EDUARDO PACIÊNCIA GODOY

**Desenvolvimento de Sistemas de Controle via Rede (NCS)
para Aplicações em Redes com Protocolo CAN**

Tese definitiva apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo para obtenção do Título de Doutor em Engenharia.

Programa de Pós Graduação: Engenharia Mecânica
Área de Concentração: Manufatura
Orientador: Prof. Tit. Arthur José Vieira Porto

**São Carlos
2011**

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

G621d Godoy, Eduardo Paciência.
Desenvolvimento de sistemas de controle via rede (NCS)
para aplicações em redes com protocolo CAN / Eduardo
Paciência Godoy; orientador Arthur José Vieira Porto. São
Carlos, 2011.

Tese (Doutorado-Programa de Pós-Graduação em
Engenharia Mecânica e Área de Concentração em Manufatura)
-- Escola de Engenharia de São Carlos da Universidade de
São Paulo, 2011.

1. Sistemas de controle via redes - NCS. 2. Redes CAN.
3. Ferramentas de simulação. 4. Período de amostragem. 5.
Controle adaptativo. 6. Técnica de identificação. I.
Título.

FOLHA DE JULGAMENTO

Candidato: Engenheiro **EDUARDO PACIÊNCIA GODOY**.

Tese defendida e julgada em 21/03/2011 perante a Comissão Julgadora:

Prof. Titular **ARTHUR JOSE VIEIRA PORTO** – (Orientador)
(Escola de Engenharia de São Carlos/USP)

APROVADO

Prof. Dr. **MARIO LUIZ TRONCO**
(Escola de Engenharia de São Carlos/USP)

APROVADO

Prof. Dr. **NEWTON MARUYAMA**
(Escola Politécnica/USP)

APROVADO

Prof. Dr. **EDILSON REIS RODRIGUES KATO**
(Universidade Federal de São Carlos/UFSCar)

APROVADO

Prof. Dr. **GERALDO NUNES SILVA**
(Universidade Estadual "Júlio de Mesquita Filho"/UNESP/campus de São José do Rio Preto)

APROVADO

Prof. Associado **JONAS DECARVALHO**
Coordenador do Programa de Pós-Graduação em
Engenharia Mecânica

Prof. Associado **PAULO CÉSAR LIMA SEGANTINE**
Presidente da Comissão de Pós-Graduação

**'Feliz aquele que transfere o que sabe
e aprende o que ensina'
(William Shakespeare)**

**'A mente que se abre a uma nova ideia,
nunca volta ao seu tamanho original'
(Albert Einstein)**

*A minha esposa Monica,
pelo amor e dedicação;
Aos meus pais, Antônio e Terezinha,
por todos os ensinamentos.*

AGRADECIMENTOS

Ao Professor Arthur José Vieira Porto pela oportunidade oferecida, pela amizade, companheirismo, dedicação e orientação.

Ao Professor Ricardo Yassushi Inamasu e a todos os amigos e funcionários do Departamento de Engenharia Mecânica da EESC/USP, em especial aos do Laboratório de Simulação e Controle, pela convivência e colaboração.

À Professora Dawn M. Tilbury e amigos da Universidade de Michigan nos Estados Unidos pela oportunidade e aprendizagem durante a realização do estágio de doutorado sanduíche.

Aos amigos da minha cidade Rio Claro e a minha família pelo apoio e incentivo nas horas de desânimo e pela alegria e descontração nas horas de felicidade.

À Fundação de Amparo a Pesquisa do Estado de São Paulo - FAPESP pela concessão da Bolsa de Doutorado (Processo 2007/05019-0) e pelo apoio financeiro junto ao Projeto de Auxílio a Pesquisa (Processo 2008/03481-1) imprescindíveis para a realização desta pesquisa.

RESUMO

GODOY, E. P. **Desenvolvimento de Sistemas de Controle via Rede (NCS) para Aplicações em Redes com Protocolo CAN**. 252f. Tese de Doutorado – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2011.

Sistema de Controle via Rede (NCS) é um sistema de controle distribuído onde os sensores, atuadores e controladores estão alocados fisicamente em locais separados e são conectados através de uma rede de comunicação industrial. O NCS representa a evolução das arquiteturas de controle em rede, fornecendo maior modularidade e descentralização do controle, facilidade de diagnóstico e manutenção e menor custo. O desafio no desenvolvimento de um NCS é contornar os efeitos degenerativos causados por fatores que afetam o seu desempenho e estabilidade. Entre estes fatores estão o período de amostragem dos sinais, a perda de informações transmitidas na rede e os atrasos de comunicação. Buscando superar este desafio, este trabalho apresenta o desenvolvimento de NCS para aplicações em redes CAN baseado no uso da simulação e na proposta de uma estratégia de controle. A utilização de ferramentas de simulação de NCS, selecionadas através de um estudo comparativo e qualitativo, permitiu analisar o impacto de fatores degenerativos no desempenho de controle e estabilidade de NCS. Essa análise por simulação permitiu evidenciar o período de amostragem como o fator de maior influência para o projeto de NCS em redes CAN. Para superar o problema do período de amostragem, uma estratégia de controle adaptativo foi proposta. Essa estratégia usa informações de saída do NCS para automaticamente adaptar o período de amostragem das mensagens, garantindo desempenho de controle e diminuindo significativamente a ocupação da rede CAN. Experimentos realizados em uma Plataforma de Pesquisa sobre NCS demonstraram a confiabilidade e robustez do uso da estratégia de controle adaptativo, mesmo em condições extremas de operação da rede CAN. Os experimentos também permitiram comprovar a eficácia de uma técnica de identificação de NCS desenvolvida, que apresenta a vantagem de utilizar informações disponíveis na rede para obtenção de um modelo do NCS com precisão aceitável.

Palavras-chave: Redes CAN, Sistemas de Controle via Redes – NCS, ferramentas de simulação, período de amostragem, controle adaptativo, técnica de identificação

ABSTRACT

GODOY, E. P. **Development of Networked Control Systems for Applications in CAN-Based Networks**. 252f. PhD Thesis – São Carlos School of Engineering, University of São Paulo, São Carlos, 2011.

Networked control system (NCS) is a distributed control system where the sensors, actuators and controllers are physically separated and connected through an industrial communication network. The NCS represents the evolution of networked control architectures providing greater modularity and control decentralization, maintenance and diagnosis ease and lower cost of implementation. The challenge in the development of NCS is to overcome the degenerative effects of factors which affect its performance and stability. Among these factors are the sampling time, the loss of information on the network and the network delays. Aiming to overcome this challenge, this work presents the development of NCS for applications in CAN-Based networks based on the simulation use and in a control strategy proposal. The use NCS simulation tools, selected by a comparative and qualitative study, allowed to analyze the impact of degrading factors in the NCS control performance and stability. This analysis using simulation highlighted the message sampling time as factor with the biggest influence for the design of CAN-based NCS. To overcome the sampling time problem, an adaptive control strategy was proposed. This strategy uses the NCS output to automatically adapt the message sampling time, ensuring NCS control performance and stability and providing significant reduction of the CAN network load. Experiments carried out on a NCS Research Platform demonstrated the reliability and robustness of the adaptive control methodology application, even under worst case conditions of operation of the CAN-based network. Experiments have also proved the effectiveness of a model identification technique developed for NCS, which presents the advantage of using information available on the network to obtain the NCS model with acceptable accuracy.

Keywords: CAN Protocol, Networked Control System – NCS, simulation tools, message sampling time, adaptive control, and model identification technique

LISTA DE FIGURAS

Figura 1. Estrutura de um Sistema de Controle via Redes (NCS).....	26
Figura 2. Pesquisa de Desenvolvimento de Sistema de Controle via Redes (NCS)	27
Figura 3. NCS com Estrutura Direta (Adaptado de Tipsuwan & Chow, 2003).....	34
Figura 4. NCS com Estrutura Hierárquica (Adaptado de Tipsuwan & Chow, 2003)	34
Figura 5. Composição do Atraso de Comunicação em NCS (Adaptado de Lian, Moyne & Tilbury, 2002).....	36
Figura 6. Diagrama Temporal da Transmissão de uma Mensagem CAN (Adaptado de Lian, Moyne & Tilbury, 2001)	37
Figura 7. Efeito do Atraso de Comunicação (t) no Desempenho e Estabilidade de um NCS (Tipsuwan & Chow, 2003)	40
Figura 8. Relação entre Período de Amostragem e Desempenho em NCS (Adaptado de Lian et al., 2006)	41
Figura 9. Influência da Perda de Pacotes ou Mensagens no Desempenho de um NCS (Valdivia & Souza, 2007).....	43
Figura 10. Fluxograma de Utilização da Ferramenta AIDA	62
Figura 11. Descrição Geral da Arquitetura da Ferramenta AIDA.....	64
Figura 12. Estrutura de Diagramas do Ambiente <i>Aidasign</i> da Ferramenta AIDA	67
Figura 13. Diagrama de Representação dos Parâmetros de uma Tarefa na ferramenta TORSCHÉ (Kutil et al., 2007)	75
Figura 14. Utilização de Gráfico para Definição de Relações de Precedência entre um Conjunto de Tarefas (Kutil et al., 2007)	77
Figura 15. Gráficos de Tarefas e Conjunto de Tarefas na Ferramenta TORSCHÉ.....	83
Figura 16. Gráfico de Saída de Escalonamento e Dados sobre Tarefas na Ferramenta TORSCHÉ	83
Figura 17. Modelos requeridos para a Ferramenta Jitterbug: (a) Modelo de Sinalização e (b) Modelo Temporal (Cervin & Lincoln, 2006)	87
Figura 18. Comandos da Ferramenta Jitterbug para Definição dos Modelos e Cálculo da Função Custo de Desempenho de um NCS (Cervin & Lincoln, 2006).....	88
Figura 19. Desempenho de Controle de um NCS de acordo com diferentes Valores da Função Custo.....	92
Figura 20. Qualidade de Controle de um NCS relacionado com os Parâmetros Atraso de Comunicação e Período de Amostragem (Cervin & Lincoln, 2006)	93
Figura 21. Arquitetura da Ferramenta PiccSIM (Björkbom, Nethi & Kohtamäki, 2010).....	97
Figura 22. Biblioteca de Blocos da Ferramenta PiccSIM (Björkbom, Nethi & Kohtamäki, 2010).....	98
Figura 23. Modelo de um NCS na Ferramenta PiccSIM: (a) Blocos do <i>Simulink</i> disponíveis para Modelagem e (b) Interface de Configuração do Bloco de Modelagem Controlador (Nethi et al., 2007b)	102
Figura 24. Interface Gráfica para Desenvolvimento e Configuração da Rede de Comunicação na Ferramenta PiccSIM: (a) Exemplo de Interface de Configuração de Parâmetros de uma Rede sem Fio e (b) Interface de Geração do <i>Script</i> de Configuração (Nethi et al., 2007b) ...	103
Figura 25. Utilização da Ferramenta PiccSIM pela Internet: (a) Interface Padrão para Simulação de NCS e (b) Interface para Realização dos Experimentos Remotos de NCS ((Nethi et al., 2007a).....	103

Figura 26. Exemplo de Resultado de Simulação de um NCS na Ferramenta PiccSIM (Nethi et al., 2007a).....	104
Figura 27. Blocos Básicos da Biblioteca da Ferramenta NCS_Simu	106
Figura 28. Diagrama referente ao Atraso de Comunicação devido ao Não-Sincronismo de Amostragem de Mensagens um em um NCS (Yang, 2006b)	107
Figura 29. Estrutura de um NCS na Ferramenta NCS_Simu (Adaptado de Yang, 2006a) ...	108
Figura 30. Exemplo de Modelo de NCS na Ferramenta NCS_Simu (Yang, 2006a).....	110
Figura 31. Visualização de um Vetor de Valores de Atraso de Comunicação Utilizado na Ferramenta NCS_Simu (Yang, 2006a)	112
Figura 32. Comandos da Ferramenta NCS_Simu para Definição dos Parâmetros de Simulação e Inicialização de Variáveis (Yang, 2006a)	113
Figura 33. Exemplo de Simulações de um NCS para Controle de Pêndulos na Ferramenta NCS_Simu: (a) Resultados Iniciais do NCS e (b) Resultados após Projeto do Controlador do NCS (Yang, 2006a).....	114
Figura 34. Blocos da Biblioteca da Ferramenta <i>TrueTime</i> (Ohlin, Henriksson & Cervin, 2010)	116
Figura 35. Tela de Configuração do Bloco de Kernel (Ohlin, Henriksson & Cervin, 2010)	118
Figura 36. Comparação entre as Formas de Implementação de Atividades no <i>TrueTime</i>	119
Figura 37. Configuração dos Blocos de Rede (<i>TrueTime</i> Network e <i>TrueTime</i> Wireless) (Cervin et al., 2003)	121
Figura 38. Tela de Configuração dos Blocos de Transmissão de Mensagens	122
Figura 39. Tela de Configuração do Bloco <i>Battery</i> da Ferramenta (Cervin et al., 2003).....	123
Figura 40. Arquitetura de Desenvolvimento da Ferramenta <i>TrueTime</i>	124
Figura 41. Modelo de Projeto de um NCS na Ferramenta <i>TrueTime</i>	126
Figura 42. Gráficos do Desempenho de Controle (acima) e do Sinal de Saída do Controlador (abaixo) da Ferramenta <i>TrueTime</i> (Cervin et al., 2003)	128
Figura 43. Gráficos de Monitoramento de Utilização de Recursos da Rede (acima) e do Processador (abaixo) da Ferramenta <i>TrueTime</i> (Cervin et al., 2003)	129
Figura 44. Estrutura Definida para Plataforma de NCS com Rede CAN	136
Figura 45 – Mecanismo de Acesso ao Meio do CAN (Adaptado de CIA, 2005).....	138
Figura 46 – Formato dos Quadros de Mensagens CAN (Adaptado de SOUSA, 2002)	139
Figura 47. Estrutura de Hardware da Plataforma de NCS com Rede CAN.....	140
Figura 48. Diagrama do Circuito da Interface CAN ou ECU (Sousa, 2002).....	141
Figura 49. Imagem da Interface CAN Utilizada (Sousa, 2002).....	142
Figura 50. Estrutura de <i>Software</i> da Plataforma de NCS com Rede CAN	144
Figura 51. Malha de Controle de Velocidade de Motor DC adaptado para a Plataforma de NCS com Rede CAN	148
Figura 52. Malha de Controle de Posição de Motor DC construído para a Plataforma de NCS com Rede CAN: (a) Vista Geral, (b) Conexão da ECU na Rede CAN e Circuito de Leitura de Encoder e (c) Circuito de Acionamento e Acoplamento do Kit da Maxon Motor	151
Figura 53. Malha de Controle de Nível de Reservatório adaptado para a plataforma de NCS com Rede CAN: (a) Vista Geral, (b) ECU e Acoplamento da Eletrobomba de Água e (c) Conexão do Sensor de Pressão para Medição do Nível de Água	153
Figura 54. Malha de Controle de Temperatura construída para a Plataforma de NCS com Rede CAN: (a) Vista Geral e (b) ECU, Circuitos de Acionamento do Aquecedor e Leitura do Sensor de Temperatura.....	156
Figura 55. Malha de Controle de Esteira de Movimentação construída para a Plataforma de NCS com Rede CAN: (a) Vista Geral, (b) Conexão na ECU na Rede CAN e Sensores de Presença e (c) Acoplamento do Motor de Movimentação da Esteira	159
Figura 56. Estrutura de Implementação de um NCS	163

Figura 57. Esquemático de um Sistema de Controle com Controlador PID	167
Figura 58. Estrutura de um Controlador PID Padrão	168
Figura 59. Estrutura de um Controlador PID com Ponderação de Referência.....	173
Figura 60. Estrutura de um Controlador PID com <i>Anti-Windup</i> da Ação Integral	175
Figura 61. Estrutura do Controlador PID Completo Definido para NCS.....	176
Figura 62. Efeito do Período de Amostragem e da Taxa de Utilização da Rede no Desempenho de um NCS com Rede CAN	178
Figura 63. Estrutura de Aplicação da Estratégia de Controle Adaptativo para NCS CAN....	181
Figura 64. Ciclo de Operação e Implementação da Estratégia de Controle Adaptativo para NCS CAN	183
Figura 65. Estrutura para Estimação e Validação do Modelo Matemático do NCS Analisado	185
Figura 66. Esquemático do Modelo Completo da Plataforma de NCS com Rede CAN na Ferramenta <i>TrueTime</i>	191
Figura 67. Desempenho do NCS para Controle de Velocidade de Motor obtido por Simulação: Resposta a Degrau e Sinal de Controle	193
Figura 68. Desempenho do NCS para Controle de Posição de Motor obtido por Simulação: Resposta a Degrau e Sinal de Controle	193
Figura 69. Desempenho do NCS para Controle de Temperatura obtido por Simulação: (a) Resposta a Degrau e Sinal de Controle	194
Figura 70. Desempenho do NCS para Controle de Nível obtido por Simulação: Resposta a Degrau e Sinal de Controle.....	194
Figura 71. Desempenho do NCS de Controle de Velocidade para Métodos de Sintonia do PID	195
Figura 72. Desempenho do NCS para Controle de Velocidade obtido da Plataforma.....	196
Figura 73. Desempenho do NCS para Controle de Posição obtido da Plataforma	196
Figura 74. Desempenho do NCS para Controle de Temperatura obtido da Plataforma	197
Figura 75. Desempenho do NCS para Controle de Nível obtido da Plataforma.....	197
Figura 76. Comparação entre os Resultados Simulados na Ferramenta <i>TrueTime</i> e Reais Obtidos para o NCS de Controle de Posição utilizando Controlador PD com $K=3$ e $T_d=0,001$	198
Figura 77. Comparação entre os Resultados Simulados na Ferramenta <i>TrueTime</i> e Reais Obtidos para o NCS de Controle de Nível utilizando Controlador PI com $K=10$ e $T_i=0,55$	199
Figura 78. Desempenho do NCS de Controle de Velocidade para Diferentes Períodos de Amostragem de Mensagens.....	201
Figura 79. Gráfico obtido com a Utilização da Ferramenta Jitterbug para o NCS de Controle de Velocidade de Motor DC da Plataforma	202
Figura 80. Sensibilidade do NCS para Controle de Velocidade de Motor DC da Plataforma sob Diferentes Condições de Período de Amostragem e Atraso de Comunicação	204
Figura 81. Sensibilidade do NCS para Controle de Posição de Motor DC da Plataforma sob Diferentes Condições de Período de Amostragem e Atraso de Comunicação.....	205
Figura 82. Sensibilidade do NCS para Controle de Temperatura da Plataforma sob Diferentes Condições de Período de Amostragem e Atraso de Comunicação	206
Figura 83. Sensibilidade do NCS para Controle de Nível da Plataforma sob Diferentes Condições de Período de Amostragem e Atraso de Comunicação	207
Figura 84. Comparação entre os Resultados Reais e Simulados para Experimentos de Estimação do Modelo do NCS para Controle de Velocidade em Malha Aberta: (a) Degrau de 6V, (b) Degrau de 9V	210

Figura 85. Comparação entre os Resultados Reais e Simulados para o Experimento de Estimação do Modelo do NCS para Controle de Velocidade em Malha Fechada com Controlador PI.....	211
Figura 86. Comparação entre os Resultados Reais e Simulados para o Experimento de Estimação do Modelo do NCS para Controle de Nível em Malha Fechada com Controlador PI	212
Figura 87. Estratégia de Controle Adaptativo no NCS de Controle de Velocidade: Resposta ao Degrau e Variações do Período de Amostragem e da Ocupação da Rede CAN durante a Operação do NCS.....	215
Figura 88. Aplicação Simultânea da Estratégia de Controle Adaptativo nos NCS da Plataforma: Respostas ao Degrau dos NCS e Variações dos Períodos de Amostragem e da Ocupação da Rede CAN	216
Figura 89. Versão Ampliada dos Resultados da Figura 88 para o NCS de Controle de Temperatura: Situação de Reinicialização do PA do NCS	218
Figura 90. Comparação do Desempenho de Controle do NCS de Velocidade Utilizando Período de Amostragem (PA) Constante e Variável (ECA).....	219
Figura 91. Desempenho de Controle dos NCS da Plataforma Obtidos Utilizando PA=10ms (Ocupação Total da Rede CAN de 37%)	220
Figura 92. Desempenho dos NCS da Plataforma Obtidos Utilizando PA=10ms e Ocupação Extra da Rede CAN (Ocupação Total de 92%)	221
Figura 93. Resultados do Experimento de Pior Caso de Aplicação da Estratégia de Controle Adaptativo nos NCS da Plataforma: Respostas ao Degrau dos NCS e Variações dos Períodos de Amostragem e da Ocupação da Rede CAN	222
Figura 94. Comparação de Desempenho do NCS de Velocidade e Avaliação de Aplicação da ECA.....	223
Figura 95. Comparação de Desempenho do NCS de Posição e Avaliação de Aplicação da ECA.....	224
Figura 96. Comparação de Desempenho do NCS de Temperatura e Avaliação de Aplicação da ECA.....	225
Figura 97. Comparação de Desempenho do NCS de Nível e Avaliação de Aplicação da ECA	225

LISTA DE TABELAS

Tabela 1. Dados do NCS Utilizado no trabalho de Tipsuwan & Chow (2003)	40
Tabela 2. Síntese da Avaliação das Ferramentas de Análise, Simulação e Projeto de NCS ..	132
Tabela 3. Informações sobre Controladores Projetados e Desempenho dos NCS	192
Tabela 4. Informação dos Modelos Matemáticos para o NCS de Controle de Velocidade de Motor Utilizado no Experimento de Identificação	209
Tabela 5. Informação dos Modelos Matemáticos para o NCS de Controle de Nível Utilizado no Experimento de Identificação	212
Tabela 6. Resumo dos Parâmetros Utilizados nos NCS da Plataforma de Pesquisa para Experimentos de Aplicação da Estratégia de Controle Adaptativo do Período de Amostragem (PA)	214

LISTA DE ABREVIATURAS E SIGLAS

AIDA	Automatic Control in Distributed Applications
CAN	Controller Area Network
CIA	CAN in Automation
CSMA	Carrier Sense Multiple Access - Acesso Múltiplo com Detecção de Portadora
ECU	Electronic Control Unit – Unidade de Controle Eletrônica
EESC	Escola de Engenharia de São Carlos
FAPESP	Fundação de Amparo a Pesquisa do Estado de São Paulo
FDMA	Frequency Division Multiple Access – Acesso Múltiplo por Divisão em Frequência
HIL	Hardware-in-the-loop
ISO	International Organization for Standardization
LQG	Linear Quadratic Gaussian
LMI	Linear Matrix Inequalities - Desigualdades Matriciais Lineares
MAC	Medium Access Control – Controle de Acesso ao Meio Físico
MBC	Model Based Control – Controle Baseado em Modelo
ECA	Estratégia de Controle Adaptativo
MIMO	Multiple Input Multiple Output – Sistema com múltiplas entradas e saídas
MOSFET	Metal Oxide Semiconductor Field Effect Transistor - Transistor de Efeito de Campo de Semicondutor de Óxido Metálico
MPC	Model based Predictive Control – Controle Preditivo baseado em Modelo
NCS	Networked Control System – Sistema de Controle via Rede (também referenciado como Sistema de Controle em Rede)
NCS_Simu	Networked Control System Simulation package
NDBA	Non-Destructive Bitwise Arbitration – Arbitragem Não Destrutiva por Operação Lógica Bit-a-Bit
OSI	Open Systems Interconnection
PCI	Peripheral Component Interconnect
PiccSIM	Platform for Integrated Communications and Control design, Simulation, Implementation and Modeling
PID	Proporcional Integral Derivativo

PWM	Pulse Width Modulation – Modulação por Largura de Pulso
RPC	Rapid Control Prototyping
SAE	The Society of Automotive Engineers
SISO	Single Input Single Output – Sistema com uma entrada e uma saída
SPI	Serial Peripheral Interface
TDMA	Time Division Multiple Access – Acesso Múltiplo por Divisão em Tempo
TORSCH	Time Optimization of Resources, Scheduling
USART	Universal Synchronous Asynchronous Receiver Transmitter
USB	Universal Serial Bus
USP	Universidade de São Paulo
ZOH	Zero Order Hold – Segurador de ordem zero

SUMÁRIO

1.	INTRODUÇÃO.....	25
1.1.	Motivação	25
1.2.	Objetivos.....	29
1.3.	Contribuições	30
1.4.	Estrutura e Conteúdo	31
2.	REVISÃO BIBLIOGRÁFICA.....	33
2.1.	Considerações Iniciais	33
2.2.	Sistemas de Controle via Redes - NCS.....	33
2.3.	Atrasos de Comunicação em NCS.....	35
2.4.	Fatores que Afetam o Desempenho e a Estabilidade de NCS	39
2.5.	Simulação e Análise de NCS	44
2.6.	Metodologias de Projeto e Controle para NCS.....	49
2.7.	Considerações Finais	56
3.	FERRAMENTAS DE ANÁLISE E SIMULAÇÃO DE NCS	57
3.1.	Considerações Iniciais	57
3.2.	Síntese das Ferramentas Analisadas e Critérios Comparativos Adotados.....	57
3.2.1.	AIDA	60
3.2.1.1.	Descrição da Ferramenta.....	60
3.2.1.2.	Arquitetura de Desenvolvimento	63
3.2.1.3.	Critérios de Avaliação da Ferramenta.....	67
3.2.2.	TORSCHÉ.....	72
3.2.2.1.	Descrição da Ferramenta.....	72
3.2.2.2.	Arquitetura de Desenvolvimento	73
3.2.2.3.	Critérios de Avaliação da Ferramenta.....	79
3.2.3.	Jitterbug	85
3.2.3.1.	Descrição da Ferramenta.....	85
3.2.3.2.	Arquitetura de Desenvolvimento	86
3.2.3.3.	Critérios de Avaliação da Ferramenta.....	88
3.2.4.	PiccSIM	94
3.2.4.1.	Descrição da Ferramenta.....	94
3.2.4.2.	Arquitetura de Desenvolvimento	97
3.2.4.3.	Critérios de Avaliação da Ferramenta.....	99
3.2.5.	NCS_Simu	105
3.2.5.1.	Descrição da Ferramenta.....	105
3.2.5.2.	Arquitetura de Desenvolvimento	108
3.2.5.3.	Critérios de Avaliação da Ferramenta.....	109
3.2.6.	TrueTime	115
3.2.6.1.	Descrição da Ferramenta.....	115
3.2.6.2.	Arquitetura de Desenvolvimento	123
3.2.6.3.	Critérios de Avaliação da Ferramenta.....	124
3.3.	Sistematização e Comparação de Ferramentas Analisadas	131
3.4.	Considerações Finais	134
4.	PLATAFORMA DE PESQUISA SOBRE NCS	135
4.1.	Considerações Iniciais	135

4.2.	Estrutura da Plataforma de Pesquisa de NCS com Rede CAN	135
4.2.1.	Rede Industrial – CAN (Controller Area Network)	137
4.2.2.	Arquitetura e Implementação de Hardware	139
4.2.3.	Arquitetura e Implementação de Software	143
4.3.	Descrição e Modelagem das Malhas de Controle da Plataforma de NCS	147
4.3.1.	NCS de Controle de Velocidade de Motor DC	147
4.3.2.	NCS de Controle de Posição de Motor DC	150
4.3.3.	NCS de Controle de Nível de Água de um Reservatório	152
4.3.4.	NCS de Controle de Temperatura	155
4.3.5.	NCS de Controle de Movimentação de Esteira	157
4.4.	Informações de Operação da Plataforma de NCS	159
4.5.	Considerações Finais	161
5.	ESTRATÉGIAS DE PROJETO E CONTROLE DE NCS	162
5.1.	Considerações Iniciais	162
5.2.	Estrutura de um NCS	162
5.3.	Estratégia de Controle PID Discreto	166
5.3.1.	Filtragem da Parte Derivativa	170
5.3.2.	Ponderação de Referência (Setpoint Weighting e Reference Off)	171
5.3.3.	Saturação do Atuador e Anti-Windup da Ação de Controle Integral	174
5.3.4.	Estrutura Definida do Controlador PID para NCS	176
5.4.	Estratégia de Controle Adaptativo do Período de Amostragem	177
5.5.	Técnica de Modelagem e Identificação de NCS	184
5.6.	Considerações Finais	187
6.	APLICAÇÕES DA PLATAFORMA DE PESQUISA NO DESENVOLVIMENTO DE NCS COM REDES CAN	189
6.1.	Considerações Iniciais	189
6.2.	Projeto de NCS Utilizando Ferramentas de Simulação	189
6.3.	Implementação de Controladores PID para Aplicações em NCS	195
6.4.	Verificação e Validação das Ferramentas de Simulação para o Projeto de NCS com Redes CAN	198
6.5.	Análise da Qualidade de Controle de NCS com Redes CAN	200
6.6.	Aplicação da Técnica de Modelagem e Identificação de NCS	209
6.7.	Aplicação da Estratégia de Controle Adaptativo do Período de Amostragem	213
6.8.	Considerações Finais	226
7.	CONCLUSÕES	228
7.1.	Lista de Publicações Realizadas	232
7.2.	Trabalhos Futuros	235
8.	REFERÊNCIAS	238

1. INTRODUÇÃO

1.1. MOTIVAÇÃO

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

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

Aplicações recentes de sistemas de controle distribuído demonstram o surgimento de uma nova abordagem para a utilização de redes *fieldbus*. Nessa abordagem, o controlador e a planta ficam fisicamente separados e são conectados por uma rede de comunicação industrial. O sinal de controle é enviado para o atuador através de uma mensagem encaminhada via rede, enquanto o sensor amostra a saída da planta e retorna a informação para o controlador também transmitindo a informação através da rede. Este tipo de implementação em sistemas industriais, onde as malhas de controle são fechadas sob uma rede de comunicação, como

mostrado na Figura 1, tem sido denominada de Sistema de Controle via Redes (NCS - *Networked Control System*) (YANG, 2006b).

Figura 1. Estrutura de um Sistema de Controle via Redes (NCS)

Os sistemas de manufatura automatizados são compostos por múltiplos subsistemas ou processos. O principal objetivo do controle nessa aplicação é a coordenação e comunicação desses processos de forma que atendam a requisitos individuais e conjuntos, garantindo um bom funcionamento e desempenho global do sistema. Em NCS, os dispositivos conectados têm que compartilhar da melhor maneira possível o meio disponível para a troca de informações e ainda cumprir com requisitos temporais de transmissão de dados. Como solução para este problema, diversos protocolos de comunicação para NCS têm sido pesquisados e desenvolvidos como o CAN (*Controller Area Network*), Profibus, Modbus e Foundation Fieldbus (MOYNE & TILBURY, 2007).

Não importando o tipo de protocolo de comunicação utilizado, o desempenho e a estabilidade de um NCS podem ser afetados pela presença da rede de comunicação e de fatores degenerativos como a amostragem dos sinais e os atrasos de comunicação (*delay*) entre os sensores, atuadores e controladores do sistema (BAILLIEUL & ANTSAKLIS, 2007). Esses atrasos de comunicação são resultados do compartilhamento do meio de comunicação

com outras malhas de controle como também do tempo de processamento requerido para codificação dos sinais e computação dos dados (LIAN, MOYNE & TILBURY, 2002). A inclusão da rede de comunicação no sistema de controle em malha fechada dificulta as tarefas de análise e projeto de NCS. Por causa da variabilidade e da dificuldade de se calcular esses atrasos, os NCS podem apresentar características de sistemas variantes no tempo, tornando essas tarefas de projeto e controle ainda mais desafiadoras (SO, 2003).

Zampieri (2008) apresenta uma revisão sobre as principais contribuições e tendências no desenvolvimento da área de NCS. As pesquisas atuais em NCS têm focado principalmente duas linhas de trabalho. Uma linha na parte relacionada com a análise da influência de fatores inerentes a NCS (atrasos de comunicação, etc.) no desempenho e estabilidade do sistema (JIANYONG, SHIMIN & HAIQING, 2004), bem como o desenvolvimento de ferramentas para simulação desses sistemas (TORNGREN et al., 2006; AL-HAMMOURI, BRANICKY & LIBERATORE, 2008). E outra linha relacionada ao desenvolvimento de metodologias de projeto e controle de NCS para compensar os efeitos dos atrasos de comunicação melhorando o desempenho e garantindo a estabilidade do sistema (TIPSUWAN & CHOW, 2003; HESPANHA, NAGHSHTABRIZI & XU, 2007). A Figura 2 sintetiza essa estrutura de pesquisa e desenvolvimento em NCS.

Figura 2. Pesquisa e Desenvolvimento de Sistema de Controle via Redes (NCS)

De acordo com a Figura 2, a pesquisa em NCS tem focado no estudo de três áreas: redes de comunicação industrial, técnicas de controle e sistemas de tempo real. Assim primeiramente se torna necessário a pesquisa das características da rede de comunicação e do protocolo escolhido, a análise dos fatores e parâmetros relacionados a redes industriais e sistemas de tempo real que exercem influência no NCS proposto e um estudo das técnicas de controle que podem ser aplicadas ao sistema. Já o desenvolvimento de NCS tem apresentado dois componentes principais: ferramentas para análise, modelagem e simulação de NCS e estratégias de projeto e controle de NCS. O objetivo é analisar e simular o NCS através de ferramentas desenvolvidas e aplicar as estratégias de controle e projeto existentes de forma a se garantir um bom desempenho e estabilidade para o NCS proposto.

Percebe-se, portanto que em torno do tema de NCS se desenvolve uma nova área de pesquisa multidisciplinar, relacionando conhecimentos de sistemas de controle, sistemas de tempo real e redes de comunicação (ZAMPIERI, 2008). Esse novo tema de pesquisa já apresenta grande difusão no meio acadêmico internacional, com diversos números especiais de revistas e congressos relacionados ao assunto. No entanto, no Brasil ainda são poucos os grupos de pesquisa que trabalham e possuem publicações relacionadas com NCS. E dentre os trabalhos desses grupos, quase não se encontram relatos de implementações práticas e reais de NCS, somente de revisões e análises a partir de trabalhos de simulação.

