.
- Santos, M. M. D.; Stemmer, M. R.; Vasques, F. (2002): "Análise de Escalonabilidade de Mensagens para uma Rede CAN Aplicada a uma Aeronave não Tripulada - O Helicóptero HELIX." In *Proceedings of IV Workshop of Real-Time Systems*. Búzios, Brazil.
- Santos, M. M. D.; Stemmer, M. R.; Vasques, F. (2002): "Schedulability Analysis of Messages in a CAN Network Applied to an Unmanned Airship." In *Proceedings of 28th IEEE Conference on Industrial Electronic Society*. Madrid, Espanha.
- Santos, M. M. D.; Stemmer, M. R.; Vasques, F. (2002): "Uma Rede CAN Aplicada ao Controle de uma Aeronave não Tripulada - O Helicóptero HELIX." In *Proceedings of XIV Brazilian Automatic Congress*. Rio Grande do Norte, Brazil.

- Santos, M. M. D.; Vasques, F.; Stemmer, M. R. (2003): "Evaluation of the Timing Properties of Two Control Networks: CAN and PROFIBUS." In *Proceedings of IEEE International Symposium in Industrial Electronics*. Rio de Janeiro, Brazil.
- Santos, M. M. D.; Vasques, F.; Stemmer, M. R. (2003): "Avaliação das Propriedades Temporais de Duas Redes de Controle: CAN e PROFIBUS." In *Acta Scientiarum*. Universidade Estadual de Maringá (UEM), Maringá, Brazil.
- Santos, M. M. D.; Vasques, F.; Stemmer, M. R. (2004): "Analysis of Timing Properties from Network Control Systems." In *Proceedings of 6th Brazilian Workshop on Real-Time Systems*. Gramado, Brazil.
- Santos, M. M. D.; Vasques, F.; Stemmer, M. R. (2004): "O Impacto da Comunicação em Sistemas de Controle via Redes CAN." In *Proceedings of XV Brazilian Automatic Congress*. Gramado, Brazil.
- Santos, M. M. D.; Vasques, F.; Stemmer, M. R. (2004): "Design of Network Control Systems – CAN and Token-Passing." In *Proceedings of VI IEEE International Conference on Industrial Applications*. Joinville, Brazil.
- Santos, M. M. D.; Vasques, F.; Stemmer, M. R.; Ataíde, F. H. (2003): "O Impacto da Rede no Atraso de Controle em um Sistema de Controle via Redes." In *Proceedings do III Congresso Brasileiro de Computação*. Itajaí, Brazil.
- Sathaye, S. S., Kish, W. S. and Strosnider, J. K. (1993). Scheduling Real –Time traffic in Packet-Switched Network. Ph.D Thesis, Carnegie-Mellon University, Pittsburgh.
- Seto, D., J. P. Lehoczky, L. Sha and K.G. Shin (1996). On task schedulability in real-time control systems. In Proc. 17<sup>th</sup> IEEE Real-Time Systems Symposium, pp. 13-21. Washington, DC.

- Sha, L., Rajkumar, R. and Lehoczky, J. (1990). Priority Inheritance Protocols: an Approach to Real-Time Synchronisation. In *IEEE Transactions on Computers*, Vol. 39, No. 9, pp. 1175-1185.
- Shin, K. G. and Chow, C.-C. (1996). Design and Evaluation of Real-Time Communication for FieldBus – Based Manufacturing Systems. *IEEE Transactions on Robotics and Automation*, vol.12, nº 3.
- Soares, L. F. G., Lemos, G. and Colcher, S. (1995). Redes de Computadores: das LANs, MANs e WANs às redes ATM. Editora Campus, 2º Edição.
- Stemmer, M. R. (1991). Einsatzmöglichkeiten digitaler Feldbussysteme in geschlossenen, maschineninternen Regelkreisen. Tese de Doutorado, RWTH-Aachen.
- Stemmer, M. R. (1999). Rede Locais Industriais. Apostila do Curso de Pós-Graduação em Eng. Elétrica, LCMI-DAS-UFSC.
- Tanenbaum, A. S. (1996). Computer networks. 3rd Edition, Prentice-Hall.
- Tarn, T.-J. & Xi, N. (1998). Hanning and Control of Internet-Based Teleoperation. *Proceedings of SPIE Telemanipulator and Telepresence Technologies V*, Vol. 3524(pp. 184-193), Boston, MA.
- Time-Triggered Protocol (TTP/C), Version 1.0, TTA Group, Vienna; <http://ttagroup.org/ttp/specification.htm>.
- Tindell, K. (2000). Add a Real-Time Kernel and free up Processor Time in Your Automotive ECU. *The Embedded Systems Show*, London, U.K, May 24-25.
- Tindell, K. and Burns, A. (19??). Guaranteeing Message Latencies on Control Area Network (CAN).