Baseando-se nas necessidades e oportunidades citadas, esta tese de doutorado estuda o desenvolvimento de NCS para aplicação em redes CAN através da análise por simulação e da proposta de uma estratégia de controle, de forma a que a influência de fatores degenerativos como os atrasos de comunicação e o período de amostragem de sinais, que afetam o desempenho e estabilidade desses sistemas, seja minimizada.

1.2. OBJETIVOS

Este trabalho apresenta o desenvolvimento de NCS baseado no uso da simulação e na proposta de uma estratégia de controle visando contornar os efeitos degenerativos causados por fatores como o período de amostragem e objetivando garantir requisitos de desempenho e estabilidade para o NCS.

Para alcançar tal objetivo, este trabalho relacionará conhecimentos das áreas de sistemas de controle, sistemas de tempo real e redes de comunicação, com ênfase no protocolo CAN, e apresentará as seguintes etapas:

- ✓ Avaliar qualitativamente ferramentas que fornecem suporte a simulação de NCS, de forma a verificar suas capacidades, vantagens e desvantagens na implementação de NCS e auxiliar nas fases de projeto e desenvolvimento desse tipo de sistema de controle distribuído;
- ✓ Analisar o impacto da rede de comunicação e de fatores de implementação (atrasos de comunicação, amostragem dos sinais, perda de informações na rede) no desempenho de controle e na estabilidade de NCS, evidenciando o mais importante ou que mais influencia o projeto de um NCS;
- ✓ Desenvolver uma estratégia de controle para NCS que garanta requisitos de controle individuais para obtenção de um melhor desempenho global do sistema estudado e que minimize a influência do período de amostragem no NCS;

1.3. CONTRIBUIÇÕES

Podem ser citadas as seguintes contribuições desta Tese de Doutorado:

- ✓ Extensa revisão bibliográfica cobrindo os principais focos de pesquisa em NCS: análise da influência da rede de comunicação no controle em malha fechada, desenvolvimento e utilização de ferramentas de simulação para NCS e estratégias de projeto e controle baseadas em diferentes abordagens para aplicação em NCS, que visam contornar os principais desafios relacionados ao desenvolvimento de NCS;
- ✓ Detalhado estudo comparativo entre ferramentas de simulação e suporte ao projeto de NCS disponíveis na literatura, o qual fornece um material de apoio para projetistas e pesquisadores da área de NCS;
- ✓ Verificação, através da aplicação de ferramentas de simulação de NCS, da grande importância do período de amostragem das mensagens para o projeto de NCS, o qual representa o principal parâmetro de configuração que afeta o desempenho e a estabilidade de NCS com redes CAN;
- ✓ Proposta de uma técnica de estimação de modelos matemáticos para NCS que faz uso da arquitetura totalmente distribuída do NCS, na qual todos os dados necessários estão disponíveis na rede de comunicação, e é realizada sem a necessidade de conexão de nenhum *hardware* adicional e sem a necessidade de acesso físico ao processo (ou planta) controlado do NCS;
- ✓ Desenvolvimento de estratégias de controle para NCS baseadas em técnicas de controle PID e controle adaptativo, as quais apresentaram bom desempenho para controle dos NCS desenvolvidos, além de ser uma solução confiável e robusta

para contornar um dos principais problemas em NCS que é a influência do período de amostragem das mensagens no NCS.

- ✓ Desenvolvimento de arquiteturas de *hardware* e *software* para a plataforma de pesquisa em NCS que apresentam grandes vantagens como, tornar o desenvolvimento dos NCS mais simples e de baixo custo, facilitar o desenvolvimento e a reutilização de códigos, e que podem ser aplicados no desenvolvimento real de NCS;
- ✓ Projeto e implementação de uma plataforma de pesquisa sobre NCS com redes CAN, flexível e de baixo custo, que possibilita a validação de desenvolvimentos teóricos e resultados simulados e que permite a obtenção de novos conhecimentos e conceitos técnicos para a área de NCS.

1.4. ESTRUTURA E CONTEÚDO

Esta tese de doutorado está organizada em seis capítulos e uma lista de referências bibliográficas. Neste **Capítulo 1 – Introdução** estão apresentadas as justificativas para a realização deste trabalho bem como seus objetivos, principais contribuições e conteúdo e organização. No **Capítulo 2 - Revisão Bibliográfica** são apresentados os conceitos e fundamentos sobre sistemas de controle via redes (NCS) que suportam o desenvolvimento deste trabalho. No **Capítulo 3 – Ferramentas de Análise e Simulação de NCS** é apresentada uma detalhada pesquisa e avaliação de ferramentas de suporte ao desenvolvimento de NCS disponíveis na literatura, a qual auxilia na escolha da ferramenta mais adequada para utilização neste trabalho e serve como referência para outros trabalhos e desenvolvedores. No **Capítulo 4 – Plataforma de Pesquisa sobre NCS**, são apresentadas todas as etapas de

desenvolvimento da plataforma de pesquisa de NCS com rede CAN construída no trabalho, descrevendo-se a modelagem das malhas de controle e a estrutura física da plataforma com ênfase nas características e vantagens das arquiteturas de *hardware* e *software* implementadas. No **Capítulo 5 – Estratégias de Projeto e Controle de NCS** são apresentadas uma estrutura padrão para a implementação de NCS e desenvolvidas estratégias de controle baseados em PID e numa abordagem de controle adaptativo para contornar um dos principais problemas no projeto de NCS que é o efeito do período de amostragem das mensagens no desempenho e estabilidade do NCS. No **Capítulo 6 - Aplicações da Plataforma de Pesquisa no Desenvolvimento de NCS com Redes CAN** são apresentados e discutidos todos os resultados do trabalho. Divididos em subseções, os resultados vão desde simulações que fornecem informações sobre o desempenho dos controladores desenvolvidos e sobre a influência de parâmetros no desempenho de NCS, passando por uma técnica de estimação de NCS até a avaliação da versatilidade e robustez de aplicação do controlador adaptativo desenvolvido para NCS com redes CAN. No **Capítulo 7 – Conclusões** são apresentadas conclusões deste trabalho e as perspectivas sobre os trabalhos futuros que serão desenvolvidos. A Lista de **Referências** é apresentada em ordem alfabética e contém todos os documentos referenciados no decorrer deste trabalho.

2. REVISÃO BIBLIOGRÁFICA

2.1. CONSIDERAÇÕES INICIAIS

Neste capítulo é apresentado o embasamento bibliográfico deste trabalho. Diversos trabalhos que apresentam o histórico de desenvolvimento e o estado da arte atual em relação à aplicação de sistemas de controle via rede (NCS) são apresentados. Os principais conceitos e desafios inerentes a aplicação de NCS são descritos e discutidos. Também são abordados trabalhos relacionados com o desenvolvimento de ferramentas de simulação e metodologias de controle para aplicação específica em sistemas de controle via redes industriais.

2.2. SISTEMAS DE CONTROLE VIA REDES - NCS

Uma grande tendência nos sistemas de manufatura industriais é a integração de comunicação e controle de vários níveis de processos de manufatura. Uma solução que vem sendo adotada para isso é a utilização de tecnologias de sistemas de controle distribuído via redes de comunicação (YANG, 2006b), também chamados de sistemas de controle via redes (NCS). De acordo com Tipsuwan & Chow (2003), os NCS podem ser classificados em dois grupos: estrutura direta e estrutura hierárquica. Na estrutura direta, mostrada na Figura 3, o controlador e a planta ficam fisicamente alocados em locais separados e são conectados diretamente por uma rede de comunicação formando uma malha de controle remota. O sinal de controle é enviado para o atuador através de uma mensagem encaminhada via rede,

enquanto o sensor amostra a saída da planta e retorna a informação para o controlador também transmitindo a informação através da rede.

Figura 3. NCS com Estrutura Direta (Adaptado de Tipsuwan & Chow, 2003)

Já a estrutura hierárquica, mostrada na Figura 4, consiste de um controlador principal e um sistema de controle em malha fechada remoto. Periodicamente, o controlador principal calcula e envia uma mensagem com o sinal de referência através da rede para o sistema remoto. O sistema remoto processa o sinal de referência e executa localmente o controle em malha fechada.

Figura 4. NCS com Estrutura Hierárquica (Adaptado de Tipsuwan & Chow, 2003)

Segundo Perez (2006), os critérios para escolha de uma das duas estruturas para o desenvolvimento de NCS dependem das restrições de projeto e das preferências do projetista. Normalmente, NCS que utilizam a Internet utilizam a estrutura hierárquica, pelo fato do tempo de transmissão das mensagens variarem consideravelmente e sua estimativa ser consideravelmente difícil. Já NCS implementados através de redes *fieldbus* ou locais, utilizam a estrutura direta, por possuírem tempos de transmissão menores e tempos de resposta mais

previsíveis. Santos (2004) define os principais domínios de aplicação de NCS como automotivo, sistemas de manufatura e controle de processos, robótica e ensino a distância. De acordo com esses domínios, focos de pesquisa diferentes são selecionados como no caso do automotivo (GAID et al., 2006), onde informações sobre sistemas de tempo real (*deadline* de mensagens) são priorizadas e no caso de sistemas de manufatura (MOYNE & TILBURY, 2007), onde informações sobre técnicas para controle e comunicação dos diversos processos de um sistema com um todo são enfatizadas.

Geralmente nos NCS, as tarefas de processamento, troca de informações e controle são distribuídas entre várias unidades de controle eletrônico (ECU – *Electronic Control Unit*), dedicadas ao controle de uma parte do processo. Para que a troca de informações entre essas ECU ocorra de maneira eficiente, um grande número de protocolos de comunicação tem sido desenvolvido e aplicado em NCS (MOYNE & TILBURY, 2007), como o Profibus, CAN e Foundation Fieldbus. Dentre esses protocolos, o CAN (BOSCH, 2006) originalmente desenvolvido para aplicações na área automotiva, se encontra atualmente muito difundido e tem sido utilizado em muitas outras aplicações industriais de controle (OTHMAN et al., 2006).

2.3. ATRASOS DE COMUNICAÇÃO EM NCS

A troca de informações pela rede de comunicação, entre as ECUs dos dispositivos componentes de um NCS acaba induzindo atrasos de comunicação na transmissão das mensagens (HUO, FANG & MA, 2004). De acordo com Tipsuwan & Chow (2003), esse atraso de comunicação entre a transmissão de mensagens em um NCS apresenta a seguinte composição:

- ✓ Tempo de espera: atraso relacionado com a espera da ECU pela disponibilidade da rede para transmissão de mensagens;
- ✓ Tempo de processamento: atraso relacionado com o tempo de processamento dos pacotes de dados ou mensagens na rede pelas ECUs;
- ✓ Propagação: atraso relacionado ao tempo de transmissão das mensagens na rede.

Figura 5 mostra um exemplo dos atrasos encontrados em uma transmissão de uma mensagem por uma rede de comunicação industrial, desde sua inicialização ou o começo da transmissão por uma ECU de origem até o término da recepção da mensagem por parte da ECU de destino.

Figura 5. Composição do Atraso de Comunicação em NCS (Adaptado de Lian, Moyne & Tilbury, 2002)

O atraso de comunicação total T_{delay} , pode ser dividido em três partes: atrasos de comunicação na ECU de origem, $T_{src} = T_{pre} + T_{wait}$, na rede de comunicação industrial, T_{bus} , e na ECU de destino das mensagens, $T_{dest} = T_{post}$, como pode ser visto no diagrama detalhado da Figura 6.

Figura 6. Diagrama Temporal da Transmissão de uma Mensagem CAN (Adaptado de Lian, Moyne & Tilbury, 2001)

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

Lian, Moyne & Tilbury (2002) afirmam que os atrasos de comunicação encontrados em NCS dependem do protocolo de comunicação escolhido e de parâmetros de configuração da rede de comunicação. Entre esses parâmetros podem ser citados a velocidade de transmissão

(largura de banda), o tamanho das mensagens de dados, os períodos de amostragem dos dispositivos e a porcentagem de mensagens perdidas. Seuret et al. (2006) afirma que a arquitetura do NCS e o sistema operacional utilizado também afetam a composição dos atrasos de comunicação. Outro critério muito importante em relação NCS diz respeito ao cumprimento do requisito temporal de cada mensagem (*deadline*), ou seja, as mensagens têm que ser transmitidas corretamente em um tempo limitado e menor que seu período de amostragem. Caso as mensagens não sejam transmitidas ou o sistema apresente um alto valor para seu tempo de transmissão, sobreposição e perdas de mensagens podem ocorrer deteriorando o desempenho do NCS (AL-HAMMOURI, BRANICKY & LIBERATORE, 2008). Esses fatores acabam por definir as características desses atrasos como sendo constantes ou variantes no tempo (SO, 2003) e como esses atrasos serão modelados (NILSSON, 1998) e considerados em NCS.

A modelagem dos atrasos de comunicação em NCS é em muitos casos um pré-requisito para o projeto de controle. Existem diversas abordagens para modelagem de atrasos de comunicação dependendo do tipo de rede de comunicação e protocolo utilizado (LIAN, 2001; ZHANG, 2001; SO, 2003; RODRIGUEZ, 2007; RAJU, 2009). Em geral, os modelos de atrasos de comunicação considerados em NCS seguem dois padrões. Atrasos constantes ou limitados (*bounded*) e atrasos variantes (NILSSON, 1998). O modelo de atraso constante ou limitado é usado em NCS onde a dinâmica do processo é mais lenta que a da rede e, portanto os atrasos de comunicação na rede são muito menores que as constantes de tempo do NCS. Assim como os atrasos de comunicação são menores que o período de amostragem do NCS, a variação desse atraso tem pouca influência e pode ser desprezada. O modelo do atraso variante é mais difundido e utilizado em NCS, pois consegue incorporar uma maior quantidade de características presentes em NCS como perda de mensagens, largura de banda

limitada, não sincronismo entre dispositivos na rede, disputa por acesso a rede e colisão, *jitter* e alta taxa de utilização da rede.

2.4. FATORES QUE AFETAM O DESEMPENHO E A ESTABILIDADE DE NCS

De acordo com a literatura (HOKAYEM & ABDALLAH, 2004; JIANYONG, SHIMIN & HAIQING, 2004; PEREZ, 2006; HESPANHA, NAGHSHTABRIZI & XU, 2007; MOYNE & TILBURY, 2007), a interconexão entre os componentes de um NCS (sensores, atuadores e controladores) através de uma rede de comunicação evidencia diversos fatores inerentes à implementação de NCS que acabam deteriorando seu desempenho e podendo torná-lo instável. Entre esses fatores ou características intrínsecas da implementação de NCS estão:

- ✓ Atrasos de comunicação do NCS;
- ✓ Amostragem dos sinais;
- ✓ Perda de mensagens transmitidas;
- ✓ Parâmetros de configuração da rede de comunicação (quantidade, tamanho, prioridade de mensagens)
- ✓ Presença de *Jitter*;
- ✓ Escalonamento de tarefas e mensagens;
- ✓ Limitação de largura de banda da rede de comunicação (velocidade de transmissão de dados)

O atraso de comunicação em NCS, ainda que constante ou variante no tempo, representa um dos mais importantes fatores que podem afetar o desempenho e a estabilidade de um NCS. Por causa disso, a influência deste fator no NCS tem sido muito pesquisada e

diversas metodologias para compensar esses efeitos têm sido desenvolvidas. O trabalho de Tipsuwan & Chow (2003) apresenta dois gráficos, mostrados na Figura 7, que relacionam a influência de um atraso de comunicação (t) no desempenho e estabilidade de um NCS, cujos dados estão descrito na Tabela 1.

Tabela 1. Dados do NCS Utilizado no trabalho de Tipsuwan & Chow (2003)

Função de Transferência - Planta	Função de Transferência - Controlador PI	Parâmetros do Controlador
$G(s) = \frac{2029,83}{(s + 26,3)(s + 2,3)}$	$C(s) = \frac{Kp.(s + Ki/Kp)}{s}$	$Kp = 0,17$ e $Ki = 0,38$

Figura 7. Efeito do Atraso de Comunicação (t) no Desempenho e Estabilidade de um NCS (Tipsuwan & Chow, 2003)

De acordo com os gráficos, percebe-se que o aumento do atraso de comunicação (t) no NCS provoca uma deterioração do desempenho do sistema (maior oscilação e aumento do tempo acomodação e *overshoot*) e um atraso inicial na resposta do sistema, visto no primeiro gráfico da Figura 7 referente à resposta a um degrau de entrada do NCS. Tal aumento acarreta também uma diminuição da margem de estabilidade do sistema (deslocamento do gráfico do lugar das raízes para a direita no eixo real), visto no segundo gráfico da Figura 7 referente ao lugar das raízes do NCS.

No entanto, a presença de atrasos de comunicação pode ter características favoráveis em casos específicos de sistemas de controle (RICHARD, 2003; NICULESCU, GU & ABDALLAH, 2003). Zhang, Branicky & Phillips (2001) discutem o benefício do atraso de comunicação sobre a estabilidade de configurações de NCS. Khan, Tilbury & Moyne (2008) mostram que se o erro de rastreamento (*tracking error*) for utilizado como métrica de desempenho para o NCS, então a qualidade de controle (desempenho de controle) pode ser melhorada para NCS com atrasos de comunicação constantes e previsíveis.

De acordo com Lian et al. (2006), um dos parâmetros de configuração de redes de comunicação que mais exerce influência sobre o desempenho de um NCS é o período de amostragem das mensagens que trafegam pela rede (ou amostragem dos sinais do sensor e atuador). Neste trabalho é apresentado um diagrama que auxilia na visualização deste problema e na tarefa de seleccionar tais períodos de amostragem. A Figura 8 apresenta um diagrama que relaciona o período de amostragem dos sinais com o desempenho de um sistema de controle implementado sob diferentes formas (controle contínuo, controle digital e controle via rede - NCS), demonstrando a grande influência da amostragem no comportamento de um NCS.

Figura 8. Relação entre Período de Amostragem e Desempenho em NCS (Adaptado de Lian et al., 2006)

Os critérios de classificação do desempenho de controle (satisfatório, insatisfatório e instabilidade) podem ser estabelecidos de forma a refletirem especificações do NCS projetado, tais como requisitos de estabilidade como margem de fase e de ganho e requisitos de resposta temporal como tempo de acomodação, valor máximo de pico e erro em regime permanente (LIAN et al., 2006).

De acordo com a Figura 8, em sistemas de controle contínuo o período de amostragem não influi no desempenho do sistema. Em sistemas de controle digital, o desempenho do sistema se aproxima do desempenho de sistemas de controle contínuo conforme o período de amostragem se torna mais rápido. Nos sistemas de controle via rede - NCS, os períodos de amostragem mais lentos (ponto A) podem representar menores desafios em relação ao cumprimento de requisitos de desempenho do sistema de controle, porém a rede de comunicação apresentará alto nível de ociosidade (capacidade de processamento e troca de informações que não é usada). Para períodos de amostragem mais rápidos, no entanto, a carga de dados trafegando pela rede se torna maior e sua ociosidade diminui, porém a possibilidade de ocorrer mais disputas pelo acesso a rede e um aumento nos atrasos de comunicação pode ser esperado (ponto B). Caso esse período de amostragem seja muito rápido, pode ocorrer a saturação da rede de comunicação. Nessa situação (ponto C), onde a rede fica sobrecarregada de mensagens, novas mensagens são enviadas para a rede sobrepondo as anteriores que ainda não tinham sido transmitidas e erros de transmissão tornam-se constantes, podendo tornar o sistema de controle instável.

Uma grande diferença entre NCS e sistemas de controle digital é a possibilidade de perda de informações durante a transmissão das mensagens pela rede. Geralmente, a perda de transmissão de pacotes ou até mensagens em NCS resulta de erros na transmissão das mensagens nos canais de comunicação da rede, erros devido à sobreposição e estouro de

mensagens nos buffers de transmissão e recepção e erros relacionados a falhas de dispositivos e acesso (colisão) à rede de comunicação (HOKAYEM & ABDALLAH, 2004).

Valdivia & Souza (2007) mostram, através de simulações, a degradação do desempenho de controle de um NCS diante do aumento da taxa de perda de pacotes ou mensagens na rede de comunicação. O gráfico da Figura 9 apresenta a comparação entre a resposta ao degrau de um NCS com taxa de perda de pacotes de 0% e 10% (em relação ao total transmitido).

Figura 9. Influência da Perda de Pacotes ou Mensagens no Desempenho de um NCS (Valdivia & Souza, 2007)

Nos NCS, o desempenho de controle do sistema não depende somente do projeto de controle, mas também do escalonamento de tarefas nos dispositivos e mensagens na rede de comunicação. A teoria de escalonamento foca no problema de, dado um conjunto de tarefas a serem processadas ou de mensagens a serem transmitidas, encontrar uma ordem de execução que garanta com que todas as tarefas e mensagens cumpram seus requisitos temporais (*deadlines*). De acordo com Walsh & Ye (2001), algoritmos de escalonamento para NCS são divididos em duas categorias: estáticos e dinâmicos. O escalonamento estático trata de uma

abordagem *offline* (sistema não operando) onde a ordem de execução é definida previamente para o NCS. É uma técnica fácil de ser implementada e analisada, no entanto necessita ser refeita a cada alteração feita nos dispositivos do NCS. No escalonamento dinâmico, a decisão de qual tarefa ou mensagem será executada é feita *online* (com o sistema operando). Uma revisão de algoritmos de escalonamento para aplicação em NCS é apresentada em Kim (2003).

2.5. SIMULAÇÃO E ANÁLISE DE NCS

Com o crescimento das pesquisas e desenvolvimentos na área de NCS, surgiu-se a necessidade de analisar a possibilidade de aplicação de diversos tipos de redes de comunicação em NCS. Os trabalhos de Lian, Moyne & Tilbury (2001) e Lian et al. (2006) apresentam uma avaliação de desempenho de protocolos de comunicação (DeviceNet, ControlNet e Ethernet) utilizados em NCS e um estudo da influência dos atrasos de comunicação no desempenho de NCS que utilizam estes protocolos. Em Santos, Vasques & Stemmer (2003, 2004), NCS baseados em rede CAN, Profibus e Token Passing são avaliados quanto à capacidade de garantir requisitos de controle sob diferentes configurações de parâmetros. Diaz, Pedro & Caurin (2008) estudam a utilização de *fieldbus* em aplicações de controle em tempo real e avaliam o impacto da presença dessas redes em NCS. Neste trabalho, através da análise da resposta temporal de um NCS sob diversos tipos de sinais de referência (degrau, quadrada, senoidal e triangular), é confirmada a viabilidade da utilização de redes CAN em aplicações de NCS na robótica.

Cervin et al. (2003) afirma que uma das principais dificuldades relacionadas ao projeto de NCS em sistemas de manufatura se encontra na análise da influência dos atrasos de comunicação e de outros fatores inerentes a NCS no desempenho e estabilidade do sistema.

Jun & Zhan-lin (2005) estudam os componentes dos atrasos de comunicação em NCS com redes Profibus e analisam como a presença dessa rede afeta o desempenho de controle em malha fechada. A relação entre o período de amostragem de mensagens e a velocidade de transmissão de dados (largura de banda) em um NCS é discutida em Lian et al. (2004) focando no impacto da escolha desses parâmetros no desempenho do sistema. Pang, Yang & Nishitani (2006) analisam as características temporais de NCS com redes Foundation Fieldbus e concluem que a influência no desempenho do sistema, do tempo de execução de blocos de funções (FB- *function blocks*) é dominante sobre a dos atrasos de comunicação. Soglo & Xianhui (2006) analisam o desempenho de NCS sob diferentes configurações como quantidade de plantas no sistema e taxa de perda de mensagens transmitidas. A avaliação da variabilidade de desempenho de um NCS de acordo com a alteração de parâmetros de configuração do protocolo CAN, como quantidade, tamanho e prioridade de mensagens, bem como a verificação da porcentagem de *deadline* de mensagens perdidas é discutida em Godoy et al. (2007). Simulações realizadas por Xia, Wang & Sun (2004) demonstram que restrições de CPU (capacidade de processamento) e da rede de comunicação (capacidade de transmissão de mensagens), devido a altas taxas de utilização desses recursos, prejudicam a qualidade de controle em NCS, pois induzem grandes atrasos de comunicação. Lluesma et al. (2006) concluem, através de simulações, que a escolha do algoritmo de escalonamento aplicado ao NCS afeta diretamente na presença de *jitter* no sistema, devendo, portanto ser levado em consideração no projeto de controle de NCS.

A influência do tipo de rede de comunicação e de suas características de alto nível no desempenho de NCS é apresentada em Lustosa & Souza (2008), onde são avaliados NCS com redes CAN (controle de acesso ao meio físico baseado em eventos – *event-driven*) e com redes TTP (controle de acesso ao meio físico baseado em tempo – *time-driven*). Nessa mesma linha de pesquisa, Cervin & Henningsson (2008) utilizam métodos numéricos para avaliar,

através de uma função custo, o desempenho de controle de NCS com vários tipos de protocolos (MAC – *Medium Access Control*) de acesso ao meio físico (CSMA, TDMA e FDMA). Os resultados desse trabalho, também vistos no trabalho de Lustosa & Souza (2008), demonstraram que NCS com controle baseado em eventos sob protocolo CSMA apresentam melhor desempenho entre os casos analisados.

De acordo com Seuret et al. (2006), a arquitetura do NCS afeta diretamente a composição dos atrasos de comunicação num NCS. Goodwin, Quevedo & Silva (2008) apresentam uma análise do desempenho de NCS sob diversas arquiteturas de configuração. Seguindo essa linha de trabalho, os autores propõem em Silva, Goodwin & Quevedo (2008) uma arquitetura para NCS com múltiplas entradas e saídas (MIMO – *multiple input multiple output*) e em Quevedo & Goodwin (2005) uma arquitetura para NCS onde a rede de comunicação pode ser utilizada de duas formas diferentes, de acordo com o requisito de operação do NCS. Num modo de operação, a arquitetura garante a transmissão de mensagens sem erros, mas com grandes atrasos de comunicação e reduzido período de amostragem e no outro modo, a transmissão de mensagens ocorre rapidamente (e com mínimos atrasos), porém a possibilidade de erros na transmissão de mensagens é maior.

O estudo da estabilidade de um NCS é outro problema básico dessa área de pesquisa, a partir do momento que a presença da rede de comunicação no controle em malha fechada diminui a margem de estabilidade de NCS. Regiões de estabilidade e técnicas para garantir a estabilidade de NCS foram propostas inicialmente em Zhang, Branicky & Philips (2001). Existem diversos estudos sobre critérios de estabilidade para NCS de forma a garantir que o sistema permaneça estável em condições impostas pela natureza do NCS. No entanto, não existe uma técnica de análise de estabilidade genérica que possa ser aplicado para qualquer tipo de NCS (ZHANG, BRANICKY & PHILLIPS, 2001). A maioria dos estudos e técnicas

propostas está condicionada a certas configurações de NCS, protocolos de comunicação utilizados, suposições e técnicas de controle utilizadas.

Técnicas de análise de estabilidade para sistemas discretos com atrasos de transporte podem ser aplicadas na avaliação de NCS com atrasos constantes. NCS com atrasos variantes no tempo representa uma área mais desafiadora, pois necessita de algoritmos mais avançados. Diversas técnicas têm sido utilizadas na análise da estabilidade de NCS. Os componentes do NCS (controlador e planta) podem ser modelados no tempo contínuo, discreto ou em ambos os casos (KAO & LINCOLN, 2004). Um grande número de trabalhos tem procurado definir um valor mínimo (ou mais lento) para o período de amostragem de NCS de forma a garantir sua estabilidade (HESPANHA, NAGHSHTABRIZI & XU, 2007). Estes trabalhos implicitamente buscam minimizar a taxa de transmissão de dados necessária para estabilizar o sistema em malha fechada. Os atrasos de comunicação do NCS podem ser conhecidos e constantes e o critério de estabilidade pode utilizar esses valores (WU et al., 2004). Para os casos de atrasos desconhecidos e variantes, os critérios de estabilidade geralmente são fornecidos no domínio do tempo (FRIDMAN & SHAKED, 2003) ou no domínio da frequência (KAO & LINCOLN, 2004).

Recentemente, trabalhos têm utilizado técnicas mais avançadas na análise da estabilidade de NCS. Uma técnica de controle preditivo e robusto é utilizada em Chen, Lin & Hwang (2007) para compensar os efeitos dos atrasos de comunicação e garantir a estabilidade de um NCS. Cloosterman et al. (2006) sintetiza um controlador através de Desigualdades Matriciais Lineares (LMIs - *Linear Matrix Inequalities*) que estabiliza um NCS na presença de comunicação variantes no tempo. Uma comparação entre a utilização de funções de *Lyapunov* dependentes de parâmetros e funções de *Lyapunov-Krasovskii* para análise de estabilidade de NCS é demonstrada em Hetel et al. (2009).

De modo a facilitar essas tarefas de análise e simulação de desempenho e estabilidade de NCS, o uso de ferramentas computacionais de simulação se torna indispensável. No entanto, ainda que exista um grande número de ferramentas que fornecem suporte ao desenvolvimento e simulação de sistemas de controle, redes de comunicação e sistemas de tempo real, poucas ferramentas suportam o desenvolvimento integrado (*codesign*) ou correlacionado entre essas áreas, que é o foco pesquisado em NCS. Assim, devido a essa carência e crescente demanda, diversos trabalhos têm sido realizados com o intuito de desenvolver tais ferramentas para aplicação em NCS (TORNGREN et al., 2006; AL-HAMMOURI, BRANICKY & LIBERATORE, 2008).

A maioria dessas ferramentas tem sido desenvolvida no ambiente *Matlab/Simulink*, norteado principalmente pela grande aceitação e disseminação desse ambiente nas áreas acadêmica e industrial. Godoy, Porto & Inamasu (2008a) apresentam uma revisão sobre algumas dessas ferramentas e uma comparação de funcionalidades para aplicação em NCS. Essas ferramentas têm sido desenvolvidas por instituições de pesquisa e cada uma focada em uma área de pesquisa (sistemas de controle, sistemas de tempo real ou redes de comunicação) de NCS. As ferramentas *TrueTime* e *Jitterbug* (CERVIN et al., 2003) foram desenvolvidas especificamente para projeto de controle. Na ferramenta *TrueTime*, o projetista consegue modelar todo o NCS usando uma biblioteca de blocos do *Simulink*. Esta biblioteca permite a definição de parâmetros de configuração para a rede de comunicação utilizada, o desenvolvimento de técnicas de controle e a seleção do modelo da planta analisada. A ferramenta *Jitterbug* apresenta uma grande abstração na definição dos atrasos de comunicação, porém permite o cálculo de uma função custo relacionada à qualidade de controle do NCS sob diferentes condições temporais do NCS. As ferramentas *AIDA* (REDELL, EL-KHOURY & TORNGREN, 2004) e *TORSCHÉ* (SUCHA et al., 2006) foram desenvolvidas para a análise de características de tempo real de NCS. A ferramenta *AIDA* é

usada para análise temporal de NCS enquanto que a aplicação de algoritmos de escalonamento para otimização da resposta do NCS representa a principal finalidade da ferramenta *TORSCHE*. A ferramenta *PiccSIM* (NETHI et al., 2007a) fornece a funcionalidade de utilização via Internet. O projetista do NCS pode carregar seu algoritmo de controle, definir alguns parâmetros de configuração e usar alguns modelos de planta para a simulação de NCS. Hasan et al. (2008) integram o ambiente *Matlab* ao *software OPNET - Optimised Network Engineering Tool* (CHANG, 1999) para simulação de NCS com redes sem fio (*wireless*).

Ainda que a maioria das ferramentas desenvolvidas para NCS utilize o ambiente *Matlab/Simulink*, não é difícil encontrar na literatura outras ferramentas para NCS desenvolvidas em outras plataformas. Pinnotti Jr & Brandão (2005) apresentam uma ferramenta baseada no ambiente *LabVIEW* para simulação de NCS. Liu & Frey (2008) desenvolvem uma biblioteca no ambiente *Modelica* para permitir a análise de viabilidade de aplicação de NCS. Al-Hammouri, Branicky & Liberatore (2008) apresentam três ferramentas para projeto de NCS, sendo duas delas, *Agent/Plant* e *NCSPlant*, extensões do *software* de simulação de redes *Network Simulator NS-2* (NS-2, 2007), e a outra sendo uma integração entre o ambiente *Modelica* e o *software* NS-2.

2.6. METODOLOGIAS DE PROJETO E CONTROLE PARA NCS

O desenvolvimento de NCS apresenta novos desafios para as tradicionais técnicas de projeto de sistemas de controle. Muitos métodos conhecidos e consolidados têm sido desenvolvidos e aperfeiçoados para a análise e projeto de sistemas de controle sujeitos a efeitos de discretização de sinais. Da mesma forma, muita pesquisa tem sido realizada para a

análise de malhas de controle com atraso de transporte e os efeitos da presença desse atraso constante no desempenho e na estabilidade de sistemas de controle em malha fechada (RICHARD, 2003). No entanto, no desenvolvimento de NCS, o projeto de controle deve levar em consideração, simultaneamente, os efeitos da amostragem dos sinais e dos atrasos de comunicação, que na maioria das vezes é variante no tempo. A planta ou processo a ser controlado e as especificações de desempenho requeridas para o NCS é que vão definir se técnicas de controle conhecidas poderão ser aplicadas para compensar os efeitos da presença da rede de comunicação no sistema, ou se será necessária a aplicação de novas estratégias para o projeto e controle do NCS (BAILLIEUL & ANTSAKLIS, 2007).

Essas novas estratégias de projeto e controle têm sido desenvolvidas baseadas em diversos tipos de rede e protocolos de comunicação em conjunto com diferentes abordagens para tratamento dos efeitos degenerativos de fatores de implementação de NCS como atrasos de comunicação, perda de pacotes e amostragem dos sinais. Uma metodologia de controle é requerida para minimizar os efeitos desses fatores em um NCS, melhorando seu desempenho e garantindo sua estabilidade. Uma revisão de algumas metodologias é apresentada em Tipsuwan & Chow (2003), Li & Fang (2005), Hespanha, Naghshtabrizi & Xu (2007), Nair et al. (2007) e Vatanski et al. (2009).

A metodologia de programação de ganho (TIPSUWAN, 2003) utiliza uma técnica de agendamento de ganhos do controlador (*gain scheduling*) de um NCS para compensar os efeitos dos atrasos de comunicação. Nesta estratégia, os ganhos são ajustados de acordo com o erro de saída do sistema causado pelo atraso da rede. Godoy et al. (2010e) aplica a metodologia de programação de tempo de amostragem para um NCS via rede CAN. Nesse trabalho, com a aplicação da estratégia de controle, uma otimização do NCS é conseguida através da correta seleção dos períodos de amostragem dos dispositivos de um NCS, minimizando os efeitos dos atrasos de comunicação no desempenho do sistema. Perez et al.

(2007) propõe uma estratégia baseada na alocação dinâmica de largura de banda da rede entre as malhas de controle de um NCS. Neste trabalho são calculados limites inferiores das necessidades de largura de banda para as malha de controle operando em regime permanente, alocando assim recursos excedentes para as malhas em regime transitório. Estratégias de *codesign* também têm sido desenvolvidas para aplicação em NCS. Uma estratégia de controle de NCS baseada no tempo de transmissão de mensagens na rede é apresentada em Santos et al. (2006). Os conceitos de margem de *jitter* (*jitter margin*) e margem de atraso (*delay margin*), que fornecem respectivamente os valores máximos de *jitter* e atraso de comunicação que um NCS pode suportar antes de perder sua estabilidade, são bastante utilizados no desenvolvimento de estratégias de controle para NCS, como pode ser visto em Santos, Vasques & Stemmer (2006) e Perez, Moreno & Montez, (2006). Pohjola (2008) apresenta o desenvolvimento de um controlador adaptativo para NCS que combina a informação de atrasos de comunicação estimados com a teoria de margem de *jitter*.

Basicamente, as metodologias de controle de NCS buscam compensar os efeitos dos atrasos de comunicação e perda de pacotes no NCS através do cálculo ou estimação desses parâmetros ou informações perdidas (NAIR et al., 2007). Em alguns NCS, os dados ou mensagens transmitidas pela rede possuem informações de tempo (*timestamp*), possibilitando que o dispositivo que está recebendo essas mensagens possa calcular o valor ou duração dos atrasos de comunicação na rede e assim utilizar esses dados na formulação de em ações corretivas. Quando essas informações de tempo não estão disponíveis, uma abordagem para o cálculo dos atrasos é a utilização de modelos dos atrasos de comunicação. No entanto, esses modelos são difíceis de serem obtidos e para contornar esse problema, é comum o uso de técnicas de estimação desse parâmetro. Vatanski et al. (2009) descreve detalhadamente as estratégias de cálculo e estimativa de atrasos de comunicação para formulação de metodologias de controle para NCS. Neste trabalho são desenvolvidas duas estratégias, sendo

uma fundamentada em controle preditivo baseado no preditor de Smith, para casos onde medidas do atraso de comunicação na rede são acessíveis, e outra fundamentada em controle robusto, para casos onde somente uma estimativa desses atrasos pode ser realizada.

A combinação da teoria de controle robusto com a de redes de comunicação tem obtido bastante sucesso para formulação de metodologias de controle para NCS. Nesse tipo de estratégia, os atrasos de comunicação do NCS são calculados ou estimados considerando-se as características da rede de comunicação utilizada. Estes atrasos de comunicação são então traduzidos em incertezas utilizadas na síntese de controladores robustos Georges et al. (2006). O trabalho de Diouri et al. (2007) apresenta uma modificação para metodologias de controle robusto para NCS. Neste trabalho, algoritmos de escalonamento de mensagens são utilizados nos dispositivos do NCS com a finalidade de reduzir o efeito da rede de comunicação no controlador robusto projetado. Cloosterman et al. (2006) desenvolvem um controlador robusto onde as incertezas dos atrasos de comunicação são propostas através da teoria de LMIs.

O estado da arte sobre metodologias de controle que consideram o efeito da perda de pacotes ou mensagens em um NCS pode ser visto em Hespanha, Naghshtabrizi & Xu (2007). Modelos de NCS com incorporação de perdas de pacotes têm sido propostos baseando-se em técnicas estocásticas (SEILER & SENGUPTA, 2001) e determinísticas (YUE, HAN & PENG, 2004) para compensar essa perda de informações. Para esses casos, uma das estratégias de controle mais comuns é a do controle baseado em modelo (MBC – *Model Based Control*) ou do controle preditivo baseado em modelo (MPC - *Model Predictive Control*). Alldredge (2007) apresenta uma revisão sobre algumas técnicas de controle preditivo baseado em modelo para aplicação em NCS. Estrada & Antsaklis (2009) descrevem o projeto de uma estratégia de controle para NCS baseada no modelo da planta de forma a reduzir o tráfego na rede de comunicação e assim diminuir a degradação do desempenho de NCS. Neste trabalho, através da estimação do comportamento da planta consegue-se diminuir

(deixar mais lenta) a amostragem dos sensores do NCS, acarretando uma menor utilização da rede. MPC utiliza um modelo da planta do NCS, inserido no controlador, para prever estados futuros da planta e calcular sinais de controle (atual + adicionais previstos), com o intuito de compensar os efeitos da perda de pacotes e atrasos de comunicação em NCS (ONAT, NASKALI & PARLAKAY, 2008). A cada período de amostragem do NCS, todos os sinais de controle calculados são enviados ao atuador, que em caso de perda de pacotes ou mensagens, aplica na planta os sinais de controle previstos e recebidos anteriormente. No entanto, metodologias de controle MPC podem ser muito sensíveis a erros de modelagem, principalmente quando são incluídos modelos dos atrasos de comunicação ao modelo da planta.

Outro trabalho interessante nessa área, Quevedo, Silva & Goodwin (2008) apresentam um método que utiliza NCS com redes de comunicação com grandes *frames* de dados para transmitir sequências de sinais de controle que cobrem múltiplos cenários de taxa de perda de pacotes e presença de atrasos. Assim, de acordo com a operação do NCS, o melhor sinal de controle é determinado e utilizado, obtendo um bom desempenho para o NCS.

A maioria das abordagens descritas anteriormente muitas vezes apresenta soluções complexas, com algoritmos que requerem grande capacidade de processamento e utilizam informações sobre a rede de comunicação e seus atrasos de comunicação, que são difíceis de serem obtidos. Dependendo da planta a ser controlada e dos requisitos de controle a serem alcançados, muitas vezes controladores mais simples podem compensar os efeitos degenerativos da rede e dos atrasos de comunicação (ERIKSSON, 2008). Entre essas estratégias estão as fundamentadas em técnicas de controle PID (POHJOLA, 2006) e controle regulador linear quadrático (LQR) (ZHANG & HUA-JING, 2006). Baseando-se nesses fatos, este tema torna-se muito relevante a partir do momento que controladores PID são extensivamente empregados na indústria e possuem grande flexibilidade de utilização.

Estratégias de controle PID para NCS têm sido relativamente pouco pesquisadas apesar de a definição de sua estrutura e sintonia de parâmetros serem decisivas para aplicações em NCS. Pohjola (2006) afirma que um controlador PID com uma sintonia de parâmetros apropriada pode tornar-se robusto em relação à presença de atrasos de comunicação e erros na transmissão de mensagens num NCS. Okano, Ohtani & Nagashima (2008) desenvolvem um controlador PID para NCS que compensa o efeito da perda de pacotes na transmissão da variável manipulada e mantém o desempenho de controle do sistema. Técnicas de sintonia de parâmetros de controladores PID para NCS são apresentadas e avaliadas comparativamente em Pohjola, Eriksson & Koivo (2006). Uma diferente estratégia de controle PID para NCS é demonstrada em Sala, Cuenca & Salt (2009), onde o controlador opera com uma técnica de *multi-rate*. Os autores mostram que com o uso desta técnica, na qual a amostragem do controlador é N vezes maior que a do sensor do NCS, consegue-se obter um tempo de acomodação mais rápido para o sistema. No trabalho em questão foi utilizada uma amostragem duas vezes maior, ou seja, uma técnica *dual-rate*.

De forma a se lidar com as incertezas dos atrasos de comunicação nos NCS, técnicas de inteligência artificial, como sistemas Fuzzy (TIAN, WANG & CHENG, 2007), redes neurais (ZHAO & XIA, 2006a) e algoritmos genéticos (SHIN & SUNWOO, 2007) também acabaram aplicadas no desenvolvimento de estratégias de controle para NCS. Os trabalhos de Lee et al. (2003, 2004) comparam e avaliam o desempenho de um NCS via rede Profibus com controladores PID, Fuzzy e sintonizados com algoritmo genético. Os resultados demonstraram que a utilização de sistemas Fuzzy pode ser uma escolha viável para NCS pelo fato de não requerer um modelo do processo controlado e apresentar certa robustez em relação aos atrasos de comunicação da rede, fatores que tornam muito interessante a utilização da lógica Fuzzy em NCS. A combinação entre controladores PID e lógica Fuzzy também tem sido largamente empregada em NCS. Zhang et al. (2008) utilizam dados do erro e da derivada

do erro de um NCS como sinais de entrada de um sintonizador Fuzzy para os ganhos de um controlador PID. Uma ideia inovadora para o desenvolvimento de controladores Fuzzy-PID para NCS é apresentada em Fadaei & Salahshoor (2008). Nesta estratégia, a resposta ao degrau do NCS é dividida em três zonas funcionais (em relação ao erro) e de acordo com essas zonas, três objetivos de controle diferentes são definidos em função do tempo de subida, do tempo de acomodação e do erro em regime permanente. A lógica Fuzzy é então usada para determinar a saída do controlador PID (tipo do controlador e valor dos ganhos) de acordo com essas três zonas criadas. Para a operação na primeira zona é definido um controlador PD, por causa do grande valor do sinal do erro, buscando tornar a resposta do NCS mais rápida e diminuindo o tempo de subida. Na segunda zona, o controlador necessita minimizar o tempo de acomodação e oscilações na resposta, assim um controlador PID é definido. E na terceira zona, de forma a eliminar o erro em regime permanente, um controlador PI é então definido.

Técnicas avançadas de simulação como simulação híbrida e *hardware-in-the-loop* (HIL) têm sido aplicadas no projeto de sistemas de controle em diversas áreas como sistemas mecatrônicos (ISERMANN, 2007) e sistemas automotivos (Park, Han & Lee, 2005). Gu et al. (2007) discute sobre o estado atual e necessidades relacionadas à pesquisa e desenvolvimento de técnicas de HIL aplicadas na área de controle e automação de sistemas de manufatura.

HIL refere-se à simulação de sistemas em tempo real onde alguns componentes ou modelos são substituídos por equipamentos reais (*hardware*). Diante dos benefícios da utilização dessa técnica, como redução de tempo de projeto e custo, trabalhos têm avaliado sua aplicação em sistemas com redes industriais como o CAN (WEI et al., 2006) e Profibus (VOGLAUER, GARCIA & JORGL, 2005). Lima et al. (2004) desenvolve um ambiente para simulação híbrida de sistemas de controle com redes Foundation Fieldbus.

De acordo com Godoy, Porto & Inamasu (2008b), o principal benefício que tem norteado a pesquisa de HIL em NCS é a possibilidade de simulação do sistema considerando-

se precisamente as influências da rede industrial no controle em malha fechada, através da utilização de uma rede de comunicação real. Mellios, Kansoulidou & Hassapis (2005) apresenta a aplicação da técnica de HIL para a avaliação de desempenho de estratégias de controle desenvolvidas para um NCS via rede PROFIBUS de uma planta de moagem de cimento. Berbra et al. (2009) utilizam a técnica de HIL para avaliar a operação de um NCS com rede CAN de um quadrotor (helicóptero com quatro rotores) sob condições de perda de pacotes e falha de sensores.

2.7. CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentada a revisão bibliográfica relacionada à sistemas de controle via redes (NCS), bem como seus principais conceitos e desafios de implementação. Pode-se perceber que a aplicação da tecnologia de NCS como solução de controle distribuído tem crescido recentemente e que esse assunto tem obtido grande atenção por parte das comunidades de sistemas de controle e redes de comunicação. De acordo com a revisão, as pesquisas e desenvolvimentos atuais em NCS têm focado principalmente duas linhas de trabalho. Uma na parte relacionada com a análise da influência de fatores inerentes a NCS no desempenho e estabilidade do sistema, bem como o desenvolvimento de ferramentas específicas para simulação desses sistemas. E outra no desenvolvimento de metodologias de projeto e controle de NCS, de forma a compensar a influência desses fatores e garantir o cumprimento dos requisitos de desempenho e estabilidade do NCS.

3. FERRAMENTAS DE ANÁLISE E SIMULAÇÃO DE NCS

3.1. CONSIDERAÇÕES INICIAIS

Neste capítulo é realizada uma revisão sobre o desenvolvimento e a utilização de ferramentas de análise, simulação e projeto de NCS, com enfoque no protocolo CAN. Um detalhado levantamento de características, requisitos de *software*, vantagens e usabilidade de diversas ferramentas é apresentado, objetivando avaliar a melhor (es) opção (ões) para o desenvolvimento de NCS e utilização neste trabalho. Também é apresentada uma tabela de comparação das ferramentas, contendo uma sistematização das principais informações e características coletadas. Busca-se a partir da sistematização das informações desta seção, gerar uma documentação de referência para orientar trabalhos de pesquisa e desenvolvimento sob o tema de NCS.

3.2. SÍNTESE DAS FERRAMENTAS ANALISADAS E CRITÉRIOS COMPARATIVOS ADOTADOS

O projeto e o desenvolvimento de um NCS é essencialmente um problema que envolve conhecimentos da teoria de controle, de redes de comunicação e de sistemas de tempo real. As escolhas e definições relacionadas ao projeto do sistema de tempo real do NCS afetarão o projeto do controle e vice-versa. Por exemplo, a decisão de utilização de um tipo de rede ou protocolo de comunicação em específico acarretará em uma determinada característica para os atrasos de comunicação do NCS que deverão ser levadas em consideração no projeto da

estratégia de controle do sistema. Por outro lado, requisitos de largura de banda para utilização pelas malhas de controle certamente irão influenciar na definição da velocidade de transmissão da rede de comunicação e nas características de *hardware* do NCS. A necessidade de uma abordagem multidisciplinar é ainda mais acentuada quando se trata do projeto e desenvolvimento de NCS.

De modo a simplificar e facilitar as tarefas de análise, simulação e projeto de NCS, diversas ferramentas de suporte têm sido desenvolvidas (TORNGREN et al., 2006). Essas ferramentas fornecem facilidades para analisar o comportamento do sistema de controle por meio da simulação. A simulação facilita a avaliação de resultados, pois permite a repetição de simulações para o mesmo modelo do NCS variando-se apenas os parâmetros desejados e obtendo-se assim informações mais específicas (CERVIN et al., 2003). A partir desta potencialidade, o projetista consegue avaliar as mudanças feitas ou parâmetros definidos no projeto do NCS, com respeito a métricas relacionadas ao desempenho desses sistemas (ACTON et al., 2008). Assim a avaliação de um NCS pode ser focalizada em:

- ✓ Analisar quais fatores (atrasos de comunicação, amostragem dos sinais, parâmetros da rede de comunicação, etc.) afetam o desempenho de um NCS;
- ✓ Analisar métricas de desempenho relacionadas ao controle e estabilidade do NCS (tempo de acomodação, máximo de pico, tempo de subida, erro em regime, margem de estabilidade e de ganho);
- ✓ Analisar métricas de desempenho relacionadas à rede de comunicação do NCS (taxa de utilização da rede, atrasos de comunicação, tempo de transmissão de mensagens e *jitter*);
- ✓ Analisar métricas de desempenho relacionadas à sistemas de tempo real do NCS (tempo de resposta, escalonamento de tarefas, cumprimento de *deadlines*).

A maioria dessas ferramentas, que englobam os assuntos da teoria de controle, sistemas de tempo real e redes de comunicação, tem sido baseada no ambiente de desenvolvimento *Matlab/Simulink*, impulsionado por sua grande difusão e aceitação no meio acadêmico e industrial. Um levantamento de informações sobre algumas destas ferramentas é realizado e apresentado. Esse levantamento buscou selecionar ferramentas de acordo com o requisito deste trabalho de estudo de NCS com o protocolo CAN. As ferramentas analisadas neste trabalho foram:

- ✓ *AIDA* (REDELL, EL-KHOURY & TORNGREN, 2004), desenvolvida pelo Instituto Real de Tecnologia na Suécia;
- ✓ *TORSCHÉ* (SUCHA et al., 2006), desenvolvido pela Universidade Técnica Tcheca em Praga na Republica Tcheca;
- ✓ *TrueTime* e *Jitterbug* (CERVIN et al., 2003), desenvolvidas pela Universidade de Lund na Suécia;
- ✓ *PiccSIM* (NETHI et al., 2007a), desenvolvido pela Universidade de Tecnologia de Helsinque na Finlândia;
- ✓ *NCS_Simu* (YANG, 2007), desenvolvido pela Universidade de Sussex na Inglaterra;

Para cada uma dessas ferramentas é apresentada uma descrição geral, que descreve suas propostas de utilização, e uma abordagem de sua arquitetura de desenvolvimento. Uma avaliação mais detalhada sobre vários aspectos de utilização e características de cada ferramenta é apresentada após essa introdução, de forma a se obter dados para uma comparação. Esses critérios de comparação são divididos em duas áreas principais, sendo uma relacionada ao contexto e finalidade da ferramenta e outra às características da tecnologia da ferramenta.

A subárea sobre o contexto e finalidade das ferramentas aborda aspectos como:

- ✓ áreas de aplicação;
- ✓ etapas de projeto (modelagem da planta, projeto de controladores, definição e configuração da rede) suportadas;
- ✓ vantagens e desvantagens da ferramenta.

A subárea sobre as características de tecnologia das ferramentas aborda aspectos mais específicos como:

- ✓ requisitos de operação (conhecimento de *softwares* necessários e conceitos e informações necessários sobre o NCS);
- ✓ dados de entrada e dados de saída;
- ✓ usabilidade (grau de facilidade de uso);
- ✓ disponibilidade (possibilidade de download, necessidade de registro, tipo de licença de uso) da ferramenta.

3.2.1. AIDA

3.2.1.1. DESCRIÇÃO DA FERRAMENTA

A ferramenta *AIDA* (*Automatic Control in Distributed Applications*), desenvolvida pelo Laboratório de Mecatrônica do Instituto Real de Tecnologia (KTH) em Estocolmo na Suécia, é um ambiente de desenvolvimento voltado para projeto e análise de modelos de sistemas de controle distribuído em tempo real. O desenvolvimento da ferramenta *AIDA* foi baseado em

outra ferramenta previamente desenvolvida pelo mesmo grupo chamada *XILO* (*X-in-the loop simulation*) (EL-KHOURY & TORNGREN, 2001).

Essa ferramenta permite ao projetista do sistema levar em consideração efeitos de implementação como atrasos de comunicação, variações de tempo de execução de tarefas, escalonamento e parâmetros de comunicação na etapa de análise de desempenho do sistema de controle (REDELL, EL-KHOURY & TORNGREN, 2004). Uma análise temporal do sistema modelado pode ser analisada, fornecendo informações a respeito do cumprimento de requisitos temporais e de controle.

O ambiente de desenvolvimento da ferramenta *AIDA* é baseado em duas outras ferramentas: *DOME* (*Domain Modeling Environment*) da *Honeywell* e *Matlab/Simulink* da *MathWorks*. A parte relacionada ao projeto do controle é implementada no ambiente *Matlab/Simulink* enquanto que a parte relacionada à arquitetura do sistema e as análises de parâmetros de tempo real são realizadas em um ambiente desenvolvido em *DOME*.

Figura 10 apresenta um fluxograma de utilização da ferramenta *AIDA*. Esse fluxograma permite verificar as quatro etapas principais necessárias para o funcionamento da ferramenta. Primeiramente, o sistema é projetado e testado de acordo com especificações de controle. Essa parte é realizada diretamente no ambiente *Matlab/Simulink* a através de simulações. O sistema de controle projetado fornece todas as informações necessárias, como o algoritmo de controle e parâmetros da dinâmica do sistema, para a etapa de importação da ferramenta.

cálculo de métricas de desempenho de redes de comunicação e sistemas de tempo real como atrasos de comunicação e tempo de resposta e *jitter* de tarefas e mensagens.

Nesta etapa do projeto do sistema, o usuário pode analisar os resultados obtidos e caso necessário continuar com a utilização da ferramenta para obtenção dos resultados relacionados a métricas de desempenho de controle do sistema. Para continuar a utilização, o modelo em fluxograma expandido na ferramenta *AIDA* é exportado de volta para o ambiente *Matlab/Simulink*. O novo modelo obtido representa uma cópia do modelo original criado, expandido com todas as características de sistemas de tempo real e redes de comunicação especificadas pelo usuário. Todas essas características são levadas em consideração nas novas simulações realizadas no *Matlab/Simulink*, obtendo resultados relacionados ao desempenho de controle do sistema projetado.

3.2.1.2. ARQUITETURA DE DESENVOLVIMENTO

A arquitetura de desenvolvimento da ferramenta *AIDA* é baseada em três partes principais mostradas na Figura 11 que são:

- ✓ *Aidasign*: ambiente desenvolvido totalmente no *software DOME* para a modelagem do sistema de tempo real. Essa modelagem é definida com a criação de diversos modelos em fluxograma que representam características do sistema de tempo real como tipo de *hardware* utilizado, parâmetros de comunicação e escalonamento de tarefas e mensagens. Esse ambiente possui a característica de ser facilmente estendido para adição de novos modelos ou funções escritas na linguagem Alter, relacionada ao *software DOME*;

- ✓ *Aidalyze*: ambiente desenvolvido em linguagem C++ para a análise das propriedades temporais do sistema de controle em tempo real. Esse ambiente implementa diversas técnicas e metodologias disponíveis na literatura (REDELL, EL-KHOURY & TORNGREN, 2004) para análise de desempenho de sistemas distribuídos em tempo real. Possui código fonte aberto permitindo a inclusão de novos algoritmos de análise;
- ✓ Interface *Matlab/Simulink* com *Aidasign*: realiza a interconexão entre os ambientes *Aidasign* e *Matlab/Simulink*, possibilitando a importação de modelos construídos no *Simulink* e sua tradução em modelos em fluxograma utilizados no ambiente *Aidasign* e posterior exportação e tradução de modelos do ambiente *Aidasign* para o *Simulink*. Como as ferramentas *Matlab/Simulink* e *DOME* não possuem uma linguagem de programação em comum que possibilitaria a intercomunicação entre elas, toda essa interconexão é realizada através de arquivos de *scripts* que são gerados em uma ferramenta para realização de tarefas na outra e vice versa.

Figura 11. Descrição Geral da Arquitetura da Ferramenta AIDA

No ambiente *Aidasign* da ferramenta *AIDA* são construídos diversos modelos ou diagramas que representam as características do sistema de tempo real que serão modeladas. Redell, El-Khoury & Torngren (2004) descreve e apresenta os sete modelos ou diagramas que são utilizados na ferramenta *AIDA*:

- ✓ Diagrama de Fluxo de Dados (*data-flow diagram DFD*): modelo que define funções específicas de acordo com as funcionalidades do sistema e determina os fluxos de dados e troca de informações entre essas funções. Essas funções são parametrizadas pelos seus tempos de execução mínimos e máximos e pela quantidade de dados (*bytes*) que ela transmite;
- ✓ Diagrama de Disparo e Temporização de Funções (*function timing and triggering diagram FTTD*): modelo que descreve as sequências de relações de precedência entre as funções do sistema (lei de controle) e a ocorrência dos disparos dessas sequências utilizando disparadores periódicos baseados em tempo (*time triggers*) ou disparadores aperiódicos baseados em eventos (*event triggers*). Esses disparadores possuem os seguintes parâmetros: período, *jitter* e *clock* no qual o mesmo está relacionado;
- ✓ Diagrama da Estrutura do *Hardware* (*hardware structure diagram HSD*): modelo que descreve a estrutura de *hardware* do computador ou dispositivo onde o sistema será implementado e de seus componentes, em termos de processadores e canais de comunicação de dados. Na ferramenta estão disponíveis a definição de características de processadores, como velocidade e tempo de execução de funções, para o *hardware* e canais de comunicação baseados no protocolo CAN, com a definição da velocidade de transmissão da rede e uso do *frame* de dados com identificador de 11 bits. Nesse modelo, as funções e fluxo de dados são mapeadas de acordo com os processadores e canais de comunicação existentes no sistema;
- ✓ Diagrama de Disparo e Temporização de Processos (*process timing and triggering diagram PTDD*): modelo utilizado para mapear as funções alocadas para cada processador existente no sistema em processos. Esse modelo especifica também

para cada processador, as características de disparo e temporização do seu conjunto de processos, os quais recebem prioridade para fins de escalonamento. A execução desses processos pode ser disparada por relações de precedência (outro processo finalizado), pelo recebimento de *frames* ou mensagens CAN e por outros disparadores baseados em tempo ou eventos. Os processos nos PTTDs são semelhantes às funções nos FTTDs;

- ✓ Diagrama de Estrutura do Processo (*process structure diagram PSD*): modelo que descreve a transmissão de mensagens entre processos, baseados nas informações dos fluxos de dados dos DFD e dos processos descritos nos PTTDs. As mensagens são compostas por fluxos de dados que são transmitidos entre funções de diferentes processos;
- ✓ Diagrama do Canal de Comunicação (*communication link diagram CLD*): modelo que define para cada canal de comunicação ou rede CAN utilizada, as mensagens de comunicação definidas no PSD. Os frames CAN podem conter mais de uma mensagem de comunicação, desde que não exceda o tamanho máximo de oito bytes. Esse modelo não suporta a fragmentação de uma mensagem CAN em múltiplos frames, sendo a o identificador (que define a prioridade) da mensagem CAN o único parâmetro determinado pelo usuário;
- ✓ Diagrama Interno de Disparo e Temporização de Processos (*process internal timing and triggering diagram PiTTD*): modelo que descreve para cada processo do sistema, as relações de precedência entre as funções alocadas para aquele processo, definindo assim sua sequência de execução de funções. Esse modelo é necessário para descrever precisamente os disparos e temporizações das funções que compõe um processo do sistema.

Figura 12 apresenta a estrutura de alguns desses modelos ou diagramas da ferramenta *AIDA* definidos para um sistema de controle simples.

Figura 12. Estrutura de Diagramas do Ambiente *Aidasign* da Ferramenta *AIDA*

3.2.1.3. CRITÉRIOS DE AVALIAÇÃO DA FERRAMENTA

Áreas de Aplicação

A ferramenta *AIDA* é um ambiente de desenvolvimento que integra tarefas de especificação de requisitos e análises de desempenho de sistemas de controle via redes e sistemas de controle com processadores operando em tempo real. Esta ferramenta permite que vários aspectos de projeto e desenvolvimento do sistema possam ser descritos, sendo eles desde especificações e requisitos do sistema de controle até características de sua

implementação em um sistema de processadores distribuído através de uma rede de comunicação.

A principal finalidade desta ferramenta é auxiliar o usuário ou projetista do sistema a avaliar uma grande quantidade de projetos e parâmetros de configuração para o sistema antes de sua implementação real (REDELL, EL-KHOURY & TORNGREN, 2004). Nas diversas iterações da ferramenta pode ser analisada a influência no sistema projetado de alterações na estrutura do *software* e do *hardware* utilizados, no esquema de prioridades de mensagens e processos, no escalonamento de funções e processos e na utilização de protocolos de comunicação como canais de comunicação. A avaliação do sistema pode ser baseada também considerando métricas de desempenho relacionadas ao sistema de tempo real, como tempo de resposta, escalonamento de tarefas e presença de *jitter* e relacionadas ao controle, como tempo de acomodação, máximo de pico e tempo de subida.

Etapas de Projeto Suportadas

A ferramenta *AIDA* pode ser utilizada em vários estágios de desenvolvimento e projeto de um NCS. Além de permitir a modelagem da planta ou processo e o desenvolvimento do controlador do sistema, ambos realizados diretamente no ambiente *Matlab/Simulink*, essa ferramenta permite comparar e avaliar características relacionadas a arquiteturas de *hardware* e *software* que podem ser utilizadas na implementação do NCS. Em relação à rede de comunicação, como citado anteriormente, somente a utilização de redes CAN está disponível na ferramenta. Além disso, somente os parâmetros velocidade de transmissão de dados da rede e prioridade das mensagens podem ser definidos e analisados pelo usuário.

Vantagens e Desvantagens

As vantagens da utilização da ferramenta *AIDA* para o desenvolvimento de NCS dizem respeito à possibilidade de obtenção de diversos resultados referente a métricas do sistema de tempo real, como tempos de execução de tarefas, *jitter*, tempos de resposta e utilização de

processadores, não encontrado usualmente em outras ferramentas. A grande quantidade de modelos em fluxograma relacionada às características de operação do NCS em tempo real que necessitam ser implementados acaba proporcionando um alto índice de compreensão do funcionamento do sistema e possibilitando a análise da influência de diversos parâmetros no desempenho do sistema. Outra vantagem interessante da ferramenta é a interface direta com o ambiente *Matlab/Simulink*, muito utilizado nas áreas acadêmicas e industrial, permitindo a modelagem do processo e todas as simulações relacionadas ao projeto do sistema de controle a ser utilizado. Existe também a possibilidade da não utilização do *Matlab/Simulink* junto com a ferramenta *AIDA*. Assim, o sistema de controle e o processo não são definidos e somente a modelagem do sistema no ambiente *Aidasign* é realizada e analisada no ambiente *Aidalyze*. Este fato proporciona maior flexibilidade de aplicação da ferramenta para análise de características estritamente relacionadas ao sistema de tempo real do NCS.

A ferramenta *AIDA* apresenta como desvantagens a impossibilidade de obtenção de dados sobre métricas de desempenho relacionadas à rede de comunicação como a taxa de utilização da rede e atrasos de comunicação, além do fato de permitir somente a análise de NCS com redes baseadas no protocolo CAN. Outra desvantagem é a necessidade de conhecimento de características e parâmetros do NCS a ser projetado, para utilização da ferramenta pelo projetista ou usuário. Dados como tempos de execução máximos e mínimos de tarefas e características do processador onde o sistema será implementado, como velocidade de processamento, são muitas vezes difíceis de serem mensurados ou definidos previamente e acabam sendo negligenciados nos projetos desenvolvidos. Outro fato importante de se citar é que existe certa dificuldade para aprender a utilizar o ambiente da ferramenta para construção dos diversos modelos em fluxogramas do NCS e também para o desenvolvimento desses modelos. Esses fatos muitas vezes acabam sendo mencionados como desvantagens dessa ferramenta.

Dados de Entrada

Os dados de entrada requeridos para o funcionamento da ferramenta *AIDA* são definidos de acordo com as duas possibilidades de utilização da ferramenta. Quando o ambiente *Matlab/Simulink* é utilizado juntamente com a ferramenta, são necessários também a definição através dos blocos do *Simulink* da função de transferência da planta ou processo modelado e do controlador desenvolvido para o sistema de controle do NCS.

Os dados de entrada requeridos estão relacionados com as características do sistema inseridas nos modelos em fluxograma construídos na ferramenta. O usuário da ferramenta necessita definir ou estimar os tempos máximos e mínimos de execução das tarefas e funções especificadas nos processos de todos os modelos, definir as características de todos os processadores e canais de comunicação utilizados e fornecer os parâmetros (quantidade de *bytes* e tamanho das mensagens) referentes à troca de informações em cada fluxo de dados mapeado nos modelos em fluxograma desenvolvidos.

Dados de Saída

A utilização da ferramenta *AIDA* possibilita obter dados de saída de um NCS como os tempos, no pior caso (*worst-case*) e no melhor caso (*best-case*), de resposta e de execução de todas as tarefas (ou funções) e processos criados e os tempos de transmissão das mensagens em um canal de comunicação ou rede CAN, além dos valores do *jitter* induzidos pelo sistema. A ferramenta calcula e fornece também a taxa (ou porcentagem) de utilização de cada processador e canal de comunicação definido no sistema.

Quando a ferramenta é utilizada juntamente com o ambiente *Matlab/Simulink*, todos os resultados obtidos são exportados e traduzidos para o modelo expandido do *Simulink*. Esse modelo expandido, contendo todas as informações e propriedades temporais modeladas na ferramenta, pode ser utilizado para analisar o desempenho do NCS projetado através de simulações. Nesta etapa todas as funcionalidades do ambiente *Matlab/Simulink* podem ser

utilizadas para geração de dados de saída relacionados a métricas de desempenho de controle (tempo de acomodação, máximo de pico, tempo de subida) como gráficos gerados com blocos do *Simulink* (*scope*) e dados coletados em variáveis do *Matlab* (*workspace*).

Usabilidade e Outras Informações

De acordo com as características analisadas, define-se a ferramenta *AIDA* como de nível de dificuldade mediano para utilização e difícil para aprendizado para a realização de tarefas relacionadas ao desenvolvimento de NCS. Os requisitos de conhecimentos para o uso da ferramenta estão relacionados ao *software Matlab/Simulink* e ao aprendizado do ambiente de desenvolvimento em *DOME* para construção dos modelos em fluxograma de sistema. A necessidade de entrada de dados específicos sobre o sistema (como tempo de execução de funções de processamento, leitura de dados e transmissão de mensagens) nesta etapa de utilização da ferramenta, acaba dificultando sua utilização.

A ferramenta possui uma arquitetura de desenvolvimento aberta, o que representa a disponibilidade de implementações por parte de usuários no sentido de estender suas funcionalidades de acordo com suas necessidades específicas. O ambiente *Aidasign* pode ser facilmente estendido para adição de novos modelos ou funções escritas na linguagem Alter (relacionada ao *software DOME*) e o ambiente *Aidalyze* possui código fonte aberto (C++) permitindo a inclusão de novos algoritmos de análise.

De acordo com as duas possibilidades de utilização da ferramenta *AIDA*, o requisito inicial de disponibilidade da ferramenta *Matlab/Simulink* pode ou não ser necessário.

Disponibilidade

A ferramenta *AIDA* é um ambiente de desenvolvimento *freeware*, que não requer licença ou registro para funcionamento. Para obtenção da ferramenta via download é necessário contato com os desenvolvedores. Maiores informações podem ser encontradas no endereço <http://www.md.kth.se/RTC/aida/>.