- Tindell, K. and Burns, A. (1994). Analysing Real-Time Communications: Controller Area Network (CAN)". *Proceedings 15<sup>th</sup> IEEE Real-Time Systems Symposium*, pp. 259-265.
- Tindell, K. and Burns, A. (1994). Guaranteed Message Latencies for Distributed Safety Critical Hard Real-Time Networks. *Technical Report YCS 229*, Department of Computer Science, University of York, England.
- Tindell, K. and Clark, J. (1994). Holistic Schedulability Analysis for Distributed Hard Real Time Systems. In *Microprocessor and Microprogramming* 40:117-134.
- Tindell, K.(2000). Embedded Systems in the Automotive Industry. *ESC Spring 2000*, Chicago, EUA.
- Tindell, K., Burns, A. and Wellings A. J. (1995). Analysis of Hard Real-Time Communications. *The Journal of Real-Time Systems*, nr. 9, pp. 147-171, Kluwer Academic Publishers.
- Tindell, K., Burns, A. and Wellings, A. (1995). Calculating Controller Area Network (CAN) Message Response Time. In *Control Engineering Practice*, Vol. 3, No. 8 - pp. 1163-1169.
- Tipswan, Y. & Chown, M.-Y. (2001). Network-based controller adaptation based on QoS negotiation and deterioration. *The 27th annual conference of the IEEE industrial electronics society (IECON 01)* Vol. 3 (pp. 1794-1799). Denver, CO.
- Tipswan, Y. & Chown, M.-Y. (2002). Gain Adaptation of Networked Mobile Robot to Compensate QoS Deterioration. *The 28th annual conference of the IEEE industrial electronics society (IECON 02)* 4 (pp. 3146-3151). Sevilla, Spain.
- Tovar E. and Vasques F. (1999a). Cycle Time Properties of the PROFIBUS Timed Token Protocol. In *Computer Communications*, Elsevier Science, n°22, pp. 1206-1216.

- Tovar E. and Vasques F. (1999b). Real-Time Fieldbus Communications Using PROFIBUS Networks. In *IEEE Transactions on Industrial Electronics*, Vol. 46, n°6, pp. 1241-1251.
- Tovar, E. (1999). Supporting Real-Time Communications with Standard Factory-Floor Networks, PhD Thesis, FEUP – Univ. Porto.
- Tovar, E. and Vasques, F. (1998). A Communication Support for Real-Time Distributed Computer Controlled Systems, In *Proceedings of the IEE International Workshop on Discrete Event Systems (WODES'98)*, Cagliari, Italy, 26-28, pp.178-183.
- Tovar, E. and Vasques, F. (1998). Guaranteeing Real-Time Message Deadlines in Profibus Networks. In *Proceedings of the 10th Euromicro Workshop on Real-Time Systems*, Berlin, Germany, 17-19, pp.79-86.
- Tovar, E. and Vasques, F. (1998). Pre-Run-Time Schedulability Analysis of P-NET FieldBus Networks. In *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (IECON'98)*, Aachen, Germany, 31 August - 4, pp.236-241.
- Tovar, E. and Vasques, F. (1998). Scheduling Real-Time Communications with P-NET. In *Digest of the IEE Real-Time Systems Colloquium*, York, England, pp.9/1-9/5. IEE Digest 98/306.
- Tovar, E. and Vasques, F. (1999). Analysis of the Worst-Case Real-Token Rotation Time in Profibus Networks. In *Proceedings of the 3rd International Conference on Fieldbus Systems and their Applications (FET'99)*, Magdeburg, Germany, 20-21.
- Tovar, E. and Vasques, F. (1999). Contributions for the Worst-Case Response Time Analysis of Real-Time Sporadic Traffic in WorldFIP Networks. In *Proceedings of the WIP session of the 11th Euromicro Conference on Real-Time Systems (RTS'99)*, York, England, 9-11.