3.2.2. TORSCHÉ

3.2.2.1. DESCRIÇÃO DA FERRAMENTA

A ferramenta *TORSCHÉ* (*Time Optimization of Resources, Scheduling*), desenvolvida pelo Departamento de Engenharia de Controle da Universidade Técnica Tcheca em Praga na República Tcheca, é um *toolbox* do *Matlab* composto por algoritmos de escalonamento que podem ser utilizados com o intuito de otimizar a resposta temporal de um NCS. Com essa otimização consegue-se atingir parâmetros de tempo real desejados para o sistema de controle, como tempo de resposta e atraso de comunicação (*delay*) (SUCHA et al., 2006). A ferramenta pode ser utilizada também para verificar o desempenho e operação de um NCS antes de sua implementação física, através da comparação de métricas de desempenho relacionadas ao sistema de tempo real com os requisitos definidos no projeto.

De acordo com o algoritmo de controle e recursos computacionais especificados para o sistema, a ferramenta permite analisar parâmetros como período de amostragem e *jitter*. De acordo com a literatura (LIAN, MOYNE & TILBURY, 2002), esses parâmetros exercem forte influência no desempenho de um NCS. A ferramenta *TORSCHÉ* permite também avaliar o desempenho de controle do NCS através de uma interface com outra ferramenta chamada *TrueTime* (CERVIN et al., 2003), com a finalidade de otimizar os parâmetros do controlador ou escolher um novo algoritmo de controle para o projeto do NCS.

A utilização desta ferramenta, desenvolvida de acordo com o paradigma da orientação a objetos, é baseada em um conjunto de funções ou rotinas do *Matlab* com sintaxe definida, especificados para cada dispositivo do sistema. Essas funções permitem ao usuário da ferramenta formalizar o projeto do sistema, escolhendo o algoritmo de escalonamento

apropriado entre os diversos disponíveis na ferramenta (KUTIL et al., 2007) e considerando diversas configurações do mesmo. Entre essas configurações estão a seleção dos recursos de *hardware* utilizados, como arquiteturas baseadas em FPGA (*Field Programmable Gate Array*), em microcontroladores ou em processadores de tempo real, a configurações de tarefas, como tempo de processamento, *deadline*, peso ou prioridade e latência e também critérios de otimização, como minimização de latência e cumprimento de *deadlines* (execução de tarefas em tempo menor que seu *deadline*).

Outra função bastante interessante da ferramenta *TORSCHÉ* é a possibilidade de geração de códigos para utilização em outras plataformas e *hardwares*. Entre essas possibilidades estão a geração de funções para interface com a ferramenta *TrueTime* no *Matlab/Simulink* e a geração de códigos de escalonamento para funcionamento em processadores baseados em FPGA.

3.2.2.2. ARQUITETURA DE DESENVOLVIMENTO

A ferramenta *TORSCHÉ* foi desenvolvida em linguagem de programação do *Matlab* (arquivos MEX) e utiliza o paradigma da orientação a objetos. A utilização da ferramenta é feita diretamente no ambiente de desenvolvimento do *Matlab* como um *toolbox* adicional do *software*.

Na arquitetura da ferramenta *TORSCHÉ*, os principais objetos são a tarefa (*task*), o conjunto de tarefas (*taskset*) e o problema (*problem*). O objeto tarefa é uma estrutura de dados composta por parâmetros de configuração de uma tarefa como tempo de processamento e *deadline* entre outros. Esse tipo de objeto é geralmente agrupado em conjuntos, definindo o objeto conjunto de tarefas. Esses conjuntos de tarefas possuem outras informações e

características como relações de precedência e sequência de execução das tarefas. O objeto problema é uma estrutura que descreve e define as características do sistema que serão usadas pelo algoritmo de escalonamento utilizado. Esse objeto obedece à notação proposta em Blazewicz et al. (2001) para apresentação e discussão de problemas de escalonamento, a qual é amplamente utilizada na literatura de sistemas de tempo real. Esses objetos são utilizados como base para a ferramenta *TORSCHÉ*, proporcionando uma maior flexibilidade e facilitando sua expansão através da incorporação de novos algoritmos e funcionalidades.

O objeto tarefa representa o termo básico da ferramenta *TORSCHÉ*. Este objeto possui uma série de parâmetros de configuração que definem as características da tarefa a ser executada pelo sistema. As tarefas podem ser periódicas (se repete a cada período de tempo) ou não. A definição de uma tarefa (T_j) na ferramenta pode ser descrita pelos seguintes parâmetros mostrados na Figura 13:

- ✓ Nome (*Name*): define o nome ou rótulo da tarefa;
- ✓ Tempo de processamento - p_j (*ProcTime*): define o tempo necessário para a execução da tarefa T_j sem interrupções (também chamando de tempo de computação);
- ✓ Tempo de liberação - r_j (*ReleaseTime*): define o tempo ou momento em que a tarefa está pronta para ser executada (também chamada de tempo de pedido)
- ✓ *Deadline* - D_j : define o tempo limite em que uma tarefa tem que ser executada sem acarretar problemas para o sistema ou causar a falha de um escalonamento;
- ✓ Tempo adequado - d_j (*DueDate*): define o tempo limite em que uma tarefa deve ser executada sem que ocorra penalidades;
- ✓ Peso ou Prioridade (*Weight*): define a prioridade da tarefa em relação às outras tarefas do sistema;

- ✓ Processador (*Processor*): define o tipo de processador em que a tarefa será executada;
- ✓ Período: parâmetro definido somente com a utilização de tarefas periódicas no sistema.

Figura 13. Diagrama de Representação dos Parâmetros de uma Tarefa na ferramenta TORSCHÉ (Kutil et al., 2007)

Com a simulação do NCS na ferramenta, podem-se obter os outros parâmetros, mostrados na Figura 13, relacionados a métricas de desempenho de sistemas de tempo real. Esses parâmetros são:

- ✓ Tempo de execução - C_j (*completion time*): define o tempo real de execução de uma tarefa no sistema;
- ✓ Tempo de início - s_j (*starting time*): define o tempo real de início de execução de uma tarefa no sistema;
- ✓ Tempo de espera - w_j (*waiting time*): define o tempo de espera do processador entre o instante em que uma tarefa esta pronta para ser executada e o instante em que ela começa a ser executada;
- ✓ Tempo de tarefa F_j (*flow time*): define o tempo total de execução de uma tarefa (desde sua liberação até o termino de sua execução);

- ✓ Tempo relativo – L_j ou D_j : define o intervalo de tempo entre o tempo real de execução da tarefa no sistema e seu tempo adequado. Valores negativos (L_j) para esse tempo relativo são sempre requisitos para um bom funcionamento do NCS.

Sintaxe de Utilização da Ferramenta no Ambiente do Matlab

A utilização da ferramenta *TORSCHÉ* é baseada em um conjunto de funções ou rotinas do *Matlab* com sintaxe definida, especificados para cada dispositivo do sistema. Essas funções ou rotinas são inseridas diretamente nas linhas de comando do ambiente *Matlab*. Para o objeto tarefa, a função utilizada tem as seguintes regras de sintaxe:

- ✓ $t1 = \text{task}([\text{Name},] \text{ProcTime} [\text{ReleaseTime} [\text{Deadline} [\text{DueDate} [\text{Weight} [\text{Processor}]]]])$ para tarefas não periódicas;
- ✓ $pt1 = \text{ptask}([\text{Name},] \text{ProcTime}, \text{Period} [\text{ReleaseTime} [\text{Deadline} [\text{Duedate} [\text{Weight} [\text{Processor}]]]])$ para tarefas periódicas.

Para a definição de objetos tarefa em um sistema, apenas os parâmetros tempo de processamento e período são obrigatórios. Os outros parâmetros (entre colchetes nas funções) são opcionais para o usuário da ferramenta.

Uma sequência de tarefas com a finalidade de realizar determinada função em um sistema pode ser agrupada em um objeto chamando conjunto de tarefas. A definição de um conjunto de tarefas na ferramenta é dada pela seguinte função:

- ✓ $T = \text{taskset}(\text{tasks}[\text{prec}])$

Nessa função a variável T define o nome do conjunto de tarefas, a variável *tasks* representa uma matriz de objetos tarefa ou um vetor de valores de tempo de processamento das tarefas contidas no conjunto e a variável *prec* define uma matriz quadrada ou um gráfico,

como mostrado na Figura 14, contendo as relações de precedência e sequência de execução das tarefas que compõe o conjunto definido. Quando são utilizados os gráficos, os nós representam as tarefas e os arcos representam as relações de precedência e sequência de execução. Esses gráficos são construídos em um ambiente chamado *Graphedit* da ferramenta *TORSCHÉ*.

Figura 14. Utilização de Gráfico para Definição de Relações de Precedência entre um Conjunto de Tarefas (Kutil et al., 2007)

O objeto problema é uma estrutura que descreve as características do sistema que serão usadas pelo algoritmo de escalonamento aplicado ao NCS. A definição de um problema na ferramenta é dada pela seguinte função:

$$\checkmark \quad p = \text{problem}('P|\text{prec}|C_{\max}')$$

Esse objeto consiste de três partes, de acordo com a notação proposta em Blazewicz et al. (2001). A primeira parte define as características relacionadas ao processador como quantidade e tipo (por exemplo P – processador simples e F – processador dedicado). A segunda parte define as características relacionadas com as tarefas e recursos como fornecimento de *deadlines* (d_j) e tempos de liberação diferentes para tarefas (r_j), permissão de preempção de tarefas ($pmtn$) e utilização de relações de precedência e sequência de tarefas

(*prec*). A terceira parte define critérios de otimização a serem utilizados no sistema como minimização de latência (*Cmax*) e cumprimento de *deadlines*. Maiores informações sobre as possibilidades de configuração e definição de objetos problema podem ser encontrados no guia de referências de funções (*reference guide*) do manual de utilização da ferramenta *TORSCHÉ* (KUTIL et al., 2007).

A definição do algoritmo de escalonamento que será aplicado ao NCS também é realizada a partir de uma função do *Matlab*. Essa função possui pelo menos dois parâmetros de entrada e um parâmetro de saída. Os parâmetros de entrada obrigatórios são o conjunto de tarefas a ser escalonado e o objeto problema que contém as características definidas para o algoritmo de escalonamento. O conjunto de tarefas escalonado representa o parâmetro de saída da função. A sintaxe definida para utilização dessa função é a seguinte:

✓ `TS = name(T,problem[,processors[,parameters]])`

Na função descrita, o parâmetro *TS* representa uma variável que conterá um conjunto de tarefas escalonado de acordo com as características definidas para o sistema. O parâmetro *name* representa o nome do algoritmo que será utilizado. O parâmetro *problem* representa o objeto problema que contém as características definidas para o algoritmo de escalonamento. O parâmetro *processors* define o número de processadores utilizado pelo sistema e o parâmetro *parameters* especifica dados adicionais relacionados à estratégia do algoritmo escolhido.

Funções Adicionais da Ferramenta

Além do ambiente Graphedit para construção dos gráficos que contem as relações de precedência e sequências de execução de um conjunto de tarefas, a ferramenta *TORSCHÉ* apresenta outras funções adicionais muito interessantes. A ferramenta CSSIM (*Cyclic Scheduling Simulator*) possui funções desenvolvidas que permitem a simulação do NCS utilizando outra ferramenta chamada *TrueTime* (CERVIN et al., 2003). A ferramenta

adicional CSSIM possui três etapas de operação sendo a primeira com a importação do sistema desenvolvido no *Matlab* para a estrutura utilizadas na ferramenta (realizada pela função *cssimin*), a segunda com a aplicação do algoritmo de escalonamento ao sistema (realizado pela função *cyclic scheduling*) e a terceira com a geração de códigos (um *script* de inicialização e uma rotina para execução das tarefas escalonadas) para a ferramenta *TrueTime* (realizada pela função *cssimout*). Outra função adicional que pode ser citada é a possibilidade de exportação para o formato XML de todos os objetos e suas informações contidas em um sistema.

3.2.2.3. CRITÉRIOS DE AVALIAÇÃO DA FERRAMENTA

Áreas de Aplicação

A ferramenta *TORSCHÉ* pode ser utilizada para o projeto de um NCS levando-se em consideração diversos parâmetros do sistema como estratégias de escalonamento de tarefas, limitações ou indisponibilidade de recursos, características de *hardware* e tarefas e critérios para otimização do sistema. A principal aplicação da ferramenta é na análise de métricas de desempenho relacionadas a sistemas de tempo real. A ferramenta permite o cálculo de diversos parâmetros como tempos de resposta de tarefas e do sistema, permite a análise dos efeitos dos períodos de amostragem e *jitter* das tarefas no desempenho do sistema e permite a avaliar a operação do sistema com a definição de características reais de implementação como quantidade de processadores e tipo de *hardware* utilizado. Para suprir alguma eventual necessidade de se obter dados sobre métricas de desempenho relacionadas ao controle do sistema, como tempo de acomodação, máximo de pico, tempo de subida, a ferramenta *TORSCHÉ* oferece uma interface com a ferramenta *TrueTime* para a realização dessas tarefas.

Etapas de Projeto Suportadas

A utilização da ferramenta *TORSCHÉ* é focada na área de projeto e avaliação de NCS levando em consideração propriedades de sistemas de tempo real. Diversas características relacionadas a esse tipo de sistema podem ser analisadas como parâmetros de execução de tarefas, seleção de recursos de *hardware*, aplicação de algoritmos de escalonamento critérios de otimização de métricas de desempenho como latência, atraso de comunicação e tempo de resposta.

Levando-se em consideração apenas o ambiente de desenvolvimento da ferramenta, não é possível a realização da modelagem do processo e do controlador utilizados no NCS. No entanto com a disponibilidade de interface da ferramenta *TORSCHÉ* com a ferramenta *TrueTime*, essas etapas de projeto também podem ser realizadas. A definição de características relacionadas à utilização de uma rede de comunicação não é feita diretamente através da seleção de um tipo de rede ou protocolo. Como a ferramenta *TORSCHÉ* não oferece suporte para seleção de nenhum tipo de rede, toda a definição de características de uma rede é realizada através da configuração de tarefas (por exemplo, tempo de transmissão da mensagem, *jitter*, período de amostragem e outros parâmetros disponíveis). Esse fato fornece para a ferramenta a flexibilidade de poder analisar sistemas operando com qualquer tipo de rede. No entanto para que isso seja possível, é necessário conhecimento prévio do usuário da ferramenta, sobre o protocolo ou rede de comunicação utilizado e sobre esses parâmetros de configuração requeridos. De acordo com a literatura (LIAN, MOYNE & TILBURY, 2002), esses parâmetros de configuração de redes de comunicação são difíceis de serem corretamente mensurados e exercem forte influência no desempenho de um NCS.

Vantagens e Desvantagens

Uma vantagem de utilização da ferramenta *TORSCHÉ* é o grande número de algoritmos de escalonamento disponíveis para serem aplicados no desenvolvimento do NCS. Como essa

ferramenta visa quase que unicamente o desenvolvimento e a avaliação de sistemas de controle em tempo real, uma grande quantidade de métricas de desempenho podem ser obtidos e parâmetros de configuração analisados. Entre essas métricas e parâmetros, um diferencial em relação às outras ferramentas que realizam análises de sistemas de tempo real, está na possibilidade de selecionar parâmetros como tempo de liberação de tarefas e recursos de *hardware* que serão utilizados para a implementação do sistema, como arquiteturas baseadas em FPGA e em microcontroladores. Também se consegue obter métricas de desempenho como tempo de espera de tarefas e dados de operação do sistema após aplicação do algoritmo de escalonamento.

No entanto a principal vantagem de aplicação da ferramenta se encontra na possibilidade de geração de gráficos, que facilitam a compreensão do sistema, para todas as suas etapas de utilização. Esses gráficos são gerados facilmente através de simples comandos (*plot*) do *Matlab* e estão disponíveis para a visualização de todas as tarefas, e conjuntos de tarefas do sistema e também para a análise dos resultados de escalonamentos e de métricas de desempenho obtidas. Outra vantagem da ferramenta é sua interface direta com a ferramenta *TrueTime*, fato que acaba expandindo as funcionalidades e possibilidades de utilização da ferramenta para áreas não cobertas.

A ferramenta *TORSCH*, por ser focada no desenvolvimento de sistemas de tempo real, apresenta como desvantagens a impossibilidade de obtenção de dados sobre métricas de desempenho relacionadas a redes de comunicação como a taxa de utilização da rede e atrasos de comunicação e métricas relacionadas ao controle do sistema. Outra desvantagem é o requisito de conhecimento, pelo projetista ou usuário da ferramenta, de características e parâmetros do NCS a ser projetado. Dados como tempos de transmissão de mensagens, *jitter* e atrasos de comunicação são muitas vezes difíceis de serem mensurados e acabam dificultando a utilização da ferramenta.

Dados de Entrada

Os dados de entrada necessários para o funcionamento da ferramenta *TORSCHE* estão relacionados com as funções, explicadas no item arquitetura do sistema, definidas para cada dispositivo do sistema. De acordo com os objetos utilizados no projeto do NCS como tarefas, conjunto de tarefas, problemas e utilização de algoritmos de escalonamento, são definidos os dados de entrada requeridos.

Dados de entrada como tempos de processamento, *deadlines*, prioridades e latências para as tarefas, matrizes ou gráficos com as relações de precedência e sequências de execução para os conjunto de tarefas, seleção do número de processadores e tipos de *hardware* para os objetos problemas e definição do algoritmo de escalonamento utilizado e de seus critérios de otimização serão sempre requisitos para a operação da ferramenta. Um fato importante e que facilita o desenvolvimento do sistema é a possibilidade de visualização através de gráficos, das funções criadas para a definição dos componentes de um sistema.

Dados de Saída

A ferramenta *TORSCHE* oferece uma grande quantidade de dados de saída que podem ser obtidos de acordo com as etapas de utilização da ferramenta e desenvolvimento do NCS. Os dados de saída podem ser visualizados na forma de lista contendo os resultados da simulação ou gráficos.

Em relação a todas as tarefas e conjunto de tarefas definidas para o sistema em análise, é possível gerar gráficos, mostrados na Figura 15, que facilitam a compreensão do funcionamento do sistema.

Figura 15. Gráficos de Tarefas e Conjunto de Tarefas na Ferramenta TORSCHÉ

Essa interface gráfica é um ponto forte da ferramenta e auxilia nas tarefas de análise de resultados. Após a simulação do sistema na ferramenta é possível a visualização de gráficos que fornecem informações a respeito do escalonamento das tarefas e informações sobre as tarefas executadas como tempos de execução, tempos de espera, tempo de início, *jitter* de tarefas e mensagens, como mostrado na Figura 16.

Figura 16. Gráfico de Saída de Escalonamento e Dados sobre Tarefas na Ferramenta TORSCHÉ

Nesse tipo de gráfico obtido na ferramenta, as três primeiras linhas representam três conjuntos de tarefas a serem realizadas (o primeiro e o terceiro conjunto com duas tarefas e o

segundo conjunto com três tarefas) e a última linha representa a operação do sistema após a aplicação do algoritmo de escalonamento.

Além dos dados de saída citados, com a ferramenta *TORSCHÉ* é possível a geração automática de códigos do sistema analisado para utilização em outras plataformas (*TrueTime*) e implementação em *hardwares* (FPGA).

Usabilidade e Outras Informações

O levantamento de características realizado permite definir um nível de dificuldade mediano para utilização e aprendizado da ferramenta *TORSCHÉ* para a realização de tarefas relacionadas ao desenvolvimento de NCS. Os requisitos de conhecimentos para o uso da ferramenta estão relacionados ao *software Matlab/Simulink* e a sintaxe de utilização das funções que compõe o *toolbox*. Juntamente com o pacote de instalação da ferramenta, são fornecidos diversos exemplos de aplicação, os quais facilitam ainda mais a familiarização com a ferramenta. A ferramenta disponibiliza um grande número de algoritmos de escalonamento para aplicação da ferramenta, o qual vem aumentando conforme novas implementações e desenvolvimentos têm sido realizados por usuários.

A ferramenta possui uma arquitetura de desenvolvimento aberta, de acordo com o paradigma da orientação a objetos, o que representa a disponibilidade de desenvolvimentos de novas funcionalidades como algoritmos de escalonamento por parte de usuários. Tais implementações podem ser realizadas em funções ou códigos do *Matlab* (arquivos Mex do tipo m) ou em C++.

Disponibilidade

A ferramenta *TORSCHÉ* é um ambiente de desenvolvimento *freeware* distribuído sob os termos de licença pública (*GNU – General Public License*). Maiores informações e arquivos para download podem ser encontrados diretamente do endereço <http://rttime.felk.cvut.cz/scheduling-toolbox/>

3.2.3. JITTERBUG

3.2.3.1. DESCRIÇÃO DA FERRAMENTA

A ferramenta *Jitterbug*, desenvolvida pelo Departamento de Controle Automático da Universidade de Lund em Lund na Suécia, é um *toolbox* do *Matlab* que permite a aplicação da teoria LQG (*Linear Quadratic Gaussian*) de controle para um NCS (CERVIN, 2003). A ferramenta é composta por um conjunto de funções do *Matlab* que permitem ao usuário desenvolver e analisar NCS através de modelos com informações temporais. Com a utilização da ferramenta pode-se obter um critério quadrático de desempenho do NCS de acordo com várias condições temporais relacionadas à rede de comunicação, como presença de atrasos de comunicação, *jitter* e períodos de amostragem das mensagens. Assim, consegue-se facilmente avaliar a influência desses parâmetros de configuração no desempenho de controle do NCS.

A ferramenta pode ser utilizada também para a análise de desempenho de estratégias de controle para NCS como controladores com compensação de *jitter*. Outra característica importante da ferramenta *Jitterbug* é a possibilidade de computar um gráfico do espectro da qualidade de controle de um NCS em função dessas condições temporais como, por exemplo, uma curva que relacionaria a qualidade do controle de uma planta em malha fechada, em função do período de amostragem e do *jitter* do NCS.

Na ferramenta *Jitterbug*, um NCS é descrito através de um conjunto de sistemas lineares contínuos e discretos, que representam os componentes do NCS como a planta ou processo e o controlador desenvolvido (CERVIN & LINCOLN, 2006). Esses subsistemas são executados em sequência durante o período de controle do NCS. Opcionalmente, para cada subsistema, podem ser definidas características adicionais como perturbações, ruídos e

especificações de custo. Para cada sistema discreto, um atraso de comunicação (descrito por uma função de probabilidade) tem que ser especificado e deve ocorrer antes da execução do próximo subsistema.

O custo total do NCS (soma dos custos de todos os subsistemas) é calculado algebricamente se o modelo do sistema é periódico ou iterativamente se o modelo do sistema é aperiódico. Para o caso mais simples de utilização da ferramenta, a execução dos sistemas discretos do NCS como, por exemplo, a tarefa de controle, pode ser descrita através de modelos de sistemas periódicos com atrasos de comunicação com distribuição randômica.

3.2.3.2. ARQUITETURA DE DESENVOLVIMENTO

A ferramenta *Jitterbug* consiste em um conjunto de funções do *Matlab* que fazem interface com o *toolbox Control Systems*, permitindo a inicialização da ferramenta, configurações do sistema e o cálculo de um índice de desempenho ou uma função custo relacionada ao controle do NCS. Na ferramenta, um NCS é descrito através de dois modelos: o modelo de sinalização (*signal model*) e o modelo temporal (*timing model*). O modelo de sinalização corresponde ao diagrama de conexão dos sistemas lineares contínuos e discretos que compõe o NCS. O modelo temporal descreve a sequência de execução dos sistemas discretos pertencentes ao NCS durante um período de controle. Os atrasos de comunicação em um período de controle do NCS são considerados independentes dos atrasos existentes nos demais períodos.

Figura 17 apresenta um exemplo para os dois modelos requeridos para a utilização da ferramenta *Jitterbug* em um NCS, onde o sistema de controle pode ser descrito por quatro subsistemas. A partir do modelo de sinalização, mostrado na Figura 17(a), pode-se visualizar

que a planta ou processo é descrita por um sistema linear contínuo (G) e o sensor (H_1), o atuador (H_3) e o controlador (H_2) são descritos por sistemas discretos.

Figura 17. Modelos requeridos para a Ferramenta Jitterbug: (a) Modelo de Sinalização e (b) Modelo Temporal (Cervin & Lincoln, 2006)

A partir do modelo temporal, mostrado na Figura 17(b), pode-se visualizar que a cada período de controle do NCS, o subsistema H_1 é executado primeiramente. Atrelado ao sensor H_1 existe um atraso de comunicação randômico τ_1 . A seguir ocorre a execução do controlador H_2 , o qual também possui um atraso de comunicação τ_2 . E finalmente ocorre a execução do subsistema H_3 que representa o atuador. Os atrasos de comunicação citados podem ser utilizados para modelar atrasos relacionados a tarefas de codificação e decodificação de dados, relacionados à execução e escalonamento de tarefas e relacionados à transmissão de mensagens em uma rede de comunicação. Esta última possibilidade acaba originando uma grande flexibilidade de utilização para a ferramenta, no sentido de aplicação para análise de NCS com qualquer tipo de rede de comunicação. No entanto, para que isso seja possível, é necessário o conhecimento por parte do usuário da ferramenta da distribuição de valores para tempos de transmissão de mensagens na rede de comunicação utilizada.

Com a definição dos modelos requeridos, pode-se utilizar das funções do *Matlab* incorporadas na ferramenta *Jitterbug*, para a análise de características do NCS estudado.

Figura 18 apresenta um *script* dos comandos relacionados à definição dos modelos citados anteriormente e cálculo da função custo e índice de desempenho do NCS.

<code>G = 1000/(s*(s+1));</code>	Define the process
<code>H1 = 1;</code>	Define the sampler
<code>H2 = -K*(1+Td/h*(z-1)/z);</code>	Define the controller
<code>H3 = 1;</code>	Define the actuator
<code>Ptau1 = [...];</code>	Define delay probability distribution 1
<code>Ptau2 = [...];</code>	Define delay probability distribution 2
<code>N = initjitterbug(delta,h);</code>	Set time-grain and period
<code>N = addtimingnode(N,1,Ptau1,2);</code>	Define timing node 1
<code>N = addtimingnode(N,2,Ptau2,3);</code>	Define timing node 2
<code>N = addtimingnode(N,3);</code>	Define timing node 3
<code>N = addcontsys(N,1,G,4,Q,R1,R2);</code>	Add plant, specify cost and noise
<code>N = adddiscsys(N,2,H1,1,1);</code>	Add sampler to node 1
<code>N = adddiscsys(N,3,H2,2,2);</code>	Add controller to node 2
<code>N = adddiscsys(N,4,H3,3,3);</code>	Add actuator to node 3
<code>N = calcdynamics(N);</code>	Calculate internal dynamics
<code>J = calccost(N);</code>	Calculate the total cost

Figura 18. Comandos da Ferramenta Jitterbug para Definição dos Modelos e Cálculo da Função Custo de Desempenho de um NCS (Cervin & Lincoln, 2006)

Para um melhor entendimento das funções mostradas no *script*, recomenda-se a leitura do guia de referência de funções disponível no manual da ferramenta Jitterbug (CERVIN & LINCOLN, 2006).

3.2.3.3. CRITÉRIOS DE AVALIAÇÃO DA FERRAMENTA

Áreas de Aplicação

A principal finalidade de utilização da ferramenta *Jitterbug* é permitir a avaliação e análise de desempenho, em termos de métricas relacionadas ao controle e a estabilidade, de diferentes estratégias de controle desenvolvidas para NCS. Nesse sentido, para um controlador desenvolvido para um NCS, a ferramenta pode ser utilizada para avaliar a

sensibilidade do sistema de controle em malha fechada em relação a várias condições temporais impostas. Entre essas condições pode-se citar a presença de *jitter* e atrasos de comunicação (constantes, randômicos ou outra distribuição) relativos a tarefas ou a transmissão de mensagens em uma rede de comunicação e a ocorrência de erros de transmissão e perda de mensagens na rede. A partir da geração do gráfico do espectro da qualidade do controle do NCS em relação a esses parâmetros, o usuário consegue também analisar as condições para estabilidade do sistema. A ferramenta permite ainda avaliar métricas de desempenho relacionadas ao controle através de gráficos da magnitude da resposta em frequência do NCS na presença de atrasos de comunicação e *jitter*.

A ferramenta *Jitterbug* pode ser utilizada também para o projeto de controladores e para o desenvolvimento de metodologias de controle para NCS.

Etapas de Projeto Suportadas

A ferramenta *Jitterbug* permite a realização das etapas de modelagem da planta ou processo e de desenvolvimento do controlador para um projeto de um NCS. O processo é modelado através de sistemas lineares contínuos e sua representação é feita através de sua função de transferência no domínio do tempo (transformada de *Laplace* – s). Já o desenvolvimento do controlador do NCS é feito através de modelos de sistemas discretos e sua função de transferência é dada no domínio da transformada Z . Em relação ao desenvolvimento do controlador do NCS, essa ferramenta permite a utilização da técnica de múltiplos controladores ou controladores em cascata (CERVIN & LINCOLN, 2006).

Uma importante característica dessa ferramenta é a possibilidade de se modelar também os sensores e atuadores utilizados no NCS. Essa modelagem é feita de maneira semelhante ao controlador, isto é, através de modelos de sistemas discretos e com sua função de transferência no domínio da transformada Z . A configuração de características relacionadas à rede de comunicação do NCS é feita indiretamente na ferramenta e de maneira bastante

simplificada. Para isso são definidos valores de atrasos de comunicação e *jitter* que corresponderiam aos efeitos da inclusão de uma rede de comunicação no sistema de controle.

Vantagens e Desvantagens

A grande vantagem da ferramenta *Jitterbug* diz respeito à facilidade de aplicação da teoria LQG de controle para o desenvolvimento de NCS, no sentido de analisar a sensibilidade do sistema em relação a alterações de parâmetros de configuração. A ferramenta permite avaliar o desempenho de controle do sistema através do cálculo de uma função quadrática de custo relacionada com diversos parâmetros de configuração de NCS, como período de amostragem, latência e *jitter*. Com a obtenção do gráfico de espectro da qualidade de controle do NCS, consegue-se avaliar e quantificar os efeitos da alteração de mais de um parâmetro de configuração no desempenho e na estabilidade do sistema. Um exemplo disso seria a obtenção de uma superfície que relacionaria a sensibilidade do controle de uma planta em malha fechada, em função do período de amostragem e do *jitter* do NCS. Entre as ferramentas de suporte avaliadas neste trabalho, somente a ferramenta *Jitterbug* fornece essa possibilidade de análise.

Outra vantagem interessante da ferramenta é a possibilidade de analisar características de estabilidade do NCS a partir dos valores obtidos com o cálculo da função custo do sistema. De acordo com Cervin et al. (2003), altos valores para a função custo em um NCS indicam uma diminuição da estabilidade do sistema em malha fechada (consequentemente valores muito alto ou tendendo ao infinito significa que o NCS é instável).

Algumas desvantagens e limitações da ferramenta *Jitterbug* podem ser citadas. O modelo temporal utilizado para configuração do NCS principalmente em termos de atrasos de comunicação é bastante simplificado. Os atrasos de comunicação são considerados independentes de cada período ou ciclo de controle do NCS. Assim, o modelo acaba não conseguindo descrever variações temporais existentes no NCS originadas, por exemplo, por

algoritmos de escalonamento ou por atrasos de comunicação variáveis no tempo. A ferramenta não apresenta a possibilidade de obtenção de métricas de desempenho relacionadas ao sistema de tempo real e a rede de comunicação como tempo de resposta e taxa de utilização da rede e nem relacionadas ao controle como tempo de acomodação, máximo de pico e tempo de subida. No entanto, a partir dos gráficos de resposta em frequência do sistema, disponíveis na ferramenta, podem-se obter correlações de valores para os parâmetros citados. Para a utilização da ferramenta *Jitterbug*, é necessário especificar valores para as distribuições de atrasos de comunicação. Como esses parâmetros são geralmente desconhecidos e difíceis de obter, muitas vezes acabam sendo escolhidos incorretamente.

Dados de Entrada

Os dados de entrada requeridos para o funcionamento da ferramenta *Jitterbug* estão relacionados com as etapas de projeto do NCS suportadas pela ferramenta. Em relação ao modelo da planta ou processo a ser estudado, é necessária a definição, através de sistemas lineares contínuos, de sua função de transferência no domínio do tempo (transformada de Laplace – s). O desenvolvimento do controlador do NCS é feito através de modelos de sistemas discretos e sua função de transferência é dada no domínio da transformada Z. Na ferramenta *Jitterbug* também é necessário a definição dos modelos do atuador e do sensor utilizados no NCS. Esses modelos, como no controlador, também são descritos através de suas funções de transferência no domínio do tempo discreto (transformada Z). A configuração de características relacionadas à rede de comunicação do NCS é feita indiretamente na ferramenta através da definição dos valores de atrasos de comunicação e *jitter*. Essa abordagem simplificada acaba proporcionando à ferramenta a possibilidade de utilização para análise de NCS com qualquer tipo de rede.

Na ferramenta *Jitterbug*, todas as definições dos modelos requeridos (modelo de sinalização e modelo temporal) e a utilização das funções disponíveis no pacote da ferramenta

são realizadas através de códigos ou *scripts* escritos diretamente na área de comando do *Matlab* (um exemplo desse *script* pode ser visto na Figura 18).

Dados de Saída

A utilização da ferramenta *Jitterbug* permite obter dados relacionados ao desempenho de controle e a estabilidade de um NCS sob a influência de diversos parâmetros. O principal dado de saída obtido pela ferramenta é o valor de uma função custo quadrático calculada para o NCS e relativa a outros parâmetros do sistema. A partir dos valores dessa função custo é possível analisar características de estabilidade do NCS. De acordo com Cervin et al. (2003), altos valores para a função custo em um NCS indicam que o sistema em malha fechada é menos estável (mais oscilatório) e valores muito altos (infinito) podem significar que o sistema de controle é instável. Figura 19 apresenta um gráfico que exemplifica a resposta de um sistema de controle com diferentes valores para sua função custo.

Figura 19. Desempenho de Controle de um NCS de acordo com diferentes Valores da Função Custo

A ferramenta *Jitterbug* permite a geração de gráficos relacionados à resposta em frequência do NCS. A partir da análise da magnitude da resposta dos sistemas em malha fechada, consegue-se avaliar o desempenho do sistema e realizar correlações desses valores para avaliar especificações de controle definidas no domínio do tempo.

Um dado de saída muito interessante e fornecido unicamente pela ferramenta *Jitterbug* é o gráfico de espectro da qualidade de controle de um NCS, mostrado na Figura 20. Com a

obtenção desse gráfico, consegue-se avaliar e quantificar os efeitos da alteração de mais de um parâmetro de configuração no desempenho e na estabilidade do sistema.

Figura 20 apresenta um exemplo desse gráfico, que relaciona a sensibilidade do controle de uma planta em malha fechada, em função do período de amostragem e do atraso de comunicação presente no NCS.

Figura 20. Qualidade de Controle de um NCS relacionado com os Parâmetros Atraso de Comunicação e Período de Amostragem (Cervin & Lincoln, 2006)

Usabilidade e Outras Informações

De acordo com as características analisadas, define-se a ferramenta *Jitterbug* como de fácil utilização e dificuldade mediana para aprendizado para a realização de tarefas relacionadas ao desenvolvimento de NCS. Os requisitos de conhecimentos para o uso da ferramenta estão relacionados ao *software Matlab* e ao aprendizado da biblioteca de funções disponíveis no pacote da ferramenta para construção dos *scripts* e realização das tarefas. Juntamente com o pacote de instalação da ferramenta, são fornecidos diversos exemplos de aplicação, os quais facilitam ainda mais a familiarização com a ferramenta. De acordo com as características referentes à impossibilidade de obtenção de diversas métricas de desempenho relacionadas ao NCS, conclui-se que a ferramenta *Jitterbug* possui grande potencial de

utilização como ferramenta auxiliar para complementação dos resultados de outras ferramentas.

A ferramenta possui uma arquitetura de desenvolvimento aberta, o que representa a disponibilidade de desenvolvimentos e implementações por parte de usuários no sentido de estender e adaptar suas funcionalidades (modificar e adicionar códigos fonte) de acordo com suas necessidades específicas.

Disponibilidade

A ferramenta *Jitterbug* é um ambiente de desenvolvimento *freeware*, não necessita de contato com desenvolvedores para fins de licença, registro ou pedido de download. Maiores informações e arquivos para download podem ser encontrados diretamente do endereço <http://www.control.lth.se/user/lincoln/jitterbug/>.

3.2.4. PICCSIM

3.2.4.1. DESCRIÇÃO DA FERRAMENTA

A ferramenta PiccSIM (*Platform for Integrated Communications and Control design, Simulation, Implementation and Modeling*), atualmente desenvolvida em parceria entre o Departamento de Tecnologia de Automação e Sistemas da Universidade de Tecnologia de Helsinque em Espoo e o Departamento de Redes e Comunicação da Universidade Aalto em Aalto, ambos na Finlândia, , é um ambiente de desenvolvimento voltado para a modelagem, projeto e simulação de NCS com redes de comunicação com e sem fio (*wireless* NCS). A integração de todas essas etapas de desenvolvimento na ferramenta possibilita o estudo dos

aspectos de comunicação e controle e também da interação entre eles em um NCS (NETHI et al., 2007a).

Essa ferramenta representa a integração entre a plataforma *MoCoNet* (*Monitoring and controlling laboratory process over Internet*), anteriormente desenvolvida pelo mesmo grupo, com o simulador de redes NS-2.3. A ferramenta *MoCoNet* (POHJOLA et al., 2005) foi desenvolvida com o intuito de permitir o acesso via Internet a uma plataforma de desenvolvimento de NCS. Essa plataforma, desenvolvida no ambiente *Matlab/Simulink*, possibilitava a modelagem, projeto e simulação de NCS. O simulador de redes NS-2.3 (NS-2, 2007) é um simulador de sistemas a eventos discretos amplamente utilizado e de grande aceitação no meio acadêmico. Esse simulador de redes (*software* livre) permite a implementação de modelos de redes de comunicação com ou sem fio para serem utilizados no desenvolvimento de NCS. Diante da grande difusão desse simulador de redes, uma prática comum tem sido a utilização de modelos de redes e protocolos de comunicação previamente desenvolvidos, validados e disponibilizados por outros grupos e instituições de pesquisa. Um modelo de redes CAN para o simulador NS-2 foi implementado no trabalho de Fummi et al. (2004). A integração entre as duas ferramentas citadas amplia as possibilidades de aplicação da ferramenta *PiccSIM*, permitindo a simulação e a análise de desempenho de NCS com uma grande diversidade de redes e protocolos de comunicação. No entanto, a ferramenta não disponibiliza nenhum desses modelos de redes e assim para sua utilização, acaba sendo necessária a obtenção ou implementação desses modelos por parte do usuário. Um guia para conhecimento do simulador NS-2 e desenvolvimento de modelos de rede pode ser encontrado no trabalho de Coutinho (2007).

A ferramenta *PiccSIM* possui também um ambiente gráfico para modelagem e desenvolvimento da rede de comunicação e/ou do sistema de controle. A partir do modelo desenvolvido para a rede de comunicação é gerado o arquivo contendo o *script* de

configuração necessário para o funcionamento do simulador NS-2. Os controladores desenvolvidos para o NCS no ambiente *Matlab/Simulink* são carregados na ferramenta para realização das simulações e análises. O NCS projetado pode também ser testado em *hardware* real. Para que isso aconteça, a ferramenta realiza a conversão dos modelos desenvolvidos em um código em linguagem C através do *Matlab/Real-Time Workshop* e compilam o programa obtido em equipamentos reais com o *Matlab/Target Language Compiler*.

O grande diferencial da ferramenta *PiccSIM* é a possibilidade de utilização pela Internet. Uma versão *online* dessa ferramenta (*PiccSIM remote interface*) está disponível no site do desenvolvedor, permitindo a realização de análises e simulações pela Internet e também de dois experimentos remotos previamente disponíveis. No processo de utilização da ferramenta pela Internet, pode ser usado um modelo fornecido (*template* do *Matlab* – *arquivo.mdl*) a partir do qual o usuário pode projetar o controlador a ser utilizado. É necessária também a escolha do tipo de rede de comunicação a ser utilizado, juntamente com o envio (*upload*) do arquivo com o modelo da rede ou protocolo no simulador NS-2 (*script* de configuração TCL). Esses modelos de simulação são carregados via uma interface em *Java* e os resultados são mostrados em telas de formato *scope* do *Matlab*.

Para utilização *online* da ferramenta *PiccSIM* é necessário um computador com o *software Matlab* e a instalação de um pacote (*PiccSIM Toolchain*) fornecido pelos desenvolvedores. Para maiores instruções de como utilizar essa interface remota da ferramenta, recomenda-se a leitura do Capítulo 5 do manual da ferramenta (BJÖRKBOM, NETHI & KOHTAMÄKI, 2010). Essa possibilidade de utilização *online* da ferramenta representa uma vantagem, pois acaba com uma eventual indisponibilidade de instalação e utilização da ferramenta devido a algum requisito ou conhecimento prévio de *software*.

3.2.4.2. ARQUITETURA DE DESENVOLVIMENTO

A ferramenta *PiccSIM* é composta de diversos componentes como interface gráfica de configuração da rede de comunicação e projeto do sistema de controle, funções implementadas no *Matlab/Simulink* para desenvolvimento de modelos e simulações, simulador de redes NS-2 e interface de geração de códigos. Esses componentes são distribuídos em dois de acordo com a arquitetura de desenvolvimento da ferramenta *PiccSIM* mostrada na Figura 21. Maiores informações sobre todos os componentes, configurações necessárias e princípios de funcionamento podem ser encontradas em Nethi et al. (2007a) e Björkbom, Nethi & Kohtamäki (2010).

Figura 21. Arquitetura da Ferramenta *PiccSIM* (Björkbom, Nethi & Kohtamäki, 2010)

A ferramenta possui uma interface gráfica principal (*toolchain*), mostrada na Figura 21, desenvolvida em linguagem *Java* que permite ao usuário a criação e desenvolvimento de um modelo referente à rede de comunicação que será utilizada no NCS. A utilização desta interface se torna necessária quando o usuário da ferramenta não possui um *script* de configuração (TCL) a ser utilizado no simulador de redes NS-2. Esse modelo desenvolvido é então utilizado pela ferramenta para a geração automática do *script* de configuração necessário.

O desenvolvimento do sistema de controle é realizado através de modelos do *Simulink*. A ferramenta *PiccSIM* possui uma biblioteca de blocos, mostrada na Figura 22, referente aos componentes de um NCS como processo, controlador e rede de comunicação com e sem fio.

Figura 22. Biblioteca de Blocos da Ferramenta PiccSIM (Björkbohm, Nethi & Kohtamäki, 2010)

Essa biblioteca contém blocos para transmissão de mensagens através de redes de comunicação simuladas e um bloco que implementa um mecanismo de sincronização de tempo, o qual deve ser utilizado em todos os modelos desenvolvidos na ferramenta. Esses blocos possuem uma interface de configuração na qual parâmetros e propriedades podem ser

definidos. Por exemplo, parâmetros P, I e D podem ser selecionados quando um controlador PID é definido e funções de transferência e valores de atrasos de comunicação são necessários quando são configurados os blocos de processo.

Na operação da ferramenta *PiccSIM*, a simulação do NCS ocorre com o funcionamento simultâneo entre o ambiente de execução em tempo real do *Matlab/Simulink* e o simulador de redes NS-2. Ao final das simulações realizadas, todos os desenvolvimentos realizados podem ser convertidos em códigos em linguagem C para serem utilizados em equipamentos reais (necessitando porém dos *toolboxes* adicionais do *Matlab: Real-Time Workshop Embedded Coder* e *Target Language Compiler*).

3.2.4.3. CRITÉRIOS DE AVALIAÇÃO DA FERRAMENTA

Áreas de Aplicação

A ferramenta *PiccSIM* pode ser utilizada para modelagem, projeto, simulação e implementação de NCS. Essa ferramenta possui uma grande flexibilidade de aplicação, obtida com a integração entre as ferramentas de projeto do *Matlab/Simulink* com o simulador de redes NS-2.

A principal aplicação da ferramenta *PiccSIM* é para a obtenção de métricas de desempenho relacionadas ao controle do NCS. A ferramenta permite o cálculo de diversos parâmetros relacionados com a resposta do sistema a uma entrada degrau, como tempo de acomodação, máximo de pico, tempo de subida e erro em regime permanente. A ferramenta possui uma interface gráfica para a sintonia de parâmetros de controladores que pode auxiliar e facilitar as tarefas de desenvolvimento de estratégias de controle para o NCS projetado.

Além das áreas citadas, a ferramenta *PiccSIM* também pode ser utilizada em aplicações de ensino a distância e realização de experimentos remotos sobre a tecnologia de NCS. Essa possibilidade de aplicação representa uma característica exclusiva dessa ferramenta.

Etapas de Projeto Suportadas

O ambiente de desenvolvimento da ferramenta *PiccSIM* permite ao usuário a realização da modelagem da planta ou processo a ser estudado, através de sua interface com o *Matlab/Simulink* e da utilização da biblioteca de blocos da ferramenta. A definição do tipo de rede de comunicação a ser utilizado é realizada através da seleção do arquivo de *script* de configuração (TCL) da rede no simulador de redes NS-2.

Diante da grande difusão desse simulador de redes, uma prática comum tem sido a utilização de modelos de redes e protocolos de comunicação previamente desenvolvidos, validados e disponibilizados por outros grupos e instituições de pesquisa (<http://www.isi.edu/nsnam/ns/ns-research.html>). Caso o usuário não possua nenhum modelo de rede ou *script* já implementado, existe a possibilidade de utilização de uma interface gráfica para desenvolvimento do modelo de rede a ser utilizado e posterior geração automática do *script* de configuração baseado nesse modelo.

No entanto esse modelo de rede utilizado para as simulações não oferece muita flexibilidade no sentido de se alterar alguns parâmetros de configuração de redes de comunicação como velocidade de transmissão de dados, tamanhos das mensagens.

Em relação à etapa de projeto do controlador do NCS, a ferramenta fornece um modelo (*template* do *Matlab* – *arquivo.mdl*) a partir do qual o usuário pode projetar o seu controlador e utilizá-lo nas simulações e análises realizadas na ferramenta.

Vantagens e Desvantagens

A grande vantagem da ferramenta *PiccSIM* é sua disponibilidade para utilização via Internet. Com a utilização da versão *online* da ferramenta é possível a realização de análises e

simulações de NCS projetados de acordo com as necessidades do usuário e também de dois experimentos remotos previamente disponíveis (circuito RC e lâmpadas). A utilização *online* acaba com uma eventual impossibilidade de utilização da ferramenta devido a algum requisito ou conhecimento prévio de *software* ou indisponibilidade de recursos de *hardware* para instalação e operação da ferramenta. Entre as ferramentas de suporte avaliadas neste trabalho, somente a ferramenta *PiccSIM* disponibiliza essa funcionalidade.

Além da vantagem citada, a integração da ferramenta com o simulador de redes NS-2 amplia suas possibilidades de aplicação, permitindo a simulação e a análise de desempenho de NCS com diversidade de redes e protocolos de comunicação com e sem fio.

A ferramenta *PiccSIM* apresenta algumas desvantagens e limitações no sentido de não permitir a obtenção de métricas de desempenho do NCS relacionados ao sistema de tempo real e à rede de comunicação como tempo de resposta, taxa de utilização e atrasos de comunicação na rede. De acordo com a arquitetura da ferramenta, o complexo processo de instalação (vários *softwares* e ferramentas) e a necessidade de equipamentos (dois computadores) para operação são fatores que podem dificultar a utilização da ferramenta.

Dados de Entrada

Os dados de entrada requeridos para o funcionamento da ferramenta *PiccSIM* são definidos de acordo com as duas possibilidades de utilização da ferramenta. Quando a ferramenta é instalada e utilizada por um usuário, todas as entradas de dados são realizadas diretamente no ambiente de desenvolvimento do *Matlab/Simulink* e no ambiente da interface gráfica da ferramenta (*toolchain*).

O desenvolvimento do sistema de controle é realizado através de modelos do *Simulink*, como mostrado na Figura 23(a). A ferramenta *PiccSIM* possui uma biblioteca de blocos referente aos componentes de um NCS como processo, controlador e rede de comunicação

com e sem fio. Esses blocos possuem uma interface de configuração, mostrada na Figura 23(b), na qual seus parâmetros e propriedades podem ser definidos.

Em relação ao modelo da planta a ser estudado, é necessária a definição, através do bloco processo, de sua função de transferência com atraso de comunicação (*delay*). Da mesma forma, o bloco controlador é utilizado para o desenvolvimento do controlador do NCS e de seus parâmetros como, por exemplo, parâmetros P, I e D, período de amostragem e regra de sintonia do controlador PID.

Figura 23. Modelo de um NCS na Ferramenta PiccSIM: (a) Blocos do *Simulink* disponíveis para Modelagem e (b) Interface de Configuração do Bloco de Modelagem Controlador (Nethi et al., 2007b)

A rede de comunicação a ser utilizada e todos os seus parâmetros de configuração são especificados através de um modelo ou *script* de configuração (TCL) da rede no simulador NS-2. Caso o usuário da ferramenta *PiccSIM* não possua um modelo para ser utilizado, a ferramenta possui uma interface gráfica, mostrada na Figura 24, para seu desenvolvimento.

Nessa interface gráfica, mostrada na Figura 24(a), o usuário pode definir o tipo de rede (com ou sem fio) e seus parâmetros de configuração como protocolos MAC, quantidade de dispositivos na rede, posição e padrão de movimentação de nós para redes *wireless* e modelos de propagação de sinais, entre outros. Essa interface gráfica também possibilita, a partir do modelo desenvolvido pelo usuário, a geração do código referente ao *script* de configuração necessário para funcionamento da ferramenta *PiccSIM*, mostrado na Figura 24(b).

(a)

(b)

Figura 24. Interface Gráfica para Desenvolvimento e Configuração da Rede de Comunicação na Ferramenta PiccSIM: (a) Exemplo de Interface de Configuração de Parâmetros de uma Rede sem Fio e (b) Interface de Geração do Script de Configuração (Nethi et al., 2007b)

Para a utilização da ferramenta via Internet, existem duas possibilidades. A primeira com a utilização da interface padrão mostarda na Figura 25(a).

(a)

(b)

Figura 25. Utilização da Ferramenta PiccSIM pela Internet: (a) Interface Padrão para Simulação de NCS e (b) Interface para Realização dos Experimentos Remotos de NCS ((Nethi et al., 2007a)

Nessa utilização pode ser usado um modelo fornecido (*template* do *Matlab* – *arquivo.mdl*) a partir do qual o usuário pode projetar o controlador a ser utilizado. É necessária também a escolha do tipo de rede de comunicação a ser utilizado, juntamente com

o envio (*upload*) do arquivo com o modelo da rede ou protocolo no simulador NS-2 (*script* de configuração TCL).

A segunda utilização refere-se à realização de dois experimentos remotos sobre NCS previamente disponíveis (experimento com lâmpadas e com circuito RC). Nesses experimentos, mostrado na Figura 25(b), o usuário utiliza toda a estrutura da ferramenta *PiccSIM* e consegue definir diversos parâmetros (modelo do controlador, tipo de rede, sinal de entrada, tempo de simulação, período de amostragem) e simular e analisar o NCS.

Dados de Saída

A ferramenta *PiccSIM* permite a obtenção de resultados relacionados principalmente com o desempenho e com a estabilidade do sistema de controle. Em relação à planta e ao controlador modelados para o NCS, dados de saída podem ser analisados a partir de gráficos gerados com blocos do *Simulink* (*scope*). Assim, todas as funcionalidades da utilização do ambiente *Matlab/Simulink* para obtenção de resultados pode ser utilizada. Este fato garante boa flexibilidade para a ferramenta *PiccSIM* para a obtenção e análise de resultados.

Na utilização da ferramenta via Internet, um gráfico de resposta do NCS a uma entrada degrau é fornecida ao usuário. A partir deste gráfico, mostrado na Figura 26, o usuário consegue obter informações a respeito do desempenho e estabilidade do NCS estudado.

Figura 26. Exemplo de Resultado de Simulação de um NCS na Ferramenta PiccSIM (Nethi et al., 2007a)

Usabilidade e Outras Informações

O levantamento de características realizado permite definir um nível de dificuldade mediano para utilização (principalmente por causa dos requisitos para instalação) e fácil para o aprendizado da ferramenta *PiccSIM* para a realização de tarefas relacionadas ao desenvolvimento de NCS. Os requisitos de conhecimentos para o uso da ferramenta estão relacionados ao *software Matlab/Simulink* e a necessidade de aprendizado dos blocos do *Simulink* e suas interfaces gráficas de configuração. A disponibilidade *online* da ferramenta representa um grande diferencial, pois possibilita sua utilização sem qualquer requisito ou conhecimento prévio de *software* e expande sua utilização para a área de aplicação de ensino a distância e realização de experimentos remotos sobre a tecnologia de NCS.

A ferramenta possui uma arquitetura de desenvolvimento fechada, não permitindo qualquer tipo de desenvolvimento de novas funcionalidades, alterações de código ou adaptações por parte de usuários.

Disponibilidade

A ferramenta *PiccSIM* é um ambiente de desenvolvimento *freeware*, fornecido de acordo com os termos de licença GNU (*General Public License version 3*). Para obtenção da ferramenta via download é necessário contato com os desenvolvedores. Maiores informações sobre a ferramenta e desenvolvedores podem ser encontrados diretamente do endereço **<http://wsn.tkk.fi/en/software/PiccSIM>**. Para utilização da ferramenta *PiccSIM* via Internet, acessar a seção de conexão remota (*PiccSIM remote interface*) no mesmo *website*.

3.2.5. NCS SIMU

3.2.5.1. DESCRIÇÃO DA FERRAMENTA

A ferramenta *NCS_Simu* (*Networked Control System Simulation package*), desenvolvida pelo Departamento de Engenharia e Projetos da Universidade de Sussex em Brighton na Inglaterra, é um pacote de modelagem e simulação de NCS composto por uma biblioteca de blocos desenvolvida para o ambiente *Simulink*. A biblioteca da ferramenta é composta basicamente por três blocos, sendo o bloco Subsystem1 (*Sampler*) relacionado ao processo, atuadores e sensores utilizados, o bloco Subsystem2 (*Generator*) relacionado ao controlador do sistema e o bloco Subsystem3 (*Channel*) relacionado à rede de comunicação, mostrados na Figura 27. A utilização da ferramenta permite a modelagem do NCS e a simulação da dinâmica do sistema de controle com a inclusão de uma rede de comunicação (YANG, 2007). Os blocos e funções da biblioteca da ferramenta *NCS_Simu* se conectam diretamente aos blocos já existentes nos *toolboxes* disponíveis no *Simulink*, facilitando sua utilização.

Figura 27. Blocos Básicos da Biblioteca da Ferramenta NCS_Simu

A ferramenta *NCS_Simu* proporciona ao usuário a possibilidade de analisar alguns efeitos que são característicos em NCS e acabam sendo negligenciados pela maioria das ferramentas de suporte existentes (YANG, 2006b), como os atrasos de comunicação variáveis e a taxa de perda de pacotes ou mensagens transmitidas na rede de comunicação. Em NCS, existe um atraso de comunicação (γ), mostrado na Figura 28, relacionado ao não-sincronismo de amostragem das mensagens (o atraso na amostragem de uma mensagem acaba influenciando na amostragem de mensagens subsequentes).

Figura 28. Diagrama referente ao Atraso de Comunicação devido ao Não-Sincronismo de Amostragem de Mensagens em um NCS (Yang, 2006b)

A ferramenta *NCS_Simu* permite ainda simular o NCS projetado com a definição de controladores baseados em tempo (*time-based*), isto é, que executam o algoritmo de controle projetado a cada período de tempo determinado pelo usuário. Geralmente nas ferramentas para NCS (YANG, 2006b), de forma a facilitar os trabalhos de modelagem e desenvolvimento, os controladores são definidos como baseados em eventos (*event-based*), por exemplo, onde o algoritmo de controle projetado é executado após a ocorrência do evento chegada da mensagem com os dados do processo.

Outra característica importante da ferramenta é que detalhes a respeito do protocolo de comunicação e do escalonamento de mensagens na rede não são requeridos, assim consegue-se simular NCS com qualquer tipo de protocolo ou rede de comunicação, podendo ser com e sem fio. Para isso foi assumido, na modelagem da rede de comunicação na ferramenta, que os principais fatores que influenciam no desempenho do NCS são o atraso de comunicação total e a taxa de perda de pacotes ou mensagens transmitidas. Assim, consegue-se generalizar a modelagem da rede a partir do fornecimento desses dois dados pelo usuário da ferramenta.

3.2.5.2. ARQUITETURA DE DESENVOLVIMENTO

A arquitetura de desenvolvimento da ferramenta *NCS_Simu* é baseada no diagrama da Figura 29.

Figura 29. Estrutura de um NCS na Ferramenta *NCS_Simu* (Adaptado de Yang, 2006a)

De acordo com esse diagrama, um NCS é definido a partir de três subsistemas que interagem entre si, sendo o Subsistema 1 relacionado ao processo total (atuadores, planta e sensores), o Subsistema 2 relacionado ao controlador do sistema e o Subsistema 3 relacionado à rede de comunicação.

A ferramenta *NCS_Simu* foi desenvolvida para utilização no ambiente *Matlab/Simulink*. Assim, cada um dos subsistemas citados, é composto por blocos desenvolvidos no *Simulink* e funções escritas em arquivos de programação do *Matlab* Mex. Para utilização da ferramenta, é necessário a configuração dos blocos disponíveis para definição dos modelos dos subsistemas do NCS e o desenvolvimento de arquivos de inicialização de variáveis, canais de comunicação e vetores de valores que representarão os atrasos de comunicação variáveis no NCS. Funções disponíveis na ferramenta, escritas em arquivos de programação do *Matlab* Mex, podem ser utilizadas como base para o desenvolvimento desses arquivos de

inicialização. Maiores informações sobre os subsistemas e utilização da ferramenta podem ser encontrados em seu manual de utilização (YANG, 2006a).

3.2.5.3. CRITÉRIOS DE AVALIAÇÃO DA FERRAMENTA

Áreas de Aplicação

A ferramenta *NCS_Simu* é um ambiente de desenvolvimento que visa facilitar a modelagem da interação entre um sistema de controle e uma rede de comunicação e posteriormente possibilitar a simulação dessa dinâmica de funcionamento. Esta ferramenta permite que algumas características específicas de NCS, como os atrasos de comunicação variáveis e a taxa de perda de pacotes ou mensagens transmitidas na rede de comunicação, possam ser levadas em consideração no projeto e desenvolvimento realizado.

A avaliação do sistema é basicamente baseada considerando métricas de desempenho relacionadas ao controle e a estabilidade, como tempo de acomodação, máximo de pico, tempo de subida, erro em regime e margem de ganho e estabilidade. A ferramenta possibilita também a análise de desempenho do NCS a partir cálculo de índices como, por exemplo, o critério da integral do erro quadrático (ISE), o critério da integral do erro absoluto (IAE) e o critério da integral do erro absoluto ponderado pelo tempo (ITAE).

A ferramenta *NCS_Simu* pode ser utilizada também para o projeto de controladores digitais e baseados no tempo (*time-based*) para NCS.

Etapas de Projeto Suportadas

A ferramenta *NCS_Simu* permite a realização das etapas de modelagem da planta ou processo e de desenvolvimento do controlador para um projeto de um NCS. A etapa de configuração da rede de comunicação na ferramenta é bastante simplificada. Na modelagem

da ferramenta, a rede de comunicação é caracterizada pelos principais fatores que influenciam no desempenho do NCS que são o atraso de comunicação total e a taxa de perda de pacotes ou mensagens transmitidas. Assim, consegue-se generalizar a modelagem da rede a partir do fornecimento desses dois dados pelo usuário da ferramenta.

Na ferramenta *NCS_Simu*, o modelo do processo é dado através de sua função de transferência ou equação no espaço de estados. O modelo do controlador digital utilizado na ferramenta é dado através de modelos de sistemas discretos e sua função de transferência é dada no domínio da transformada Z.

Um exemplo de um modelo de projeto de um NCS desenvolvido na ferramenta *NCS_Simu* pode ser visto na Figura 30.

Figura 30. Exemplo de Modelo de NCS na Ferramenta *NCS_Simu* (Yang, 2006a)

Vantagens e Desvantagens

A principal vantagem da ferramenta *NCS_Simu* se encontra na possibilidade de se incluir nas simulações realizadas, os efeitos causados pelo atraso de comunicação relacionado ao não-sincronismo de amostragem das mensagens. De acordo com o Yang (2006b), esse efeito é característico de NCS e não é analisado em outras ferramentas disponíveis.

A ferramenta não necessita de detalhes de características do protocolo de comunicação utilizado e nem informações sobre escalonamento de mensagens e operação da rede de comunicação. Com a generalização da modelagem da rede a partir do fornecimento de somente dois parâmetros (do atraso de comunicação total e da taxa de perda de mensagens na rede), consegue-se flexibilizar a aplicação da ferramenta para análise de NCS com qualquer tipo de rede de comunicação (com e sem fio).

Outra vantagem interessante da ferramenta é a interface direta com o ambiente *Matlab/Simulink*, muito utilizado nas áreas acadêmicas e industrial, permitindo a modelagem do processo e todas as simulações relacionadas ao projeto do sistema de controle a ser utilizado.

A ferramenta *NCS_Simu*, por ser focada no desenvolvimento do sistema de controle do NCS, apresenta como desvantagens a impossibilidade de obtenção de dados sobre métricas de desempenho da rede de comunicação como a taxa de utilização da rede e cálculo de atrasos de comunicação e métricas relacionadas ao sistema de tempo real como *jitter* e tempo de resposta do sistema. Outra desvantagem é o requisito de conhecimento, pelo projetista ou usuário da ferramenta, de características do atraso de comunicação presente na rede. Para a utilização da ferramenta *NCS_Simu*, é necessário especificar um vetor de valores referente ao atraso de comunicação variável do NCS. Como esses parâmetros são geralmente desconhecidos e difíceis de obter, muitas vezes acabam sendo escolhidos incorretamente.

Outro fato importante de se citar é que existe certa dificuldade para aprender a utilizar os blocos disponíveis da ferramenta para desenvolvimento do NCS. A utilização desses blocos demanda uma grande quantidade de adaptações e conexões com outros blocos disponíveis na biblioteca do *Simulink*.

Dados de Entrada

Os dados de entrada requeridos para o funcionamento da ferramenta *NCS_Simu* estão relacionados com as etapas de projeto do NCS suportadas pela ferramenta. Cada modelo é configurado através dos blocos da biblioteca da ferramenta e as adaptações e conexões necessárias para funcionamento, são realizadas com a utilização dos blocos disponíveis no *Simulink*. Em relação ao modelo da planta ou processo a ser estudado, é necessária a definição, através de sistemas contínuos, de sua função de transferência ou equação no espaço de estados. O desenvolvimento do controlador digital do NCS é feito através de modelos de sistemas discretos e sua função de transferência é dada no domínio da transformada Z.

Cada parâmetro de configuração do NCS é definido através das funções disponíveis na ferramenta. A configuração de características relacionadas à rede de comunicação do NCS é feita através da definição do vetor de valores de atrasos de comunicação e da taxa de perda de mensagens na rede. Com a utilização de funções do *Matlab* para criação de valores randômicos, consegue-se simular os efeitos dos atrasos de comunicação variáveis encontrados em NCS. Figura 31 apresenta um gráfico com a visualização de um exemplo de atraso de comunicação variável definido para a simulação de um NCS. A partir dessas funções da ferramenta, também é necessário a definição dos períodos de amostragem dos sensores e do período de execução do controlador do NCS.

Figura 31. Visualização de um Vetor de Valores de Atraso de Comunicação Utilizado na Ferramenta

NCS_Simu (Yang, 2006a)

Na ferramenta *NCS_Simu* todas as definições dos modelos requeridos, parâmetros e condições iniciais do sistema e a utilização das funções disponíveis no pacote da ferramenta são realizadas através de arquivos ou *scripts* de inicialização da ferramenta.

Esses arquivos podem ser escritos diretamente na área de comando do *Matlab* ou armazenados em arquivos de programação *Matlab* Mex (um exemplo de arquivo de programação pode ser visto na Figura 32). A principal função da ferramenta é a “NCS_ini”, responsável pela inicialização do canal de comunicação utilizado no NCS.

```
global delay_v psamp simfi index_v

psamp= 0.01; % Plant fixed-step T = 0.01 sec.
simfi=2.99; % Simulation ends at Tend = 2.99 sec.

load r_n % load delay data in a vector: delay_v

[sam1, cha1]=NCS_ini(1, 5, 0, 1);
% channel 1, sampler 5, skew=0
% delay data from No. 1 of delay_v (use No. 1-61)
[sam2, cha2]=NCS_ini(2, 4, 2, 70);
% channel 2, sampler 4, skew=2, 0.02 = 2 times of psamp
% delay data from No. 70 of delay_v (use No. 70-145)
```

Figura 32. Comandos da Ferramenta NCS_Simu para Definição dos Parâmetros de Simulação e Inicialização de Variáveis (Yang, 2006a)

Dados de Saída

A utilização da ferramenta *NCS_Simu* permite obter dados relacionados ao desempenho de controle e a estabilidade de um sistema de controle em malha fechada através de uma rede de comunicação. Em relação à planta e ao controlador estudados, dados de saída podem ser analisados a partir de gráficos gerados com blocos do *Simulink* (*scope*), como mostrado na Figura 33.

Figura 33. Exemplo de Simulações de um NCS para Controle de Pêndulos na Ferramenta NCS_Simu: (a) Resultados Iniciais do NCS e (b) Resultados após Projeto do Controlador do NCS (Yang, 2006a)

Estes dados de saída podem ser utilizados para análise de métricas de desempenho de controle e estabilidade de NCS. Assim, todas as funcionalidades conseguidas com a utilização do ambiente *Matlab/Simulink* para obtenção de resultados pode ser utilizada. Este fato garante boa flexibilidade para a ferramenta *NCS_Simu* para a obtenção e análise de resultados.

Usabilidade e Outras Informações

De acordo com as características analisadas, define-se a ferramenta *NCS_Simu* como de dificuldade mediana para a realização de tarefas relacionadas ao desenvolvimento de NCS. Os requisitos de conhecimentos para o uso da ferramenta estão relacionados ao *software Matlab/Simulink* e ao aprendizado da biblioteca de blocos do *Simulink* fornecidos. Para a utilização da biblioteca de blocos fornecida pela ferramenta é necessária a utilização de um grande número de blocos do *Simulink*, fato que acaba aumentando a complexidade do sistema desenvolvido. Juntamente com o pacote de instalação da ferramenta, são fornecidos alguns exemplos de aplicação, que podem auxiliar no aprendizado da ferramenta. O uso da ferramenta permite a inclusão de efeitos relacionados à rede de comunicação não utilizados em outras ferramentas.

A ferramenta possui uma arquitetura de desenvolvimento aberta, o que representa a disponibilidade de desenvolvimentos e implementações por parte de usuários no sentido de estender e adaptar suas funcionalidades (modificar e adicionar códigos fonte) de acordo com suas necessidades específicas. No entanto algumas funções principais da biblioteca da ferramenta, como as de inicialização de variáveis, não permitem a alteração de seu conteúdo.

Disponibilidade

A ferramenta *NCS_Simu* é um ambiente de desenvolvimento *freeware*, que não requer licença ou registro para funcionamento. Para obtenção da ferramenta completa via download é necessário contato com o desenvolvedor. Maiores informações sobre a ferramenta, download de uma versão “reduzida” e endereço de contato com o desenvolvedor podem ser encontrados diretamente do endereço **<http://www.sussex.ac.uk/Users/taiyang/>**.

3.2.6. TRUETIME

3.2.6.1. DESCRIÇÃO DA FERRAMENTA

A ferramenta *TrueTime*, desenvolvida pelo Departamento de Controle Automático da Universidade de Lund em Lund na Suécia, é um *toolbox* do *Matlab/Simulink* utilizado para facilitar a simulação de operação e comportamento temporal de um NCS ou um sistema distribuído com múltiplos controladores operando em tempo real (CERVIN et al., 2003). Essa ferramenta é composta basicamente por uma biblioteca do *Simulink* contendo seis blocos, sendo o bloco de *kernel* (*TrueTime Kernel*), dois blocos de rede (com fio – *TrueTime Network* e sem fio – *TrueTime Wireless Network*), dois blocos de transmissão de mensagens (envia

mensagem – *ttSendMsg* e recebe mensagem – *ttGetMsg*) e um bloco de bateria (*TrueTime Battery*), mostrados na Figura 34.