- Tovar, E. and Vasques, F. (1999). Cycle Time Properties of the PROFIBUS Timed Token Protocol. *Computer Communications*, vol. 22, pp. 1206-1216, Elsevier Press.
- Tovar, E. and Vasques, F. (1999). Engineering Real-Time Applications with P-NET Networks. In *Proceedings of the 6th International P-NET Conference*, Vienna, Austria, 25-26.
- Tovar, E. and Vasques, F. (1999). From Task Scheduling in Single Processor Environments to Message Scheduling in a Profibus Fieldbus Network. In *Proceedings of the 7th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'99)*, San Juan, Puerto Rico, 12-13.
- Tovar, E. and Vasques, F. (1999). Setting Target Rotation Time in Profibus Based Real-Time Distributed Applications. In *Proceedings of the 15th IFAC Workshop on Distributed Computer Control Systems (DCCS'98)*, Como, Italy, 9-11, pp.1-6. Published as Distributed Computer Control Systems 1998 by Elsevier Science.
- Tovar, E., Vasques, F. and Cardoso, A. (1999). Guaranteeing DCCS Timeliness Requirements Using P-NET Fieldbus Networks", In *Proceedings of the 7th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'99)*, Barcelona, Spain, 18-22.
- Tracery, N. (2000). True Real-Time Embedded Systems Engineering – Building Hard Real-Time Systems Using Deadline Monotonic Analysis. ESC Europe, Maastricht.
- Vasques, F. (1996). Sur l'intégration de mécanismes d'ordonnancement et de communication dans la sous – couche MAC de réseaux locaux temp réel. Dr. Thesis, LASS, Setembro.
- Vasques, F. and Juanole, G. (1994). Pre Run-Time Schedulability Analysis in Fieldbus Networks. In *Proceedings of the International Conference on Industrial*

- Electronics, Control and Instrumentation (IECON'94)*, Bologna, Italy, pp. 1200-1204.
- Vasques, F. and Pinho, L. M. (2001). Atomic Multicast Protocols for Reliable CAN Communication Simposio Brasileiro de Redes de Computadores – SBRC'2001, Florianopolis – Brasil, pp. 194-209.
- Vivanco, G. (1999). Projeto de uma Camada de Enlace de Dados para um Sistema de Comunicação Tempo Real. Master Thesis, LCMI-UFSC.
- Walsh, G. and Ye, H. (2001). Scheduling of networked control systems. In *IEEE Control Systems Magazine*, pp. 57-65.
- Walsh, G., Beldiman O. and Bushnell, L. (1999a). Error encoding algorithms for networked control systems. In *Proc. IEEE Conf. Decision and Control*. Phoenix, AZ., pp. 4993-4938.
- Walsh, G., Beldiman, O. and Bushnell, L. (1999b). Asymptotic behavior of networked control systems. In *Proc. IEEE Int. Conf. Control and Applications*. Hawaii, pp. 1448-1454.
- Walsh, G., Ye, H. and Bushnell, L. (1999c). Stability analysis of networked control systems. In *Proc. Amer. Control Conf.*, San Diego, CA., pp. 2876-2880.
- Wargui, M., Tadjine, M. & Rachid, A. (1996). Stability of Real Time Control of an Autonomous Mobile Robot. *The fifth IEEE International Workshop on Robot and Human Communicatio*. Tsukuba, Japan, pp. 311-316.
- Witternmark, B., Nilsson, J. and Torngren, M. (1995). Timing problems in real-time control systems. In *Proceedings of the 1995 American Control Conference*. Seattle, Washington.
- Wood, G. G. (1988). International Standards Emerging for Fieldbus. *Control Engineering*, Vol. 2, pp. 22-25.

- Yanaga, E. and Fonseca, K. V. O. (2000). A P-NET Schedulability Analysis Tool. *II Workshop de Sistemas em Tempo Real*, Florianópolis, Brasil.
- Ye, H., Walsh, G. and Bushnell, L. (2000). Wireless local area networks in manufacturing industry. In *Proc. Amer. Control Conf.*, Chicago, IL., pp. 2363-2367.
- Zhang, W. (2001). Stability Analysis of Networked Control Systems, PhD Thesis, Case Western Reserve University.
- Zhang, W., Branicky, M. S. and Phillips, S. M. (2001). Stability of networked control systems". In *IEEE Control Systems Magazine*, pp. 84-99.
- Zhi-Ping, W., Guang-Se, X., Jin, L., Ming-Zhi, L. and ZHOU, W. (19??). A Hard Real-Time Communication Control Protocol Based on the Ethernet.
- Zimmerman, H. (1980). OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transaction on Communication*. Vol.28, pp. 425-432.
- Zuberi, K. and Shin, K. (1997). Scheduling messages on Controller Area Network for Real-Time CIM Applications. In *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 2, pp 310-314.
- Zuberi, K. M. and Shin, K. G. (1998). Non-Preemptive Scheduling of Messages on Controller Area Network for Real-Time Control Applications.