A utilização desses blocos permite simular e analisar a inclusão de uma rede de comunicação e do tráfego de mensagens (blocos de rede e blocos de transmissão de mensagens) num sistema de controle e a operação de um processador de tempo real (bloco *kernel*) executando tarefas de controle e interface de dados (I/Os). Juntamente com os blocos da biblioteca, a ferramenta possui um conjunto de funções adicionais (fornecidas em arquivos C++ ou *Matlab Mex*) desenvolvidas para configurar e definir a execução de atividades realizadas pelo bloco *kernel*. Entre essas funções pode-se citar a conversão AD/DA, envio e recebimento de mensagens, criação e gerenciamento de tarefas, interrupções, temporizadores e monitoração de eventos. Os blocos e funções da biblioteca da ferramenta *TrueTime* se conectam diretamente aos blocos já existentes nas bibliotecas disponíveis no *Simulink*, facilitando sua utilização.

Figura 34. Blocos da Biblioteca da Ferramenta *TrueTime* (Ohlin, Henriksson & Cervin, 2010)

Os blocos de rede e de *kernel* da biblioteca são baseados em eventos, podendo ser eventos internos ou externos, os quais determinam a execução de suas atividades. Eventos internos são relacionados ao tempo, como por exemplo, o tempo definido terminou, a tarefa foi executada ou a mensagem foi transmitida. Eventos externos são relacionados com interrupções ou fatos externos aos blocos como, por exemplo, uma mensagem foi detectada na rede ou a velocidade do motor ultrapassou a referência.

Um modelo da ferramenta *TrueTime* pode conter diversos blocos, como por exemplo, mais de um bloco de rede (simulando um sistema com mais de uma rede) e um mesmo bloco de *kernel* conectado em mais de uma rede. Cada rede (ou bloco de rede) no modelo é identificada por um número e cada dispositivo (ou bloco de *kernel*) conectado em uma rede é identificado por um número, o qual é único para essa rede.

Bloco de *Kernel* (*TrueTime Kernel*)

O bloco de *kernel* da ferramenta é composto por uma função do *Matlab* que simula um computador ou processador com um *kernel* baseado em eventos, operações de conversão A/D e D/A, interface de dados (I/O), comunicação via rede, canal externo para interrupções e variáveis de monitoramento (*Schedule* e *Monitors*) (CERVIN, OHLIN & HENRIKSSON, 2007). Internamente no bloco de *kernel*, estruturas como filas, memórias, buffers, temporizadores estão implementados e são utilizados nas simulações. A comunicação via rede é realizada através de uma entrada de recepção de mensagens da rede (*Rcv*), uma saída de envio de mensagens para a rede (*Snd*). O bloco de *kernel* contém ainda uma entrada (*E*) e uma saída (*P*) que são utilizadas juntamente com o bloco de bateria (*TrueTime Battery*).

A configuração do bloco de *kernel* é realizada através da definição de parâmetros em sua tela de configuração, como mostrado na Figura 35.

Figura 35. Tela de Configuração do Bloco de Kernel (Ohlin, Henriksson & Cervin, 2010)

O nome do *script* de inicialização do bloco (*name of the init function*) é o parâmetro mais importante, pois determina o código (*code*) responsável pela definição do número de entradas e saídas do bloco, utilização de uma política de escalonamento de tarefas e criação das tarefas (*tasks*), interrupções (*interrupts*) e variáveis de monitoramento a serem utilizadas pela ferramenta nas simulações. Maiores informações sobre a criação de *scripts* de inicialização de blocos de *kernel* podem ser encontradas no manual de utilização da ferramenta (Ohlin, Henriksson & Cervin, 2010). A seleção do campo *battery* define a utilização do bloco de bateria e consequentemente das portas E e P citadas anteriormente.

A execução de atividades, como tarefas, interrupções e transmissão de mensagens, pelo bloco *kernel* é definida pelo usuário nos *scripts* de inicialização através de funções escritas ou códigos (*codes*) (em C++ ou em arquivos do *Matlab* Mex do tipo m). Algoritmos de controle podem também serem definidos de forma gráfica através de blocos do *Simulink* (modelo do

Matlab do tipo mdl). Figura 36 apresenta um gráfico que relaciona o grau de dificuldade de implementação da atividade com sua velocidade de execução.

Figura 36. Comparação entre as Formas de Implementação de Atividades no *TrueTime*

O bloco de *kernel* permite também a definição de estratégias de escalonamento (*scheduling*) e priorização (*priorities*) na execução de suas tarefas e atividades. A estratégia de escalonamento adotada pode ser introduzida pelo usuário através de códigos em C++ ou através da seleção de estratégias tradicionais encontradas na literatura. Entre as estratégias disponíveis na ferramenta estão, por exemplo, a *deadline-monotonic* (as tarefas com menor valor de *deadline* recebem as maiores prioridades de execução), *fixed-priority* (as tarefas recebem previamente prioridades de execução) e *earliest-deadline-first scheduling* (com o sistema em funcionamento, as tarefas a serem executadas são inseridas em uma fila e a que tiver o menor tempo para vencimento do *deadline* é executada primeiro).

Blocos de Rede (TrueTime Network e TrueTime Wireless)

Os blocos de rede da ferramenta também são baseados em eventos e simulam o mecanismo de acesso ao meio de comunicação, a transmissão das mensagens na rede e tempos de processamento relativo à transmissão das mensagens, de acordo com o tipo de rede ou protocolo escolhido (CERVIN, OHLIN & HENRIKSSON, 2007). Cada mensagem

transmitida na rede contém informações sobre o dispositivo de origem e de destino da mensagem, dados definidos pelo usuário (medição realizadas, sinais de controle) e tamanho da mensagem. Outras informações específicas sobre características de tempo real como prioridade e *deadline* das mensagens são opcionais e definidas de acordo com o protocolo escolhido.

A ferramenta suporta a simulação de redes com fio como CSMA/CD (Ethernet), CSMA/AMP (CAN), Switched Ethernet, FDMA, TDMA (TTP - *Time triggered protocol*), Token bus, FlexRay e Profinet. Desenvolvimentos (códigos implementados nos blocos de *kernel*) têm sido feitos para possibilitar a simulação de protocolos de alto nível que utilizam as redes disponíveis na ferramenta. Como por exemplo, TCP (HENRIKSSON & CERVIN, 2003) e Modbus/TCP baseados no CSMA/CD, Profibus e ControlNet baseados no Token bus e TTCAN (ALBERT, PIETSCH & VOETZ, 2005) baseados no TDMA. Rede sem fio ou *wireless* também podem ser simuladas como a IEEE 802.11 (WLAN) e IEEE 802.15.4 (ZigBee). Ohlin, Henriksson & Cervin (2010) apresentam o desenvolvimento do protocolo de roteamento AODV (*Ad-hoc On-demand Distance Vector*) e De Biasi (2008) implementa o protocolo wireless HART para utilização em simulações de redes *wireless* na ferramenta *TrueTime*.

A configuração dos blocos de rede é realizada através da definição de parâmetros em suas telas de configuração, como mostrado na Figura 37. Parâmetros de configuração como a taxa de transmissão de dados (*data rate*), número de identificação da rede (*network number*), quantidade de nós ou dispositivos conectados na rede (*number of nodes*) e tamanho mínimo do quadro de mensagem (*minimum frame size*) são comuns aos blocos disponíveis. Outros parâmetros de configuração mais específicos também podem ser definidos de acordo com o protocolo de comunicação escolhido (Ohlin, Henriksson & Cervin, 2010).

Figura 37. Configuração dos Blocos de Rede (*TrueTime Network* e *TrueTime Wireless*) (Cervin et al., 2003)

Os blocos de rede podem ser utilizados de duas maneiras. A primeira determina que para cada dispositivo conectado em uma rede, um bloco de *kernel* é necessário para transmissão das mensagens (através de tarefas definidas pelo usuário). Esta forma de utilização é mais flexível e poderosa, no entanto requer um tempo maior de desenvolvimento e programação. A segunda maneira é através da utilização dos blocos de transmissão de mensagens.

Blocos de Transmissão de Mensagens (*ttSendMessage* e *ttReceiveMsg*)

Os blocos de transmissão de mensagens (envio e recebimento) são utilizados para simplificar a simulação e análise de redes de comunicação e sistemas de controle via redes. A configuração dos blocos de transmissão de mensagens é realizada através da definição de parâmetros em suas telas de configuração, como mostrado na Figura 38.

Figura 38. Tela de Configuração dos Blocos de Transmissão de Mensagens

A transmissão de dados na rede é realizada através das portas *Rcv* e *Snd* e o acesso aos dados transmitidos através da porta *Data*, disponíveis nesses blocos. A principal característica desses blocos é a não necessidade de utilização de blocos de *kernel* para transmissão de mensagens em um sistema. Assim, a criação de *scripts* de inicialização de blocos de *kernel* não é necessária e simplifica-se a criação de códigos para transmissão de mensagens. No entanto, os blocos de transmissão de mensagens não permitem a criação de códigos mais complexos como algoritmos de controle. Num mesmo modelo do *TrueTime* podem ser utilizados blocos de rede e blocos de transmissão de mensagens simultaneamente.

Bloco de Bateria (*TrueTime Battery*)

O bloco de bateria foi desenvolvido com o intuito de permitir a simulação de sistemas de controle via redes em ambientes como robôs móveis, no qual a alimentação dos dispositivos é fornecida por baterias.

A configuração do bloco de bateria é realizada através da definição do parâmetro potencia inicial (*watt seconds*) em sua tela de configuração, como mostrado na Figura 39.

Para a utilização deste bloco é necessário primeiramente a seleção do campo *battery* no bloco de *kernel* e a conexão entre as portas *E* do bloco de bateria e do bloco de *kernel*. Após isso, todas as saídas *P* dos blocos utilizados no modelo do *TrueTime* (que representarão os dispositivos que consumirão energia) são conectadas a entrada *P* do bloco de bateria.

Figura 39. Tela de Configuração do Bloco *Battery* da Ferramenta (Cervin et al., 2003)

3.2.6.2. ARQUITETURA DE DESENVOLVIMENTO

A ferramenta *TrueTime* foi desenvolvida para ser utilizada juntamente com o *software Matlab/Simulink*. Toda a ferramenta é desenvolvida em C++, desde os blocos do *Simulink* até o conjunto de funções primitivas e comandos internos. Os objetos da ferramenta como tarefas, variáveis, temporizadores, interrupções e eventos são definidos como classes em C++. Assim a ferramenta possui uma arquitetura de desenvolvimento aberta, que permite que essas classes e funções primitivas possam ser modificadas e adaptadas pelos usuários da ferramenta. Tanto para as classes definidas quanto para as funções primitivas da ferramenta, existe uma sintaxe específica que possibilita suas utilizações através de códigos escritos em linguagem C++ ou em arquivos de programação do *Matlab Mex*.

Para a utilização da ferramenta, de acordo com a Figura 40, é necessário a criação de um *script* de inicialização formado por códigos, definidos pelo usuário, que utilizam as classes e funções primitivas da ferramenta. Uma tabela com a descrição e explicação de todas

as funções primitivas do *kernel* da ferramenta *TrueTime* está disponível na seção 14 do seu manual de utilização (Ohlin, Henriksson & Cervin, 2010).

Figura 40. Arquitetura de Desenvolvimento da Ferramenta *TrueTime*

3.2.6.3. CRITÉRIOS DE AVALIAÇÃO DA FERRAMENTA

Áreas de Aplicação

A ferramenta *TrueTime* fornece suporte às tarefas de análise, simulação e controle de sistemas de controle via redes. A aplicação da ferramenta permite analisar a influência de diversos fatores no desempenho de um NCS ou de um sistema de controle distribuído em tempo real. Entre esses fatores estão a inserção de uma malha de controle fechado através de uma rede de comunicação no sistema, a presença de atrasos de comunicação na rede, a amostragem dos sinais, os tempos de processamento e execução de tarefas nos dispositivos e o escalonamento de atividades (tarefas e mensagens) nos processadores.

A principal utilização dessa ferramenta diz respeito ao desenvolvimento do NCS considerando métricas de desempenho relacionadas ao controle e a estabilidade do sistema, como tempo de acomodação, máximo de pico, tempo de subida, erro em regime e margem de estabilidade e de ganho. A simulação de um NCS na ferramenta pode ser realizada variando-se diversos parâmetros relacionados à NCS, resultando em correlações destes parâmetros com alterações de desempenho do sistema de controle. Um fato importante desta ferramenta está na possibilidade de se simular e dimensionar parâmetros relacionados à configuração de redes de comunicação como tipo de rede ou protocolo utilizado, período de amostragem de mensagens, tamanho de mensagens e quantidade de dispositivos na rede. De acordo com a literatura (LIAN, MOYNE & TILBURY, 2002), esses parâmetros específicos exercem forte influência no desempenho de um NCS.

A ferramenta *TrueTime* pode ser utilizada também para o projeto de controladores (baseados em técnicas convencionais como PID, adaptativo e na utilização de inteligência artificial como Fuzzy e redes neurais) e para o desenvolvimento de estratégias de controle para NCS.

Etapas de Projeto Suportadas

Uma característica única da ferramenta *TrueTime*, que lhe garante grande flexibilidade de aplicação, é sua possibilidade de utilização em todas as etapas de projeto de um NCS. O ambiente de desenvolvimento da ferramenta permite ao usuário a realização da modelagem da planta ou processo a ser estudado, através de sua interface direta com os blocos do *Simulink*. A definição do tipo de rede a ser utilizado e de seus parâmetros de configuração pode ser feita diretamente pelos blocos de rede disponíveis na ferramenta. O projeto do controlador pode ser realizado nos blocos de *kernel* da ferramenta, podendo ser realizado na forma de funções escritas ou na forma de modelos compostos por blocos do *Simulink*.

Um exemplo de um modelo de projeto de um NCS pode ser visto na Figura 41. Na Figura 41, os blocos de *kernel* (*actuator*, *interference*, *sensor*, *controller*) estão representados pelos blocos na cor cinza escuro, o bloco de rede (*network*) pela cor cinza claro e o modelo matemático do processo (*DC Servo*) na cor branca.

Figura 41. Modelo de Projeto de um NCS na Ferramenta TrueTime

Vantagens e Desvantagens

A grande vantagem de utilização da ferramenta *TrueTime* para o desenvolvimento de NCS se encontra no fato citado anteriormente de permitir o desenvolvimento de todas as etapas do projeto de um NCS (definição do modelo da planta, do controlador e do tipo de rede utilizado). Entre diversas ferramentas analisadas, essa funcionalidade está presente somente na ferramenta *TrueTime*.

Ainda em relação à ferramenta, outras vantagens podem ser citadas como a disponibilidade de análise de diversos tipos de redes e a possibilidade de implementação, por parte de usuários, de protocolos de alto nível para os tipos de redes disponíveis na ferramenta, aumentando sua flexibilidade de utilização em simulações e projetos. Outra vantagem

interessante da ferramenta é permitir a definição ou especificação dos tempos de processamento e execução de tarefas utilizados pelos dispositivos. Esses tempos são componentes do atraso de comunicação nas redes, os quais exercem forte influência no desempenho de NCS (TIPSUWAN & CHOW, 2003).

A ferramenta *TrueTime* apresenta como desvantagens a impossibilidade de obtenção de dados sobre métricas de desempenho relacionadas a redes de comunicação como a taxa de utilização da rede e atrasos de comunicação e a dificuldade de obtenção de informações sobre requisitos de tempo real de mensagens e tarefas como tempo de transmissão, tempo de execução e cumprimento de *deadlines*. Essa dificuldade citada se encontra no fato da necessidade de implementação de códigos de armazenamento de informações (*log*) de cada tarefa executada para obtenção dos dados requeridos.

Dados de Entrada

Os dados de entrada requeridos para o funcionamento da ferramenta *TrueTime* estão relacionados com as etapas de projeto do NCS. Em relação ao modelo da planta a ser estudado, é necessária a definição, através de blocos do *Simulink*, de sua função de transferência. A rede de comunicação a ser utilizada e todos os seus parâmetros de configuração são especificados através dos blocos de rede disponíveis. Como citado anteriormente, dados de entrada como a taxa de transmissão de dados, número de identificação da rede, quantidade de nós ou dispositivos conectados na rede e tamanho mínimo do quadro de mensagem são comuns aos blocos disponíveis, além de outros específicos de acordo com o protocolo de comunicação escolhido.

O desenvolvimento de controladores na ferramenta é realizado nos blocos de *kernel*. Assim, para cada bloco de *kernel* utilizado, torna-se necessário a criação do seu *script* de inicialização. O *script* de inicialização é o código responsável pela especificação do número de entradas e saídas do bloco, utilização de uma política de escalonamento de tarefas e criação

das interrupções, variáveis de monitoramento e tarefas que serão realizadas pela ferramenta. Exemplos de uma dessas tarefas são os algoritmos de controle ou controladores criados pelo usuário de acordo com as necessidades de operação do NCS.

Dados de Saída

A ferramenta *TrueTime* oferece uma grande quantidade de dados de saída que podem ser obtidos das etapas de projeto do NCS. Em relação à planta e ao controlador estudados, dados de saída podem ser analisados a partir de gráficos gerados com blocos do *Simulink* (*scope*), como mostrado na Figura 42.

Figura 42. Gráficos do Desempenho de Controle (acima) e do Sinal de Saída do Controlador (abaixo) da Ferramenta *TrueTime* (Cervin et al., 2003)

Em relação ao processador de tempo real dos dispositivos conectados na rede e a transmissão de mensagens na rede, gráficos disponíveis através das portas de saída *Schedule* de cada bloco de *kernel* e do bloco rede são gerados automaticamente, como mostrado na Figura 43. Para cada processador (ou bloco de *kernel*) são fornecidas informações sobre a realização de tarefas e interrupções durante a simulação do NCS.

Para a rede de comunicação utilizada são fornecidas informações sobre a transmissão das mensagens durante a simulação realizada. Para os gráficos de monitoramento, Figura 43, nível alto do sinal significa que uma tarefa esta sendo executada pelo processador e que uma mensagem esta sendo transmitida pela rede. Nível médio do sinal significa a preempção de uma tarefa (tarefa está pronta para ser executada, mas está esperando a disponibilidade do processador), e o tempo de espera de uma mensagem ser transmitida na rede (mensagem está pronta para ser transmitida, mas está esperando o término da transmissão de outra mensagem). Nível baixo do sinal significa que tanto o processador quanto a rede de comunicação estão ociosos naquele instante de tempo.

Figura 43. Gráficos de Monitoramento de Utilização de Recursos da Rede (acima) e do Processador (abaixo) da Ferramenta *TrueTime* (Cervin et al., 2003)

Para uma análise e entendimento dos gráficos de saída da Figura 43, todos os dispositivos que estiverem conectados e utilizando recursos da rede de comunicação serão mostrados na área de monitoramento da rede. O gráfico de monitoramento do processador

relaciona todas as diferentes tarefas que utilizam recursos desse processador (ou bloco de *kernel*) especificamente. Para cada bloco de *kernel* existente num projeto de NCS desenvolvido, um gráfico de monitoramento pode ser obtido.

A obtenção de informações sobre requisitos de tempo real de mensagens e tarefas, como tempo de transmissão e cumprimento de *deadlines*, é opcional e está atrelada a criação, por parte do usuário, de códigos para armazenamento e disponibilização dos dados em variáveis do *Matlab* (*workspace*).

Usabilidade e Outras Informações

De acordo com as características analisadas, define-se a ferramenta *TrueTime* como de fácil utilização e aprendizado para a realização de tarefas relacionadas ao desenvolvimento de NCS. Os requisitos de conhecimentos para o uso da ferramenta estão relacionados ao *software Matlab/Simulink* e ao aprendizado da biblioteca de blocos do *Simulink* fornecidos. Juntamente com o pacote de instalação da ferramenta, são fornecidos diversos exemplos de aplicação, os quais facilitam ainda mais a familiarização com a ferramenta.

O uso da ferramenta demanda um maior conhecimento do NCS que será estudado por parte do usuário, pois necessita da realização de todas as etapas do projeto (definição do modelo da planta, do controlador e das configurações do tipo de rede utilizado).

A ferramenta possui uma arquitetura de desenvolvimento aberta, o que representa a disponibilidade de desenvolvimentos e implementações por parte de usuários no sentido de estender e adaptar suas funcionalidades (modificar e adicionar códigos fonte) de acordo com suas necessidades específicas.

Kusnadi (2007) apresenta resultados iniciais dos desenvolvedores do *TrueTime* no sentido de disponibilizar a ferramenta no ambiente de desenvolvimento *freeware* (gratuito) *Scilab/Scicos* (SCILAB, 2008). A disponibilização da ferramenta *TrueTime* para o *software Modelica*, na forma de uma biblioteca de simulação de redes, é discutida e mostrada em

Reuterswärd et al. (2009). Farias et al. (2009) apresentam o desenvolvimento de uma biblioteca em *Java (Open Source)* baseada na ferramenta *TrueTime* para permitir a simulação de NCS para desenvolvedores que utilizam a linguagem *Java*.

Portanto, verifica-se a grande difusão da ferramenta *TrueTime* na forma de portabilidade para outros ambientes de desenvolvimento e linguagens de programação, fato que retira assim o requisito inicial de disponibilidade da ferramenta *Matlab/Simulink* para uso da ferramenta *TrueTime*.

Disponibilidade

A ferramenta *TrueTime* é um ambiente de desenvolvimento *freeware*, não necessita de contato com desenvolvedores para fins de licença, registro ou pedido de download. Maiores informações e arquivos para download podem ser encontrados diretamente do endereço <http://www.control.lth.se/truetime>.

3.3. SISTEMATIZAÇÃO E COMPARAÇÃO DE FERRAMENTAS ANALISADAS

Tabela 2 sintetiza as informações obtidas e apresenta uma comparação das ferramentas analisadas considerando-se um levantamento sobre características, requisitos e conhecimentos de *software*, vantagens e usabilidade.

Tabela 2. Síntese da Avaliação das Ferramentas de Análise, Simulação e Projeto de NCS

Ferramenta	Aplicação	Usabilidade (Fácil, Médio e Difícil)	Vantagens	Desvantagens	Requisitos de Conhecimentos
AIDA	Análise de métricas de desempenho de sistemas de tempo real e de controle	(Difícil) - Facilidade na parte relacionada ao Matlab Dificuldade no fluxograma do sistema	Desempenho do sistema baseado no cálculo dos tempos de transmissão e atrasos de comunicação	Somente suporte para aplicações baseadas no CAN ou com escalonamento de mensagens	Conhecimentos de Matlab Informações sobre tempos de execução das tarefas
TORSCHÉ	Análise de métricas de desempenho de sistemas de tempo real	(Médio) - Aprendizado da sintaxe das funções relacionadas à entrada de dados do sistema	Análise e otimização da rede (cumprimento de <i>deadlines</i> , diminuição de atrasos) Interface gráfica de resultados	Não fornece dados sobre controle ou resposta do sistema Necessita de dados de entrada difíceis de obter (latência)	Conhecimentos de Matlab Informações sobre sistemas de tempo real para escolha dos algoritmos de escalonamento
TrueTime	Análise de métricas de desempenho de controle e da rede e estabilidade do sistema	(Fácil) - Necessita aprendizado dos blocos da ferramenta Maior trabalho para projeto total do NCS	Análise de vários tipos de redes Desenvolvimento total do NCS (modelo da planta, controlador e rede utilizada)	Não fornece dados sobre utilização da rede e requisitos temporais de mensagens (tempo de transmissão, atrasos, jitter)	Conhecimentos de Matlab/Simulink Parâmetros da rede (dados sobre mensagens, velocidade da rede)
Jitterbug	Análise de métricas de desempenho de controle e estabilidade do sistema	(Médio) - Necessita aprendizado da sintaxe das funções desenvolvidas do Matlab	Utilização da teoria LQG de controle Gráfico do espectro da influência de parâmetros na qualidade do controle	Não geração de resultados no domínio do tempo como overshoot, tempo de acomodação Impossibilidade de lidar com vários NCS simultaneamente	Conhecimentos de Matlab Informações sobre parâmetro do sistema (atraso de comunicação, <i>jitter</i>)
PiccSIM	Análise de métricas de desempenho de controle	(Fácil) - Disponibilidade online e interface gráfica auxiliam e facilitam a utilização (Difícil) – Aprendizado da ferramenta NS-2	Possibilidade de utilização online Análise de vários tipos de rede (com e sem fio) devido à utilização do simulador de redes NS-2	Necessidade do modelo da rede para o simulador NS-2 Somente resultados sobre o controle do NCS Requisitos de instalação	Conhecimentos de Matlab/Simulink Informações sobre a parâmetros da rede de comunicação para configuração na ferramenta
NCS_Simu	Análise de métricas de desempenho de controle e estabilidade do sistema	(Médio) - Necessita aprendizado dos blocos relacionados à rede e alguns conceitos apresentados pelo desenvolvedor	Inclusão de efeitos relacionados à rede de comunicação, não utilizados em outras ferramentas	Necessidade de utilização de grande número de blocos (aumenta complexidade do sistema)	Conhecimentos de Matlab Informações sobre atrasos de comunicação total e taxa de perdas de mensagens

Neste trabalho foram analisadas ferramentas de suporte a NCS baseadas principalmente na plataforma *Matlab/Simulink* e que atendessem a um requisito inicial de permitir, no mínimo, a análise de NCS com redes CAN. É importante citar que existem ainda várias outras ferramentas de suporte a NCS que não foram analisadas, como a ferramenta *XILO* (El Khoury & Tornngren. 2001), a ferramenta *SynDEx* (Sorel, 2005), a ferramenta *RTSIM* (Palopoli et al., 2002) e as ferramentas desenvolvidas nos trabalhos Casanova et al. (2006) e Pinnoti Jr & Brandão (2005), além das ferramentas citadas na revisão bibliográfica.

A escolha de uma ferramenta para utilização no projeto e desenvolvimento de um NCS deve-se basear nas definições de seu projetista. Tais definições têm sido baseadas em relação ao tipo de rede e protocolo de comunicação em que o NCS será baseado e aos parâmetros mais interessantes a serem analisados. Ressalta-se que essa escolha de parâmetros depende das necessidades de conhecimento da aplicação e do tipo de métrica de desempenho (de controle, de rede ou de tempo real) requeridos pelo projetista.

Entre as ferramentas avaliadas, diante da grande difusão e utilização no meio acadêmico internacional (GAID et al., 2006; ZHAO & XIA, 2006B; CERVIN, OHLIN & HENRIKSSON, 2007) e por causa de características que facilitam o desenvolvimento e avaliação de NCS sob diferentes configurações, a ferramenta *TrueTime* apresenta-se como a melhor avaliada e indicada para pesquisadores e desenvolvedores na área de NCS. A ferramenta *Jitterbug* também se mostra vantajosa quando da necessidade de análise da qualidade de controle do NCS quanto à alteração em seus parâmetros. Portanto essas duas ferramentas serão escolhidas para a realização das análises e simulações relacionadas às propostas de projeto e desenvolvimento de NCS com redes CAN deste trabalho.

É importante citar que dificilmente se encontrará uma ferramenta de suporte ao desenvolvimento de NCS que seja completa, permitindo a realização de todos os tipos de análises e simulações. De acordo com as informações obtidas com o levantamento realizado,

o ideal seria utilizar mais de uma ferramenta simultaneamente no desenvolvimento do NCS, cada uma desempenhando as tarefas que fazem melhor e assim potencializando os resultados obtidos.

3.4. CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentada uma revisão geral sobre o desenvolvimento e a utilização de ferramentas de análise, simulação e projeto de NCS e buscou-se demonstrar a importância da utilização dessas ferramentas para o desenvolvimento de NCS. Um enfoque principal foi direcionado para as ferramentas que possibilitassem a pesquisa de NCS com protocolo CAN, entre outros.

Foram sistematizadas as informações pesquisadas sobre características, requisitos de *software*, vantagens e usabilidade de diversas ferramentas disponíveis na literatura. De acordo com esse levantamento, entre as ferramentas analisadas, a *TrueTime* foi a melhor avaliada e apropriada para utilização na análise, simulação e projeto de NCS. Essa conclusão foi obtida diante de características como possibilitar o desenvolvimento total (modelo da planta, projeto do controlador e definição da rede), ser de fácil utilização e apresentar grande respaldo e aplicação no meio acadêmico. No entanto é importante reiterar que os resultados obtidos em um eventual projeto e desenvolvimento de um NCS seriam potencializados pela utilização conjunta de duas ou mais ferramentas.

A sistematização das informações obtidas neste capítulo permitiu criar uma documentação de referência e uma tabela de informações sintetizadas para orientar trabalhos de pesquisa e desenvolvimento sob o tema de NCS e auxiliar no conhecimento de ferramentas de desenvolvimento existentes.

4. PLATAFORMA DE PESQUISA SOBRE NCS

4.1. CONSIDERAÇÕES INICIAIS

A revisão bibliográfica realizada mostrou poucos trabalhos focados no desenvolvimento e aplicação prática de NCS. A maioria dos trabalhos apresentavam resultados validados somente através de dados teóricos obtidos por simulações. Visando suprir essa carência de resultados fundamentados em aplicações reais, este capítulo apresenta o desenvolvimento de uma plataforma de pesquisa de NCS. O objetivo dessa plataforma é a investigação experimental em NCS com redes CAN, permitindo a realização de análises de desempenho de parâmetros, modelagem e testes de controle de NCS. Uma descrição detalhada da plataforma de pesquisa é apresentada, enfatizando as arquiteturas de *hardware* e *software* desenvolvidas.

4.2. ESTRUTURA DA PLATAFORMA DE PESQUISA DE NCS COM REDE CAN

De acordo com Pinotti Jr & Brandão (2005), o crescente aumento do uso de sistemas de controle distribuídos baseados em arquiteturas em barramento em aplicações industriais tem demandado a inclusão de tecnologias de redes industriais ou *fieldbus* no ensino acadêmico e o desenvolvimento de laboratórios e plataformas de pesquisa e didáticas. Diversos trabalhos e instituições de ensino têm desenvolvidos essas plataformas para pesquisa e ensino de tecnologias *fieldbus* (MARTÍN & TADEO, 2006; KOLLA, 2007). No entanto, para NCS, que representa uma abordagem mais recente e diferente para a aplicação de sistemas de controle

com *fieldbus*, essas plataformas de ensino e pesquisa não são encontradas com tanta frequência.

A plataforma de pesquisa de NCS definida neste trabalho é constituída de uma bancada de testes de NCS que representaria um sistema de manufatura de pequeno porte. Este sistema é composto por diferentes malhas de controle fechadas, geralmente utilizadas na área industrial, integradas através de uma rede CAN como mostrada na Figura 44.

Na Figura 44 podem ser visualizadas as malhas de controle construídas (controle de temperatura, de posição de motor DC, de uma esteira de movimentação, de velocidade de motor DC e de nível de reservatório) para a plataforma de NCS. Dois computadores (*desktops*) com o *software LabVIEW* e interfaces PCI-CAN da *National Instruments* são utilizadas para desenvolvimento das estratégias de controle para os NCS e um notebook com *software LabVIEW* conectado em uma interface USB-CAN da *National Instruments* é utilizado como estação de registro de informações e supervisão dos processos ou malhas de controle.

Figura 44. Estrutura Definida para Plataforma de NCS com Rede CAN

A arquitetura proposta para a plataforma de NCS possui grande flexibilidade de aplicação tanto em pesquisa como em ensino. A plataforma fornece recursos para tarefas de análise, modelagem, simulação e controle de NCS, as quais serão demonstradas na seção 6. Também existe a capacidade de estudar e aplicar técnicas avançadas para o projeto de NCS como avaliações em tempo real, *hardware-in-the-loop (HIL)*, *rapid control prototyping (RPC)*. Para isso, é necessária a utilização de um sistema operacional de tempo real como, por exemplo, o *LabVIEW Real Time*.

4.2.1. REDE INDUSTRIAL – CAN (CONTROLLER AREA NETWORK)

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

Como descrito em Sousa (2002), o CAN é um protocolo de comunicação digital serial, onde a comunicação de dados é baseada em mensagens formadas por quadros de bits com determinada função. Entre esses quadros de bits, existe o campo identificador (ID - *identifier*) que caracteriza e define a prioridade de cada mensagem. O valor do identificador de uma

mensagem em uma rede CAN é exclusivo e quanto mais baixo seu valor, maior será a prioridade da mensagem.

Os sinais elétricos digitais do CAN são representados pelo nível recessivo (nível lógico 1) e nível dominante (nível lógico 0), sendo eles sinais diferenciais entre os dois fios do barramento (condutores CAN_H e CAN_L).

O mecanismo de acesso ao meio é fundamentado no conceito CSMA/NDBA - *Carrier Sense Multiple Access with Non-Destructive Bitwise Arbitration* (Acesso Múltiplo com Detecção de Portadora com Arbitragem Não Destrutiva por Operação Lógica Bit-a-Bit) que significa que os módulos (ECU CAN) possuem acesso ao barramento com prioridades determinadas.

De acordo com a Figura 45, ao verificar o estado do barramento, os módulos iniciam a transmissão de suas mensagens. De acordo com o valor do identificador, o módulo com a mensagem de prioridade menor cessa sua transmissão e o módulo com a mensagem de maior prioridade continua enviando sua mensagem deste ponto, sem ter que reiniciá-la. Isto é realizado pelo processo de arbitragem bit a bit não destrutivo, ou lógica "E" por fios, quando dois ou mais módulos iniciam a transmissão simultaneamente. Cada bit transmitido é comparado, sendo que o dominante sobrepõe o recessivo.

Figura 45 – Mecanismo de Acesso ao Meio do CAN (Adaptado de CIA, 2005)

Dentre as especificações do protocolo CAN em relação à camada de enlace de dados, estão os dados relacionados aos formatos existentes do quadro de dados. São definidos dois formatos de quadros dados de mensagem, onde a única diferença está no tamanho do identificador, sendo CAN A *Standard* (ID 11 bits) e CAN B *Extended* (ID 29 bits) especificados segundo a Figura 46.

Figura 46 – Formato dos Quadros de Mensagens CAN (Adaptado de SOUSA, 2002)

4.2.2. ARQUITETURA E IMPLEMENTAÇÃO DE HARDWARE

Para o desenvolvimento da plataforma de pesquisa de NCS com redes CAN, foi buscada uma arquitetura de *hardware* que fornecesse modularidade e versatilidade, facilitando sua implementação, simplificando sua expansão e possibilitando a realização de experimentos com diversas configurações. Além disso, outro fator importante seria o custo de implementação do NCS, com preferência por soluções simples e de baixo custo. Portanto, para cada NCS implementado na plataforma foi utilizada a estrutura de *hardware* apresentada na Figura 47.

Figura 47. Estrutura de Hardware da Plataforma de NCS com Rede CAN

De acordo com Figura 47, o processo a ser controlado (ou planta) é admitido ser contínuo no tempo e o atuador possui um segurador de ordem zero (ZOH – *zero order hold*) que armazena o último sinal de controle amostrado até a chegada do próximo valor ou até o próximo período de amostragem. A rede de comunicação do NCS é então usada para transmissão de medidas de saída da planta para o controlador, fazendo com que a planta seja medida de acordo com um período de amostragem (h). O funcionamento do controlador do NCS pode ser baseado em tempo (*time-driven*) ou baseado em eventos (*event-driven*). Assim um novo sinal de controle pode ser calculado a cada intervalo de tempo com um período de amostragem constante (*time-driven*) ou um novo sinal de controle é calculado imediatamente após o recebimento, através de uma mensagem na rede, de uma nova medida da saída da planta pelo sensor (*event-driven*). Atualmente, os controladores dos NCS da plataforma operam baseados em eventos.

A unidade de controle ou ECU (SOUSA, 2002), mostrada na Figura 47, baseada em um microcontrolador PIC18F258 é responsável pela aquisição de dados (sensor), atuação na planta (atuador) e pela comunicação com a rede CAN. Assim, na estrutura utilizada, o sensor

e o atuador estão alocados fisicamente em conjunto com a planta. O diagrama do circuito desta interface CAN, mostrado na Figura 48, é composto basicamente por três módulos integrados, que são:

- ✓ Transceptor CAN: módulo responsável pela adaptação dos níveis de tensão entre os circuitos da interface CAN e do barramento CAN;
- ✓ Transceptor RS232: módulo responsável pela adaptação dos níveis de tensão entre circuito do nó CAN e a interface RS232, baseada em um controlador tipo USART (*Universal Synchronous / Asynchronous Receiver Transmitter*);
- ✓ Microcontrolador com Controlador CAN: módulo constituído por CPU, memória, programas computacionais, interfaces para outros dispositivos e controlador CAN, que é o módulo central da implementação e do controle do protocolo utilizado para comunicação.

Figura 48. Diagrama do Circuito da Interface CAN ou ECU (Sousa, 2002)

Entre as especificações e características dessa interface, mostradas de acordo com as legendas da imagem da interface da Figura 49, podem-se citar:

- ✓ Microcontrolador PIC18F258 da Microchip com controlador CAN – Legenda 4;

- ✓ Transceiver MCP2551 da Microchip – Legenda 2;
- ✓ Circuito integrado de condicionamento da porta serial MAX232 – Legenda 3;
- ✓ Barramento CAN a 4 fios (GND, VCC, CAN_H, CAN_L) – Legenda 6;
- ✓ Conexão de dispositivos através de interface Serial RS232 – Legenda 1;
- ✓ Conexão de sensores e atuadores através de portas de E/S (entrada e saída) – Legenda 5.

Figura 49. Imagem da Interface CAN Utilizada (Sousa, 2002)

O controlador do NCS, de acordo com a Figura 47, se encontra fisicamente separado da planta e é implementado utilizando de computadores (*desktop*). Esta configuração facilita o desenvolvimento e a implementação de algoritmos de controle, acelerando o tempo de projeto e testes de estratégias de controle para NCS. Esses controladores utilizam uma placa com interface PCI para comunicação com a rede CAN.

A estrutura de *hardware* definida para a plataforma fornece os requisitos de modularidade (cada NCS é implementado separadamente e pode ser facilmente conectado na plataforma) e versatilidade (vários experimentos e configurações para os NCS podem ser definidos) almejados. Outra grande vantagem da estrutura diz respeito ao seu baixo custo para

o desenvolvimento de novos NCS e/ou expansão da plataforma, a partir do momento que somente são necessárias a definição da planta do NCS e o desenvolvimento dos programas do sensor e atuador que seriam embarcados na ECU CAN. O desenvolvimento do controlador do NCS poderia ser realizado no mesmo microcomputador já utilizado (*desktop*), fato que representaria a economia da parte de maior investimento necessária (por causa do alto custo relativo a interface PCI CAN para microcomputadores).

4.2.3. ARQUITETURA E IMPLEMENTAÇÃO DE SOFTWARE

O desenvolvimento de *software* para a plataforma de NCS com redes CAN requereu uma arquitetura bastante flexível e expansível para facilitar o desenvolvimento de código e englobar as características definidas pela arquitetura de *hardware* (como por exemplo, a possibilidade de utilização de um microcomputador como controlador de mais de um NCS simultaneamente).

Em cada um dos NCS implementados na plataforma, o sensor periodicamente realiza a medição da planta (período de amostragem – h) e envia a informação através de mensagens na rede CAN para o controlador. Após o recebimento da mensagem, o controlador calcula o sinal de controle através da estratégia de controle projetada e envia a informação pela rede CAN para o atuador. O atuador então recebe a mensagem e atua na planta do sistema. Portanto tanto o controlador quanto o atuador operam baseado em eventos (chegada de uma nova mensagem). Todos os NCS da plataforma estão compartilhando recursos de banda da rede CAN e existe a possibilidade também de compartilhamento recursos de processamento (CPU) dos *desktops* por mais de um NCS.

O desenvolvimento do *software* embarcado nos microcontroladores das ECUs dos NCS foram desenvolvidos em linguagem C através do ambiente de desenvolvimento *Mplab* com Compilador C18 da *Microchip*.

O *software* dos controladores e de supervisão geral da plataforma de NCS, implementados nos computadores através do ambiente de desenvolvimento *LabVIEW* da *National Instruments*, utiliza um padrão de projeto baseado numa estrutura produtor/consumidor (es) com máquina de estado, mostrada na Figura 50. Essa estrutura de *software* facilita o desenvolvimento de código e fornece uma arquitetura de desenvolvimento para o controlador do NCS que pode ser facilmente expandida.

Figura 50. Estrutura de *Software* da Plataforma de NCS com Rede CAN

O conceito de produtor/consumidor foi utilizado para possibilitar um paralelismo de processamento para o caso de múltiplos controladores de NCS rodando no mesmo microcomputador (CPU). Nesse caso, o laço de repetição produtor do *software* é responsável por criar as tarefas a serem executadas nos respectivos laços de repetição consumidor, que representam cada um controladores dos NCS conectados na plataforma. As tarefas criadas pelo produtor são inseridas em uma fila e disponibilizadas ao (s) consumidor (es). No caso de mais de um consumidor, as tarefas são respectivamente retiradas da fila por cada consumidor

e processadas paralelamente. Esse processamento paralelo de controladores dos NCS representa uma grande vantagem da estrutura implementada, eliminando um conhecido problema da utilização de uma estrutura somente com máquina de estado, que seria a impossibilidade de tratar mais de um estado (no caso da plataforma: de se calcular um novo sinal de controle para mais de um NCS) simultaneamente.

O conceito de máquina de estados foi utilizado devido à natureza sequencial de operação dos NCS e para simplificar a divisão das ações a serem realizadas. A máquina de estados implementada possui 8 estados (Inicialização, SeleçãoNCS, Monitoramento, RecebimentoMSG, DeterminaçãoNCS, Controle, EnvioMSG e Parada) mostrados na Figura 50, cada um com uma finalidade específica para o funcionamento do controlador (GODOY et al., 2010d).

- ✓ 1. Inicialização: estado responsável pela configuração e inicialização da rede CAN e dos dispositivos conectados;
- ✓ 2. SeleçãoNCS: estado responsável pela seleção de qual (ou quais) NCS será controlado por aquele controlador (microcomputador);
- ✓ 3. Monitoramento: estado responsável por monitorar a interface de usuário e a rede CAN, indicando a alteração de parâmetros e a presença de mensagens CAN disponíveis na fila de recebimento;
- ✓ 4. RecebimentoMSG: estado responsável pelo recebimento e decodificação da mensagem CAN nos campos e informações requeridos (identificador e dados);
- ✓ 5. DeterminaçãoNCS: estado responsável por identificar as mensagens CAN recebidas, verificando se elas serão usadas no controle de algum dos NCS selecionados previamente (estado 2) para o controlador. Esse estado também é responsável por produzir as tarefas e passar as informações necessárias para os respectivos laços consumidores;

- ✓ 6. Controle: estado responsável por utilizar a informação recebida do sensor e calcular o novo valor para o sinal de controle, respectivamente para todos os NCS selecionados previamente para o controlador. O cálculo do sinal de controle é realizado a partir de estratégias de controle que podem ser carregados ou desenvolvidos pelo usuário;
- ✓ 7. EnvioMSG: estado responsável pela transmissão das mensagens CAN, contendo as informações do sinal de controle a ser implementado pelos atuadores, respectivamente para todos os NCS selecionados para o controlador;
- ✓ 8. Parada: estado responsável por finalizar a operação do controlador e enviar mensagens CAN de parada de execução para todos os NCS em operação.

É importante observar que os laços consumidores existentes na arquitetura de *software* desenvolvida só são executados na existência de tarefas específicas criadas previamente pelo produtor. De forma que laços consumidores, relacionados a NCS que não estão configurados para aquele microcomputador, não irão compartilhar (ou desperdiçar) capacidade de processamento disponível de sua CPU.

A arquitetura de *software* desenvolvida para o controlador do NCS fornece uma configuração bastante flexível, pois permite a reutilização de código e também a replicação do programa em outros microcomputadores (ou controladores para NCS) com pequenas alterações (GODOY et al., 2010d).

Além disso, essa configuração de *software* fornece também facilidade para inserção de novos NCS e expansão da plataforma (através da configuração do estado 5 e inclusão de um novo laço consumidor para lidar com o novo NCS) e para testar diferentes estratégias de controle desenvolvidas para NCS (através da inclusão dos algoritmos de controle desenvolvidos na lista de opções utilizada no estado 6).

4.3. DESCRIÇÃO E MODELAGEM DAS MALHAS DE CONTROLE DA PLATAFORMA DE NCS

Para a realização do projeto e simulação da plataforma de pesquisa de NCS via rede CAN, é necessário o levantamento dos modelos matemáticos relativos às malhas de controle que compõem a mesma. Tais modelos matemáticos foram obtidos através de equações disponíveis na literatura e através de procedimentos experimentais para estimação de parâmetros e validação dos modelos (que podem ser vistos na seção 6). As seções seguintes apresentam uma descrição de cada um dos NCS implementados na plataforma, além de detalhes técnicos de componentes utilizados na construção das malhas de controle.

4.3.1. NCS DE CONTROLE DE VELOCIDADE DE MOTOR DC

Para experimentos de controle de velocidade de motores de corrente contínua, foi adaptado de um kit didático da T&S Equipamentos uma malha de controle para a plataforma de NCS, a qual é apresentada na Figura 51. Essa malha de controle é constituída por um motor DC, um encoder incremental *Hohner* de 600 pulsos por revolução para medição da posição, um drive de acionamento PWM baseado no MOSFET IRF9640, um circuito eletrônico de leitura do encoder, um display digital com 4 dígitos para apresentação da velocidade do motor em RPM, uma fonte AC/DC para alimentação dos dispositivos, conectores específicos para a rede CAN e um disco inercial para aplicação de cargas ao eixo do motor com o objetivo de gerar diferentes torques no motor. O objetivo deste NCS é a realização de experimentos de controle de velocidade do motor numa referência (em RPM) definida pelo usuário e sob diferentes cargas aplicadas ao sistema.

Figura 51. Malha de Controle de Velocidade de Motor DC adaptado para a Plataforma de NCS com Rede CAN

Os sinais de input (sinal de controle PWM para acionamento do motor DC) e output (medição da velocidade de rotação do motor através do encoder ou retransmissão analógica 0-5V de velocidade) do sistema são transmitidos através da rede CAN da plataforma, sendo que a integração da malha de controle à rede CAN foi feita utilizando-se da interface CAN descrita anteriormente.

O motor DC usado nesta malha de controle foi um motor Motron modelo M-910 de 24V. As Equações (1) a (4), disponíveis na literatura, apresentam as relações básicas utilizadas na modelagem matemática para controle de velocidade de motores de corrente contínua.

$$V = R.I + L.\frac{dI}{dt} + V_{emf} \quad (1)$$

$$V_{emf} = K_e \cdot w_m, \text{ com } w_m = \frac{d\theta_m}{dt} \quad (2)$$

$$T_m = \eta_m \cdot K_m \cdot I \quad (3)$$

$$T_m = J_m \cdot \frac{dw_m}{dt} + B_{eq} \cdot w_m \quad (4)$$

Com os parâmetros: R = Resistência de armadura do motor (Ω), V = tensão aplicada no motor (V), I = corrente aplicada no motor (A), L = indutância de armadura do motor (H), V_{emf} = tensão contra-eletromotriz do motor (V), K_e = constante contra-eletromotriz do motor (V.s/rad), w_m = velocidade angular do eixo do motor (rad/s), θ_m = rotação angular do eixo do motor (rad), K_m = constante de torque do motor (N.m/A), T_m = torque aplicado no eixo do motor (N.m), η_m = coeficiente de eficiência do motor (%), J_m = momento de inércia aplicado no eixo do motor (kg.m^2), B_{eq} = coeficiente de atrito equivalente (N.m.s/rad).

Aplicando a transformada de *Laplace* e rearranjando as equações, a função de transferência para controle de velocidade de um motor DC pode ser dada pela Equação (5), ignorando-se os baixos valores da indutância do motor utilizado. A função de transferência final, mostrada na Equação (6), foi obtida através da substituição de parâmetros de *datasheet*, procedimentos experimentais para validação do modelo para o NCS (mostrados na seção 6.6).

$$\frac{w_m(s)}{V} = \frac{\eta_m \cdot K_m}{J_{eq} \cdot L s^2 + (B_{eq} \cdot L + J_{eq} \cdot R)s + K_m \cdot K_e + B_{eq} \cdot R} \quad (5)$$

$$\frac{w_m(s)}{V} = \frac{0,95 \cdot \eta_m \cdot K_m}{J_{eq} \cdot L s^2 + 0,5 \cdot (B_{eq} \cdot L + J_{eq} \cdot R)s + K_m \cdot K_e + B_{eq} \cdot R} \cdot e^{-0,5s} = \frac{0,085}{0,003982s + 0,008144} \cdot e^{-0,5s} \quad (6)$$

4.3.2. NCS DE CONTROLE DE POSIÇÃO DE MOTOR DC

Para experimentos de controle de posição de motores DC, foi construída uma malha de controle para a plataforma de NCS, a qual é apresentada na Figura 52.

Essa malha de controle é constituída por um motor DC, um encoder para medição da posição, um drive de acionamento PWM baseado no circuito integrado ponte H LMD18200, um circuito eletrônico de leitura do encoder, uma fonte AC/DC para alimentação dos dispositivos, conectores específicos para a rede CAN e uma haste de alumínio na qual serão acoplados pesos com o objetivo de gerar diferentes torques no motor. O objetivo deste NCS é a realização de experimentos de controle de posição da haste em localização (referência em graus) definida pelo usuário e sob diferentes pesos acoplados ao sistema.

(a)

Figura 52. Malha de Controle de Posição de Motor DC construído para a Plataforma de NCS com Rede CAN: (a) Vista Geral, (b) Conexão da ECU na Rede CAN e Circuito de Leitura de Encoder e (c) Circuito de Acionamento e Acoplamento do Kit da Maxon Motor

Os sinais de input (sinal de controle PWM para acionamento do motor DC) e output (medição da posição da haste através do encoder) do sistema são transmitidos através da rede CAN da plataforma, sendo que a integração da malha de controle à rede CAN foi feita utilizando-se da interface CAN descrita anteriormente.

O NCS construído utiliza um kit da *Maxon Motor* composto por um motor DC modelo REmax de 11W e 24V, uma redução planetária de 109:1 e um encoder incremental de 500 pulsos por revolução. A modelagem matemática necessita levar em consideração a presença do redutor. Para isso, as Equações (7) a (10) são utilizadas juntamente com as Equações (1) a (4) definidas anteriormente (modelo do NCS de controle de velocidade).

$$\theta_m = k_r \cdot \theta_r \quad (7)$$

$$J_m \cdot \frac{d^2 \theta_m}{dt^2} = T_m + \frac{T_r}{\eta_r \cdot k_r} \quad (8)$$

$$J_r \cdot \frac{d^2 \theta_r}{dt} = T_r - B_{eq} \cdot \frac{d\theta_r}{dt} \quad (9)$$

$$J_{eq} = J_r + \eta_r \cdot k_r^2 \cdot J_m \quad (10)$$

Com os seguintes parâmetros: k_r = relação de redução (109:1), θ_r = rotação angular do eixo da redução (rad), T_r = torque na saída da redução (N.m), η_r = eficiência da redução (%), J_r = momento de inércia da redução (kg.m²), J_{eq} = momento de inércia equivalente do sistema (kg.m²).

Utilizando as equações apresentadas e aplicando a transformada de *Laplace*, a função de transferência para controle de posição de um motor DC com redução pode ser dada pela Equação (11). A função de transferência foi obtida através da substituição de parâmetros obtidos de *datasheet*, procedimentos experimentais para validação do modelo e ignorando-se os efeitos dos baixos valores da indutância do motor.

$$\frac{\theta_r}{V}(s) = \frac{\eta_r \cdot k_r \cdot \eta_m \cdot K_m}{J_{eq} \cdot L \cdot s^3 + (B_{eq} \cdot L + J_{eq} \cdot R) \cdot s^2 + (\eta_r \cdot k_r^2 \cdot \eta_m \cdot K_m \cdot K_e + B_{eq} \cdot R) \cdot s} = \frac{1,318}{0,02476 \cdot s^2 + 3,509 \cdot s} \quad (11)$$

4.3.3. NCS DE CONTROLE DE NÍVEL DE ÁGUA DE UM RESERVATÓRIO

Para experimentos de controle de nível em reservatórios, foi adaptada de um kit didático da T&S Equipamentos, uma malha de controle para a plataforma de NCS, a qual é apresentada na Figura 53. Essa malha de controle é constituída por dois taques, sendo o primeiro graduado (cm), mais alto e com uma torneira para alteração da vazão de saída de água para o segundo tanque, uma eletrobomba de água para enchimento do reservatório, um sensor de pressão da *Freescale* modelo MPX5010 para medição do nível de água no

reservatório, uma fonte AC/DC para alimentação dos dispositivos, conectores específicos para a rede CAN e um drive de acionamento PWM baseado no MOSFET IRF9640. O objetivo deste NCS é a realização de experimentos de controle do nível de água do reservatório superior numa localização (referência em cm) definida pelo usuário, sob diferentes vazões de saída do mesmo.

(a)

(b)

(c)

Figura 53. Malha de Controle de Nível de Reservatório adaptado para a plataforma de NCS com Rede CAN: (a) Vista Geral, (b) ECU e Acoplamento da Eletrobomba de Água e (c) Conexão do Sensor de Pressão para Medição do Nível de Água

Os sinais de input (sinal de controle PWM para acionamento da eletrobomba) e output (medição de nível do reservatório do sensor de pressão) do sistema são transmitidos através da rede CAN da plataforma, sendo que a integração da malha de controle à rede CAN foi feita utilizando-se da ECU descrita anteriormente.

A modelagem matemática para um sistema de controle de nível de um reservatório utilize as relações fornecidas pelas leis de balanço de massa e fluxo turbulento em sistemas fluidicos. Assim, de acordo com a literatura, as Equações (12) e (13) são utilizadas para a modelagem. A linearização da Equação não-linear (12) pode ser realizada em um ponto de operação selecionado $(\bar{h}, \bar{q})$, na Equação (14), considerando que erros no modelo obtido para pequenas variações em torno do ponto selecionado são insignificantes, resultando na Equação final (15).

$$A \cdot \frac{dh}{dt} = q_i - q_o \quad (12)$$

$$q_o = k \cdot \sqrt{h} \quad (13)$$

$$R = \frac{dh}{dq} = \frac{2 \cdot \bar{h}}{\bar{q}} = \frac{2 \cdot \sqrt{\bar{h}}}{k} \quad (14)$$

$$A \cdot \frac{dh}{dt} = q_i - \frac{2}{R} \cdot h \quad (15)$$

Com os parâmetros: q_i = vazão de entrada de água dada pela bomba no reservatório [cm^3/s], q_o = vazão de saída de água do reservatório [cm^3/s], h = nível de água do reservatório [cm], k = características da válvula de saída considerando fluxo de água turbulento (obtido experimentalmente $k = 8$), A = área da seção transversal do reservatório ($179,5 \text{ cm}^2$).

Aplicando a transformada de *Laplace* e rearranjando as equações, a função de transferência para controle de nível de um reservatório pode ser dada pela Equação (16). A

função de transferência final, mostrada na Equação (17), foi obtida a partir da substituição de parâmetros estimados em procedimentos experimentais, experimentos para validação do modelo (mostrados na seção 6.6) e utilizando o ponto de operação $\bar{h}=9$ cm.

$$\frac{h(s)}{q_i(s)} = \frac{1}{A.s + \frac{2}{R}} = \frac{\frac{R}{2}}{A.R.s/2 + 1} \quad (16)$$

$$\frac{h(s)}{q_i(s)} = \frac{1}{A.s + \frac{2}{R}} = \frac{\frac{R}{2}}{A.R.s/2 + 1} = \frac{\frac{R}{2}}{1,05 \cdot A.R.s/2} = \frac{0,375}{70,5.s + 1} \quad (17)$$

4.3.4. NCS DE CONTROLE DE TEMPERATURA

Para experimentos de controle de temperatura, foi construída uma malha de controle para a plataforma de NCS, a qual é apresentada na Figura 54. Essa malha de controle é constituída por um dispositivo aquecedor dado por um resistor de potência de 3,9 ohms e 20W, um sensor de temperatura da *National Semiconductor* modelo LM35, um drive de acionamento PWM baseado no MOSFET IRF9640 para aplicação de tensão variável, um circuito eletrônico de condicionamento da leitura do sensor, uma fonte AC/DC para alimentação dos dispositivos, conectores específicos para a rede CAN e um ventilador (*cooler*) para possibilitar um resfriamento mais rápido do sistema e aplicar perturbações.

Figura 54. Malha de Controle de Temperatura construída para a Plataforma de NCS com Rede CAN: (a) Vista Geral e (b) ECU, Circuitos de Acionamento do Aquecedor e Leitura do Sensor de Temperatura

O sensor de temperatura foi colocado conectado junto ao dispositivo aquecedor (dissipador de calor acoplado ao resistor) de forma a melhorar a medição da temperatura e a dinâmica do sistema. O objetivo deste NCS é a realização de experimentos de controle de temperatura do dispositivo aquecedor em relação a uma referência (em °C) definida pelo usuário.

Os sinais de input (sinal de controle PWM para aplicação de tensão variável sobre o resistor) e output (medição da temperatura do dispositivo aquecedor através do sensor) do sistema são transmitidos através da rede CAN da plataforma, sendo que a integração da malha de controle à rede CAN foi feita utilizando-se da interface CAN descrita anteriormente.

A modelagem, matemática para controle de temperatura utiliza as relações fornecidas pelas leis da conservação de energia e transferência de calor por convecção em sistemas térmicos. Assim, de acordo com a literatura, as Equações (18) a (20) são utilizadas para obtenção do modelo matemático.

$$q_i - q_o = C \cdot \frac{d(T_s - T_a)}{dt} \quad (18)$$

$$q_o = \frac{1}{R} \cdot (T_s - T_a) \quad (19)$$

$$T = T_s - T_a \quad (20)$$

Com os parâmetros: q_o = fluxo de calor de saída ou taxa de resfriamento [W], q_i = fluxo de calor de entrada ou taxa de aquecimento [W], T_s = temperatura medida pelo sensor [°C], T_a = temperatura ambiente [°C], T = alteração ou mudança de temperatura [°C], R = resistência térmica do dispositivo [°C/W], θ = tempo morto do sistema (tempo necessário para alteração da temperatura após aplicação de um fluxo de calor no sistema [s], C = capacitância térmica do dispositivo (J/°C).

Utilizando as equações descritas anteriormente juntamente com o tempo morto inerente em sistemas térmicos e aplicando a transformada de *Laplace*, a função de transferência para controle de temperatura pode ser dada pela Equação (21). A função de transferência final foi obtida a partir de procedimentos experimentais para validação do modelo e estimação de parâmetros de sistemas de primeira ordem baseados na resposta do sistema a um degrau de entrada em malha aberta.

$$\frac{T(s)}{q_i(s)} = \frac{R}{R \cdot C \cdot s + 1} \cdot e^{-\theta \cdot s} = \frac{17,19}{78,42 \cdot s + 1} \cdot e^{-s} \quad (21)$$

4.3.5. NCS DE CONTROLE DE MOVIMENTAÇÃO DE ESTEIRA

Para experimentos de controle de movimentação de esteira transportadora, foi construída uma malha de controle para a plataforma de NCS, a qual é apresentada na Figura 55. Essa malha de controle é constituída por uma esteira de movimentação em pequena escala (50cm) construída em alumínio e nylon, e acionada por um motor DC da Bosch com redução de 75:1, um drive de acionamento PWM baseado no circuito integrado ponte H L298, quatro conjuntos de emissor/receptor de infravermelho utilizados como sensores de presença, uma fonte AC/DC para alimentação dos dispositivos, conectores específicos para a rede CAN e peças de tamanhos diferentes para serem movimentadas. O objetivo deste NCS é a realização de experimentos de controle on/off de movimentação de peças inseridas na esteira entre dois pontos distintos na mesma. De acordo com o tamanho da peça, mensurada na primeira estação após a peça ser inserida na esteira, o controle é realizado para movimentar a respectiva peça até a posição requerida, entre duas estações possíveis.

(a)

(b)

(c)

Figura 55. Malha de Controle de Esteira de Movimentação construída para a Plataforma de NCS com Rede CAN: (a) Vista Geral, (b) Conexão na ECU na Rede CAN e Sensores de Presença e (c) Acoplamento do Motor de Movimentação da Esteira

Os sinais de input (sinal de controle PWM para acionamento do motor DC) e output (medição do tamanho da peça na primeira estação e verificação de presença na estação requerida através do sensor de presença) do sistema são transmitidos através da rede CAN da plataforma. A integração da malha de controle à rede CAN foi feita utilizando-se da interface CAN descrita anteriormente.

Conforme dito anteriormente, por se tratar de uma malha de controle on/off, esse NCS não necessitou ser modelado. A dinâmica de funcionamento desse sistema será simulada através de uma ECU com capacidade de enviar e receber mensagens pela rede CAN.

4.4. INFORMAÇÕES DE OPERAÇÃO DA PLATAFORMA DE NCS

Uma informação muito importante relacionada à aplicação de NCS e, portanto a respeito dos NCS que compõe a plataforma desenvolvida é seu valor limite de amostragem.

Este valor limite representa o menor (ou mais rápido) período de amostragem das mensagens em que um NCS pode operar sem que ocorra sobreposição ou perda de mensagens transmitidas e/ou grandes atrasos de comunicação que sobrecarreguem as filas (GODOY et al., 2010d) do controlador. Este parâmetro de desempenho, juntamente com o tempo de ciclo em malha fechada do NCS (tempo desde a amostragem até a atuação), pode ser utilizado, por exemplo, para se definir a capacidade de operação em tempo real e a viabilidade de utilização da arquitetura desenvolvida para a plataforma para aplicação em NCS.

Nos experimentos realizados neste trabalho, o máximo valor obtido para o valor limite de amostragem foi igual a 1ms para operação de um único NCS na rede CAN. Já para o tempo de ciclo em malha fechada do NCS, um valor igual a 5ms pode ser garantido com a arquitetura desenvolvida e utilizada nos experimentos.

Como a rede CAN da plataforma de NCS desenvolvida utiliza uma velocidade de transmissão de dados de 250kb/s (compatível com padrões industriais de redes baseadas no CAN como CANOpen, DeviceNet, SAE J1939 e ISO11783), que representa uma baixa velocidade quando comparada com redes industriais baseadas em Ethernet (que podem chegar a 1Gbit/s), foi verificado a variabilidade deste valor limite de amostragem de acordo com a operação conjunta de diversos NCS na plataforma ou da presença de tráfego adicional de mensagens. Assim, um aumento do valor limite de amostragem de 1ms para até 10ms pôde ser verificado de acordo com os experimentos realizados (aumento para 5ms: operação da plataforma com os cinco NCS em funcionando e aumento para 10ms: operação da plataforma com os cinco NCS em funcionamento mais tráfego extra de mensagens na rede CAN representando uma carga de 55% de sua capacidade total).

4.5. CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas as etapas referentes à proposta e ao desenvolvimento da plataforma de pesquisa de NCS com redes CAN. A aplicação da plataforma desenvolvida será importante no sentido de fornecer informações, como principais problemas e formas de implementação, sobre o desenvolvimento de NCS com redes CAN, além de permitir a validação de resultados teóricos obtidos através de simulações.

A arquitetura de *hardware* desenvolvida para a plataforma forneceu grande flexibilidade para realização de experimentos, tornando o desenvolvimento dos NCS mais simples, fácil e de baixo custo. A utilização da teoria de produtor/consumidor e máquinas de estado na implementação de *software* facilitou o desenvolvimento de código e forneceu uma configuração bastante flexível e expansível para os NCS.

5. ESTRATÉGIAS DE PROJETO E CONTROLE DE NCS

5.1. CONSIDERAÇÕES INICIAIS

Neste capítulo são abordados os conceitos e definições necessárias para o desenvolvimento de uma estratégia de controle para NCS. Apresentam-se uma estrutura padrão para desenvolvimento de NCS e as justificativas que norteiam o desenvolvimento dos controladores do NCS em tempo discreto. Baseando-se na necessidade de desenvolver estratégias de controle para NCS menos complexas, de simples implementação e maior aplicabilidade prática, uma estrutura de controlador PID discreto é proposta para aplicação em NCS. Utilizando-se da estrutura desse controlador PID proposto e com o objetivo de lidar com um dos principais problemas de NCS com redes CAN, uma nova estratégia de controle adaptativo, que gerencia automaticamente o período de amostragem das mensagens do NCS, é desenvolvida. Uma técnica de estimação de modelos matemáticos para auxiliar o projeto de NCS é apresentada e seus benefícios discutidos.

5.2. ESTRUTURA DE UM NCS

Pesquisas recentes sobre modelagem de NCS têm sido realizadas utilizando modelos baseados em tempo contínuo e tempo discreto. No entanto, é necessária e mais realista, a utilização do ponto de vista de modelagem discreta a partir do momento que na operação de um NCS, os sinais ou informações do sensor e do controlador são amostrados e então transmitidos numa rede de comunicação. Para modelos contendo estruturas no tempo discreto,

muitos pesquisadores assumem que a rede de comunicação possui sincronismo e que os sensores, controladores e atuadores possuem o mesmo período de amostragem.

No desenvolvimento de NCS, a estrutura do sistema de controle utilizada é na maioria das vezes dada pela Figura 56.

Figura 56. Estrutura de Implementação de um NCS

De acordo com a Figura 56, o processo a ser controlado (ou planta) é admitido ser contínuo no tempo e o atuador possui um segurador de ordem zero (ZOH – *zero order hold*) que armazena o último sinal de controle amostrado até a chegada do próximo valor ou até o próximo período de amostragem. A rede de comunicação do NCS é então usada para transmissão de medidas de saída da planta para o controlador, fazendo com que a planta seja medida de acordo com um período de amostragem (h). Esta discretização dos sinais acaba motivando o desenvolvimento e a aplicação de controladores baseados em tempo discreto. O funcionamento do controlador do NCS pode ser baseado em tempo (*time-driven*) ou baseado em eventos (*event-driven*). Assim um novo sinal de controle pode ser calculado a cada intervalo de tempo com um período de amostragem constante ou um novo sinal de controle é

calculado imediatamente após o recebimento, através de uma mensagem na rede, de uma nova medida da saída da planta pelo sensor.

A troca de informações através dos dispositivos do NCS (sensor, controlador, atuador) origina uma transmissão de mensagens (*msg*) na rede de comunicação. Essa transmissão de mensagens acaba induzindo a presença de atrasos de comunicação (T_{delay}) na rede do NCS. Estes atrasos de comunicação na rede, detalhadamente explicados anteriormente (seção 2.3), dependem de uma série de fatores relacionados, por exemplo, ao *hardware* e *software* dos dispositivos e a rede de comunicação utilizada no NCS e muitas vezes podem ter características variantes no tempo (SO, 2003).

O período de amostragem (*h*) também representa um importante parâmetro no desenvolvimento do NCS. Em sistemas de controle digital, o desempenho do sistema se aproxima do desempenho de sistemas de controle contínuo conforme o período de amostragem se torna mais rápido (ou de valor pequeno). No entanto, em NCS um período de amostragem muito rápido pode acarretar alta carga e tráfego de mensagens na rede, aumentando o risco de congestionamento de mensagens e originando atrasos de comunicação maiores e conseqüentemente uma degradação do desempenho do NCS. Portanto o atraso de comunicação e o período de amostragem em NCS são parâmetros de configuração inter-relacionados e que necessitam ser balanceados para atingir a estabilidade e desempenho requeridos pelo NCS (ZHANG, BRANICKY & PHILLIPS, 2001).

O desenvolvimento de NCS apresenta novos desafios para as tradicionais técnicas de projeto de sistemas de controle. Em NCS, o projeto de controle deve levar em consideração, simultaneamente, os efeitos da amostragem dos sinais e dos atrasos de comunicação, que na maioria das vezes é variante no tempo. A planta ou processo a ser controlado e as especificações de desempenho requeridas para o NCS é que vão definir se técnicas de controle conhecidas poderão ser aplicadas para compensar os efeitos da presença da rede de

comunicação no sistema, ou se será necessária a aplicação de novas estratégias para o projeto e controle do NCS (BAILLIEUL & ANTSAKLIS, 2007).

A literatura sobre metodologias de projeto e controle para NCS geralmente tem focado nos atrasos de comunicação da rede e em diferentes estratégias para lidar com esse e outros fatores degenerativos que afetam o desempenho do NCS. De acordo com Tipsuwan & Chow (2003), uma metodologia de controle é requerida para mitigar os efeitos desses fatores de forma a garantir requisitos de estabilidade e desempenho do NCS. Essas metodologias de controle, na maioria das vezes, diferem em diversos aspectos das técnicas de controle convencionais e ainda não existe uma solução padrão que possa ser utilizada (ERIKSSON, 2008). Uma questão que tem obtido importância crescente é a arquitetura do controlador escolhido. Uma revisão de algumas metodologias de controle é apresentada em Tipsuwan & Chow (2003), Hespanha, Naghshtabrizi & Xu (2007), Nair et al. (2007) e Vatanski et al. (2009).

A maioria das abordagens descritas nesses trabalhos apresenta soluções complexas, com algoritmos que necessitam de grande capacidade de processamento e muitas vezes utilizam informações sobre a rede de comunicação e seus atrasos de comunicação, que são difíceis de serem obtidos. Estes fatores acabam por tornar a aplicabilidade prática dessas estratégias de controle na indústria tanto quanto questionável (ERIKSSON, 2008). Este fato tem norteado o desenvolvimento de controladores para NCS com estruturas e algoritmos simples e de fácil implementação. Dependendo da planta a ser controlada e dos requisitos de controle a serem alcançados, muitas vezes controladores bastante conhecidos podem compensar os efeitos degenerativos da rede e dos atrasos de comunicação.

No entanto, o desenvolvimento de estratégias de controle PID para NCS ainda é pouco difundido e seu projeto, estrutura e sintonia de parâmetros têm sido discutidos e investigados em alguns trabalhos (POHJOLA, 2006; GHUDE, 2007; ALLDREDGE, 2007; RAJU, 2009).

Baseando-se nesses fatos, este tema torna-se muito relevante a partir do momento que controladores PID são extensivamente empregados na indústria e possuem grande flexibilidade de utilização.

5.3. ESTRATÉGIA DE CONTROLE PID DISCRETO

Os controladores usados em controle de processos e sistemas podem ter diversas estruturas. A escolha de uma estrutura pode determinar quão bem uma planta pode ser controlada. A planta impõe alguns requisitos sobre o controlador utilizado e a escolha do controlador e sua sintonia podem definir o melhor desempenho possível que pode ser alcançado. Controladores PID são amplamente utilizados na indústria principalmente devido a sua grande versatilidade e pelo fato seu algoritmo ser bastante intuitivo (ÅSTRÖM & HÄGGLUND, 1995). Esse tipo de controlador ideal não leva em consideração os atrasos de comunicação presentes no NCS. No entanto esses fatores degenerativos podem ser mitigados através de uma correta definição da estrutura do controlador e de sua sintonia de parâmetros (ERIKSSON, 2008).

Estratégias de controle PID para NCS têm sido relativamente pouco pesquisadas apesar de a definição de sua estrutura e sintonia de parâmetros serem decisivas para aplicações em NCS. Controladores PID têm sido discutidos na literatura recente em aplicações conjuntas com NCS (ALLDREDGE, 2007; ERIKSSON, 2008, RAJU, 2009), porém raramente as características da rede de comunicação industrial utilizada são levadas em consideração na sintonia do controlador. No entanto, com uma sintonia de parâmetros apropriada, um controlador PID pode tornar-se robusto em relação a presença de atrasos de comunicação e erros na transmissão de mensagens na rede (POHJOLA, 2006).

Em aplicações de controladores PID para NCS, algumas modificações na estrutura do controlador e na forma de implementação de seu algoritmo têm sido propostas e discutidas. O único conhecimento já consolidado e difundido é a necessidade do desenvolvimento de controladores PID no tempo discreto, devido ao fato de, em NCS, as medições serem transmitidas através de mensagens na rede de comunicação e o controlador ser discretizado de acordo com esse período de amostragem das mensagens (POHJOLA, 2006).

O diagrama de blocos de um sistema de controle clássico com realimentação de saída é mostrado na Figura 57.

Figura 57. Esquemático de um Sistema de Controle com Controlador PID

O processo ou planta (G) a ser controlado é mensurado através de sensores (H) e controlado através do controlador (C). As variáveis (r), (u), (y) e (e) representam respectivamente o valor do sinal de referência de entrada, o valor do sinal de controle, o valor de saída medido da planta e o erro (diferença) entre a referência de entrada e o valor de saída medido. Geralmente, em sistemas de controle, o modelo do sensor (H) é assumido igual a 1. No entanto este modelo pode levar em consideração as características do sinal medido. Um controlador PID no tempo contínuo pode ser calculado através da Equação (22) que origina um sinal de controle (u) relacionado ao erro do sistema (e).

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (22)$$

Na Equação (22), os parâmetros de sintonia do controlador são K (ganho proporcional), T_i (tempo integrativo) e T_d (tempo derivativo). Uma forma equivalente de controladores PID é apresentada na Equação (23). Nesse formato, os parâmetros de sintonia do controlador são $K_p = K$ (ganho proporcional), $K_i = K/T_i$ (ganho integrativo) e $K_d = K \cdot T_d$ (ganho derivativo).

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(t) dt + K_d \cdot \frac{de(t)}{dt} \quad (23)$$

Transformando a equação do controlador PID usando a transformada de *Laplace*, obtém-se a função de transferência apresentada na Equação (24) relacionada com o diagrama de blocos padrão de um controlador PID apresentada na Figura 58.

$$C_{PID}(s) = \frac{u(s)}{e(s)} = K_p \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d s \right) = P + \frac{I}{s} + D \cdot s \quad (24)$$

Figura 58. Estrutura de um Controlador PID Padrão

Um controlador PID possui três partes que realizam ações diferentes num sistema de controle. A parte proporcional (u_p) possui ação de controle proporcional ao valor do erro do sistema, a parte integral (u_i) possui ação de remover o erro em regime permanente do sistema e a parte derivativa (u_d) possui a ação de remover o overshoot e melhorar a estabilidade do

sistema. O peso dessas ações de controle num controlador PID é então ajustado através dos valores dos ganhos P, I e D.

No entanto, conforme dito anteriormente um controlador PID para NCS necessita ser desenvolvido no tempo discreto, onde a discretização deve ser feita através do cálculo do erro do sistema de acordo com intervalos de tempo discretos, $t.k = k.h$, relacionados com o período de amostragem (h) das mensagens transmitidas na rede de comunicação e substituindo a parte integral por uma soma e a parte derivativa por uma diferença.

Assim, a parte proporcional do controlador PID discretizado em intervalos de tempo (k) pode ser dada pela Equação (25).

$$u_p[k] = u_p(tk) = K_p \cdot e(kh) \quad (25)$$

A parte integral do controlador PID discreto, mostrada na Equação (26), pode ser obtida através da aproximação da integral com uma soma e realizando-se após, uma simplificação através do cálculo da diferença, de acordo com a Equação (27), resultando na Equação (28).

$$u_i[k] = \frac{K_p}{T_i} \cdot \sum_{n=0}^k h \cdot e[n] \quad (26)$$

$$u_i[k] - u_i[k-1] = \frac{K_p}{T_i} \cdot \sum_{n=0}^k h \cdot e[n] - \frac{K_p}{T_i} \cdot \sum_{n=0}^{k-1} h \cdot e[n] \quad (27)$$

$$u_i[k] = u_i[k-1] + \frac{K_p \cdot h}{T_i} \cdot e[k] \quad (28)$$

De acordo com a literatura, a parte derivativa do controlador PID pode ser discretizado de diversas formas de forma a se priorizar estabilidade e precisão do mesmo. A forma mais simples de discretização é através da aproximação de Euler ou diferença para frente (*forward difference*) dada pela Equação (29).

$$u_d[k] = K_p \cdot T_d \cdot \frac{e[k+1] - e[k]}{h} \quad (29)$$

No entanto, essa forma de discretização não é frequentemente usada em controladores PID por amplificar erros devido a ruídos do sistema de controle.

Em aplicações de controladores PID discretos em NCS, diversas modificações na estrutura do controlador e em sua forma de implementação podem ser encontradas objetivando-se adequar o mesmo aos vários aspectos reais presentes como presença de ruídos, erros de medição, rápidas alterações de referência e saturação dos atuadores utilizados.

Nos tópicos a seguir são apresentadas algumas dessas modificações no sentido de sistematizar um controlador PID discreto completo para ser aplicado em NCS.

5.3.1. FILTRAGEM DA PARTE DERIVATIVA

Como observado na ação de controle da parte derivativa, o controlador PID possui uma alta sensibilidade ao ruído do sistema. A parte derivativa amplifica ruídos existentes originados na medição dos sinais. Para evitar que isto aconteça, é aplicada uma filtragem da parte derivativa que é responsável por suavizar a ação derivativa do controlador PID (ÅSTRÖM & HÄGGLUND, 1995). No domínio do tempo contínuo, a implementação da parte derivativa com filtragem pode ser obtida de acordo com a Equação (30), que pode ser simplificada na Equação (31).

$$u_d(t) = -\frac{T_d}{N} \cdot \frac{du_d(t)}{dt} + K_p \cdot T_d \cdot \frac{de(t)}{dt} \quad (30)$$

$$u_d(s) = \frac{K_p T_d s}{1 + \frac{s T_d}{N}} \quad (31)$$

Esta Equação (31) pode ser interpretada como o termo derivativo $s.T_d$ sendo filtrado por um sistema de primeira ordem com tempo constante T_d/N , ou seja, atua como derivativa para sinais de baixa frequência ou filtro passa baixa. O ganho, no entanto, fica limitado a $K_p.N$, isto significa que a medição do ruído a altas frequência é amplificado pelo fator $K_p.N$. Quanto menor o valor de N maior será a filtragem da parte derivativa realizada. Geralmente o valor de N varia de 1 a 20 (ÅSTRÖM & HÄGGLUND, 1995).

De acordo com Isaksson & Graebe (2002), um controlador PID com filtragem da parte derivativa acaba tendo quatro graus de liberdade, pois a constante N necessita ser definida em adição aos parâmetros P , I e D . No entanto, o mesmo trabalho afirma que esse tipo de tarefa não é comumente realizado, definindo-se o valor de $N=10$ ou de acordo com a quantidade de ruído que pode existir nas medições realizadas no sistema de controle.

A discretização da parte derivativa do controlador PID com filtragem é então realizada através de outra forma de aproximação chamada de diferença para trás (*backward difference*), de acordo com a Equação (32), que pode ser manipulada até a forma da Equação (33).

$$u_d[k] = -\frac{T_d}{N} \cdot \frac{u_d[k] - u_d[k-1]}{h} + K_p T_d \frac{e[k] - e[k-1]}{h} \quad (32)$$

$$u_d[k] = \frac{T_d}{T_d + N.h} u_d[k-1] + \frac{K_p T_d N}{T_d + N.h} (e[k] - e[k-1]) \quad (33)$$

5.3.2. PONDERAÇÃO DE REFERÊNCIA (SETPOINT WEIGHTING E REFERENCE OFF)

O sinal de entrada usual de cada parte (proporcional, integrativo e derivativo) do controlador PID é o erro entre o sinal de referência e o sinal de saída medido. Se o sinal de

erro está sujeito a mudanças repentinas de valor, o modo derivativo tende a produzir instantaneamente valores muito altos que podem saturar o sinal de controle.

O sinal de saída medido não apresenta alterações bruscas de valor, pois a própria dinâmica associada ao processo funciona como um filtro. Uma mudança brusca (degrau) no sinal de referência (*setpoint*) pode gerar um impulso no sinal de controle. Estes dois problemas na maioria das vezes são indesejáveis e podem ser contornados ponderando-se (ou ajustando-se) o sinal de referência antes de enviá-lo ao controle, numa modificação que é chamada de ponderação de referência (*setpoint weighting e reference off*) (ÅSTRÖM & HÄGGLUND, 1995).

Com isso obtém-se um controlador PID que é muito utilizado para não derivar e nem amplificar variações bruscas no sinal de referência e que assume no domínio do tempo contínuo a forma apresentada na Equação (34) (ÅSTRÖM & HÄGGLUND, 1995).

$$u(t) = K_p \left(e_p(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de_d(t)}{dt} \right) \quad (34)$$

Onde o erro na parte proporcional (e_p), o erro na parte derivativa (e_d) e o erro na parte integrativa (e) que representa o verdadeiro erro do sistema de controle, são dados respectivamente pelas Equação (35), Equação (36) e Equação (37).

$$e_p(t) = B.r(t) - y(t) \quad (35)$$

$$e_d(t) = C.r(t) - y(t) \quad (36)$$

$$e(t) = r(t) - y(t) \quad (37)$$

A parte integrativa do controlador PID não apresenta ponderação, caso contrário o erro em regime permanente não poderia ser compensado. Assim, a Equação (38) apresenta o

formato do algoritmo e a Figura 59 apresenta o diagrama de blocos de um controlador PID com ponderação de referência.

$$u(t) = K_p \left(B.r(t) - y(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \left[C \frac{dr(t)}{dt} - \frac{dy(t)}{dt} \right] \right) \quad (38)$$

Figura 59. Estrutura de um Controlador PID com Ponderação de Referência

Nos controladores PID com ponderação de referência, a resposta para a variação do sinal de referência pode ser modificada pelos parâmetros B e C (ÅSTRÖM & HÄGGLUND, 1995). Na técnica chamada *setpoint weighting*, o sobressinal do sistema para mudanças do *setpoint* é menor para valores menores do parâmetro B que ajusta a entrada da parte proporcional. O parâmetro C é geralmente zero, numa técnica chamada de *reference off*, que utiliza somente a saída do processo (y) como entrada para a parte derivativa, para evitar grandes mudanças no sinal de controle devido a mudanças repentinas do *setpoint* (ÅSTRÖM & HÄGGLUND, 1995).

5.3.3. SATURAÇÃO DO ATUADOR E ANTI-WINDUP DA AÇÃO DE CONTROLE INTEGRAL

Os processos industriais são submetidos a restrições construtivas, ou seja, a faixa de valores do sinal de controle aplicado ao processo é definida pelas suas características intrínsecas. Por exemplo, um controlador opera num intervalo limitado de 0–10 volts ou 4–20 mA, uma válvula não pode abrir mais que 100% e menos que 0%, um motor funcionando como atuador tem um limite de velocidade, etc. Tais restrições são usualmente descritas como limitações na entrada da planta. Por tanto, é preciso limitar os sinais de controle aplicados ao processo dentro uma faixa de operação especificada. O sinal de entrada do processo deve ser limitado dentro desta faixa de operação, chamada de saturação. Como resultado das limitações, a entrada real da planta é temporariamente diferente da saída do controlador. Quando isto acontece, se o controlador é inicialmente projetado para operar em uma região linear, o desempenho em malha fechada deteriora-se em relação ao desempenho linear esperado. Esta deterioração de desempenho é denominada *windup* (ÅSTRÖM & HÄGGLUND, 1995).

Essa modificação na parte integrativa de controladores PID chamada de *Anti-Windup*, consiste em atenuar o efeito produzido pela saturação do sinal de controle no desempenho do sistema. Um das formas de aplicar essa ação é evitar que o modo integral mantenha o atuador saturado mesmo quando o erro diminui. A técnica de *back-calculation* funciona da seguinte forma: quando a saída do atuador satura, o termo integrativo é novamente calculado de forma que seu valor permaneça no limite do atuador. É vantajoso fazer esta correção não instantaneamente, mas dinamicamente com uma constante de tempo T_t . O diagrama de blocos da Figura 60 exemplifica a inclusão de uma ação *Anti-Windup* num controlador PID e a

Equação (39) apresenta a forma de implementação da parte integrativa de um controlador PID discreto com *Anti-Windup* (ÅSTRÖM & HÄGGLUND, 1995).

$$u_i(k) = u_i(k-1) + \frac{K_p \cdot h}{T_i} e(k) + \frac{K_p \cdot h}{T_t} e_s(k-1) \quad (39)$$

Figura 60. Estrutura de um Controlador PID com *Anti-Windup* da Ação Integral

O sistema apresenta uma malha de realimentação adicional. A diferença entre a entrada e saída do atuador constitui um erro adicional (e_s) que é realimentado à entrada do integrador com ganho $1/T_t$. Note que quando não existe saturação o erro adicional (e_s) é nulo e, portanto, a malha não tem nenhum efeito quando o controlador está operando linearmente, ou seja, quando a saída $u(t)$ não está saturada. Se existe a saturação, o erro adicional (e_s) é diferente de zero. O tempo para que a entrada do integrador chegue a zero é determinado pelo ganho $1/T_t$, onde T_t pode ser interpretado como a constante de tempo que determina o quão rápido a entrada do integrador é levada a zero. Assim, a seleção de valores pequenos para T_t pode parecer vantajosa à primeira vista.

Entretanto, deve-se ter cuidado na seleção de T_t especialmente em sistemas com ação derivativa (ÅSTRÖM & HÄGGLUND, 1995). O que pode acontecer é que ruídos podem

levar a saída do controlador à saturação provocando a atuação rápida da malha *Anti-Windup* e tornando a entrada do integrador indesejavelmente zero. Na prática deve-se ter T_t maior que T_d e menor que T_i . Uma regra empírica sugerida é selecionar T_t igual a $\sqrt{T_i T_d}$ para controladores PID e igual a T_i para controladores PI (ÅSTRÖM & HÄGGLUND, 1995).

5.3.4. ESTRUTURA DEFINIDA DO CONTROLADOR PID PARA NCS

Diante da sua grande versatilidade, um controlador PID foi projetado para aplicação em NCS. A estrutura definida é composta de controlador PID discreto com aproximação derivativa (*backward difference*), ponderação de referência (*setpoint weighting* e *reference off*), filtragem da ação derivativa e *Anti-Windup* (*back-calculation*) da ação integral. Figura 61 apresenta um esquemático do controlador PID resultante com constante de filtragem da parte derivativa (N) e constantes de ponderação (B e C).

Figura 61. Estrutura do Controlador PID Completo Definido para NCS

O algoritmo de implementação do controlador PID discreto definido para NCS com período de amostragem (h) é dado pelas Equações (40) até (45), finalmente resultando na Equação (46).

$$e(k) = r(k) - y(k) \quad (40)$$

$$e_p(k) = B.r(k) - y(k) \quad (41)$$

$$e_d(k) = C.r(k) - y(k) \quad (42)$$

$$u_p(k) = K.e_p(k) \quad (43)$$

$$u_i(k) = u_i(k-1) + \frac{K.h}{T_i} e(k) + \frac{K.h}{T_t} e_s(k-1) \quad (44)$$

$$u_d(k) = \frac{T_d}{T_d + Nh} u_d(k-1) + \frac{KT_d N}{T_d + Nh} [e_d(k) - e_d(k-1)] \quad (45)$$

$$u_{PID}(k) = u_p(k) + u_i(k) + u_d(k) \quad (46)$$

5.4. ESTRATÉGIA DE CONTROLE ADAPTATIVO DO PERÍODO DE AMOSTRAGEM

Conforme discutido na revisão de literatura, um dos principais problemas relacionados ao desenvolvimento e a aplicação de NCS diz respeito ao efeito do período de amostragem das mensagens no desempenho e estabilidade de um NCS (LIAN et al., 2006). A influência desse parâmetro de configuração de NCS tem sido investigada e avaliada em diversos trabalhos (XIA, WANG & SUN, 2004; VALDIVIA & SOUZA, 2007).

Em se tratando de NCS com redes CAN, a correta definição do período de amostragem é ainda mais necessária e importante. Godoy; Porto & Inamasu (2010a) verificaram através de simulações que o período de amostragem das mensagens representa o principal parâmetro que afeta o desempenho de controle de NCS com redes CAN. Isso se deve não somente devido à grande influência desse parâmetro no desempenho do NCS, como também pelo fato desse

parâmetro estar fortemente relacionado à taxa de utilização da rede CAN. Caso o período de amostragem dos NCS seja muito rápido, pode ocorrer a saturação da rede de comunicação. Nessa situação, onde a rede CAN apresenta alta taxa de utilização e fica sobrecarregada de mensagens, grandes atrasos de comunicação são induzidos e erros de transmissão de mensagens tornam-se constantes, deteriorando o desempenho e podendo tornar o NCS instável (GODOY; PORTO & INAMASU, 2010a). A Figura 62 exemplifica, através de dois gráficos obtidos através de simulações, os efeitos da alteração do período de amostragem e da taxa de utilização da rede CAN no desempenho de controle de um NCS com rede CAN.

Figura 62. Efeito do Período de Amostragem e da Taxa de Utilização da Rede no Desempenho de um NCS com Rede CAN

A abordagem de projeto e implementação de NCS mais comum consiste na execução periódica de um algoritmo de controle, através da definição de um período de execução fixo do controlador (*time-driven*) ou através da definição de um período de amostragem das mensagens do sensor (*event-driven*) (GUPTA & CHOW, 2010). Essa técnica utiliza de maneira estática a largura de banda disponível para comunicação de dados sem considerar outros fatores como a taxa de utilização momentânea da rede de comunicação e mudanças nos sistemas que estão sendo controlados.

No entanto, resultados de trabalhos recentes têm comprovado que a utilização de técnicas de definição dinâmica desse período de execução pode acarretar num melhor desempenho de controle do NCS quando comparado à utilização de técnicas consolidadas de definição estática ou de período de execução fixo (CERVIN et al., 2010). Duas tendências recentes podem ser identificadas para o projeto de estratégias de controle que usam definição dinâmica desse período de execução (CAMACHO et al., 2010).

A primeira tendência utiliza técnicas de alteração de período de execução do controlador de acordo com a dinâmica de informações do NCS como resposta e condições de largura de banda (ou taxa de utilização) da rede de comunicação. O objetivo principal dessas abordagens é melhorar o desempenho de controle dos NCS utilizando eficientemente toda a largura de banda disponível da rede para transmissão de dados. Martí et al. (2010) apresenta uma técnica que consegue melhorar o desempenho de um conjunto de NCS com redes CAN através da transmissão adicional (não periódica) de mensagens de controle, realizada de maneira proporcional à disponibilidade de largura de banda da rede CAN no momento.

A segunda tendência usa técnicas baseadas em eventos para disparo do controlador, o qual origina execuções não periódicas do mesmo e, portanto envio de mensagens na rede da mesma forma. O objetivo dessas abordagens é diminuir a taxa de utilização da rede, garantindo requisitos de desempenho e estabilidade do NCS. Camacho et al. (2010) apresenta um estudo experimental sobre a implementação de controladores disparados automaticamente (*self-triggered*) para NCS com rede CAN. Nesse trabalho, além das tarefas padrões realizadas a cada ciclo de execução, o controlador utiliza um algoritmo para calcular quando o próximo ciclo de execução será disparado. Os resultados desse trabalho mostram semelhante desempenho de controle para o NCS com o controlador disparado automaticamente projetado, em relação a um controlador periódico, porém obtendo-se uma pequena diminuição na taxa de

utilização da rede CAN (o qual é calculada em termos de mensagens enviadas, com redução de 18000 para 15384 mensagens durante o experimento realizado).

De forma a lidar com o problema do período de amostragem em NCS com redes CAN e baseando-se nas abordagens citadas para definição dinâmica deste parâmetro, este trabalho apresenta o desenvolvimento de uma nova estratégia de controle adaptativo para NCS com redes CAN (GODOY; PORTO & INAMASU, 2010b). A definição pelo controle adaptativo foi norteadada pela analogia entre o objetivo deste tipo de controle e o objetivo desejado (a ser atingido) para o NCS. Åström & Wittenmark (1995) descrevem o controle adaptativo como a capacidade de sistemas de controle de modificar seus próprios parâmetros em resposta a alterações constatadas em alguma variável monitorada. Neste sentido, adaptar-se significa mudar o comportamento em resposta a novas circunstâncias de operação, com o objetivo de manter um nível esperado de desempenho.

A estratégia de controle adaptativo proposta neste trabalho compartilha do mesmo objetivo da técnica de disparo a eventos, que foca na diminuição da taxa de utilização da rede ao mesmo tempo garantindo o desempenho de controle requerido. No entanto, essa estratégia inova e difere da citada, uma vez que utiliza os conceitos da outra técnica descrita, de definição dinâmica do período de amostragem de acordo com informações do NCS, para obter esse objetivo de diminuição da taxa de utilização da rede CAN de forma mais ampla, fato que representa uma vantagem dessa estratégia de controle adaptativo desenvolvida.

A nova estratégia desenvolvida neste trabalho é baseada num controlador PID adaptativo que gerencia automaticamente o período de amostragem (PA) do NCS de acordo com informações de desempenho de controle do próprio NCS. Com a definição dinâmica desse período de amostragem (PA) das mensagens do sensor do NCS, obtém-se da mesma forma uma alteração do período de execução do controlador e, portanto uma alteração na transmissão de mensagens de controle no NCS. Essa definição dinâmica na transmissão de

mensagens no NCS promove, consequentemente, uma redução na taxa de ocupação da rede CAN, minimizando possíveis efeitos degenerativos no NCS e ao mesmo tempo garantindo os requisitos de controle e estabilidade requeridos.

Outro diferencial da estratégia de controle adaptativo do PA desenvolvida neste trabalho em relação às abordagens que utilizam técnicas de definição dinâmica do período de execução do controlador disponíveis na literatura está na adaptação do controlador PID. O controlador PID discreto que compõe a estratégia, mostrada na Figura 64, utiliza a mesma estrutura do controlador PID definido para NCS e apresentado anteriormente (na seção 5.3.4). Portanto, a estratégia de controle desenvolvida além de automaticamente (de acordo com a resposta do NCS) adaptar o PA (h) do NCS, alterando o período de execução do controlador, adicionalmente também adapta o controlador PID aplicado, a partir do momento que os parâmetros P,I e D do controlador são dependentes do período de amostragem (h) utilizado.

O princípio de operação desta estratégia de controle adaptativo desenvolvida é apresentado na Figura 63.

Figura 63. Estrutura de Aplicação da Estratégia de Controle Adaptativo para NCS CAN

De acordo com a Figura 63, o usuário necessita definir três parâmetros para a aplicação da estratégia:

- ✓ Período de amostragem (PA) máximo (ms): parâmetro que define o valor máximo (ou mais lento) de amostragem que poderá ser utilizado no NCS;
- ✓ Passo de alteração do PA (ms): parâmetro que define quão rápido as alterações na amostragem do NCS (tornando-o mais lento ou mais rápido) serão realizadas durante a operação do NCS;
- ✓ Erro de saída (%): parâmetro que determina a área de operação da estratégia de controle adaptativa em relação ao valor final de saída desejado para o NCS. É importante perceber que esse parâmetro é totalmente diferente do valor de variação aceitável do erro em regime permanente para fins de cálculo do tempo de acomodação (normalmente 3% ou 5%).

Conforme Figura 63, durante a operação do NCS, o valor padrão definido para o PA do NCS (geralmente utilizado em aplicações com PA fixo) é utilizado como valor inicial do PA para a estratégia de controle adaptativo. Esse valor inicial do PA é mais rápido de forma a se obter uma boa resposta transitória para o sistema. Quando a saída do NCS se encontra dentro da área determinada pelo erro de saída ou dentro da área de operação da estratégia, conforme mostrada na Figura 63, o controlador automaticamente diminui o PA do NCS (aumentando seu valor e tornando-o mais lento), mantendo o desempenho do sistema e consequentemente diminuindo a ocupação da rede CAN utilizada. Se em determinado momento de operação, a saída do NCS estiver fora da região determinada (de operação da estratégia), o controlador realiza ação contrária, reinicializando e aumentando o PA do NCS para seu valor inicial (diminuindo o valor e tornando-o mais rápido) (GODOY; PORTO & INAMASU, 2010b).

A Figura 64 apresenta um diagrama explicando o ciclo de operação e implementação da estratégia de controle adaptativo desenvolvida. Essa definição dinâmica do período de amostragem (PA) do NCS é implementada através da transmissão de mensagens do controlador para o sensor (localizado na ECU).

Figura 64. Ciclo de Operação e Implementação da Estratégia de Controle Adaptativo para NCS CAN

Conforme pode ser visto na Figura 64, o controlador do NCS possui dois blocos distintos, um executando a estratégia de controle adaptativo e outro executando o controlador PID discreto definido para NCS. Utilizando-se das informações da variável de processo, provenientes das mensagens CAN do sensor, a estratégia de controle adaptativo calcula os novos valores do PA do NCS. Esses novos valores do PA são então utilizados pelo controlador PID para o cálculo dos sinais de controle a serem transmitidos para o atuador. Em cada mensagem CAN de controle enviada pelo controlador, são transmitidos agora o dado do novo valor do PA para o sensor, juntamente com o dado do sinal de controle para o atuador. O sensor, ao receber um novo valor de PA, começa a partir daquele momento a transmitir as informações da variável de processo de acordo com esse novo período de amostragem (PA) definido, reiniciando o ciclo de operação.

5.5. TÉCNICA DE MODELAGEM E IDENTIFICAÇÃO DE NCS

No desenvolvimento de NCS, é importante conhecer o modelo matemático que descreve a dinâmica do NCS de forma a auxiliar no projeto de estratégias de controle. Essas estratégias podem ser baseadas em simples controladores como PID ou em algoritmos mais complexos como o controle baseado em modelo (MBC – *Model Based Control*) e o controle preditivo baseado em modelo (MPC – *Model based Predictive Control*). Esses algoritmos mais complexos utilizam um modelo matemático da planta do NCS para lidar com os problemas de perda de informações na rede e/ou para calcular sinais de controle aperfeiçoados para lidar com os atrasos de comunicação. (ESTRADA & ANTSAKLIS, 2009). Um exemplo disso é a utilização do estado atual desse modelo para cálculo de um sinal de controle para o NCS quando informações do sensor (realimentação) não estão presentes ou para o desenvolvimento de controladores baseados em técnicas *multi-rate*, na qual o período de execução do controlador é N vezes maior que a do sensor do NCS.

Outra importante aplicação em NCS que tem utilizado desses modelos matemáticos diz respeito a técnicas de diagnóstico de falhas e manutenção preditiva. A execução paralela desses modelos junto aos processos reais fornece informação (sinal de controle e informação do sensor) sobre a operação ideal do NCS, os quais são confrontados com dados reais e usados para diagnóstico de erros e decisão sobre manutenção de equipamentos antes da ocorrência de falhas e problemas. No entanto, quando é difícil ou até mesmo impossível conhecer de forma exata o modelo matemático que descreve a dinâmica do NCS, uma estimação deste modelo pode ser realizada através de dados experimentais.

Visando superar essa dificuldade verificada e auxiliar nesse processo de identificação de NCS, foi desenvolvida uma técnica para a modelagem e validação de modelos matemáticos para NCS (GODOY et al., 2010d). A estrutura desenvolvida neste projeto para realizar essas

tarefas é apresentada na Figura 65 e utiliza uma rede CAN como rede de comunicação industrial utilizada pelo NCS.

Figura 65. Estrutura para Estimação e Validação do Modelo Matemático do NCS Analisado

De acordo com a Figura 65, o processo simulado (modelo matemático definido pelo usuário) é executado em um dispositivo separado do NCS em análise, porém conectado a mesma rede CAN. Esse dispositivo, chamado de módulo identificador, pode ser outro microcomputador (*desktop*) ou um *notebook* (visando maior portabilidade). Esse módulo identificador recebe os mesmos sinais de controle (u) calculados pelo controlador e transmitidos na rede CAN para o processo (ou planta) real do NCS e os aplica no processo simulado. A variável de processo real (y) transmitida na rede CAN pelo sensor do NCS, também é recebida pelo módulo identificador de forma que ambos os dados de saída dos processos (real e simulado) sejam comparados no mesmo instante. Esse procedimento é então realizado e repetido até que o modelo matemático que descreve o NCS seja estimado. Consequentemente, essa situação ocorre quando ambos os resultados dos processos (real e

simulado) forem similares ou iguais, indicando que o modelo matemático estimado pode ser assim validado.

A estrutura para modelagem e estimação de NCS proposta neste projeto e apresentada na Figura 65 possui uma importante vantagem. O funcionamento dessa estrutura utiliza-se da arquitetura totalmente distribuída do NCS, na qual todos os dados necessários estão disponíveis na rede de comunicação. Dessa forma, o processo de identificação é realizado sem a necessidade de conexão de nenhum *hardware* adicional como sensores e sem a necessidade de acesso físico ao processo controlado do NCS.

Outro fator importante é que como o módulo identificador está conectado a mesma rede industrial do NCS em análise, qualquer influência de fatores inerentes ao NCS como atrasos de comunicação, perda de informações transmitidas e período de amostragem irá afetar da mesma forma o processo real e o processo simulado.

Também existe a possibilidade de utilização da estrutura de estimação proposta sem a necessidade de um *hardware* extra (microcomputador, *notebook*) para o módulo identificador. Para esta configuração, onde o módulo identificador é inserido dentro do controlador do NCS, o requisito principal é a necessidade de uma interface com duas (2) portas de comunicação para a rede de comunicação industrial utilizada, no caso a rede CAN. Nessa configuração, o módulo identificador é executado paralelamente ao programa do controlador e utiliza uma (1) porta de comunicação CAN para recepção dos dados do NCS através da rede CAN. A segunda porta (2) CAN é utilizada normalmente pelo controlador do NCS, para envio dos dados do sinal de controle (u) e recepção dos dados da variável de processo (y).

Um índice de desempenho de estimação (I), dado pela Equação(47), foi criado para mensurar o grau de semelhança entre as respostas real e simulada do NCS. Em relação ao índice de desempenho de estimação, geralmente para valores de 0 a 1, quanto menor o seu valor, maior é o grau de semelhança entre as respostas analisadas. Uma vantagem é que esse

índice pode ser utilizado para avaliação do modelo estimado de qualquer tipo de NCS e para qualquer tempo de execução de experimentos, padronizando diferentes análises realizadas.

$$I = \frac{\int_0^T |VP_{real}(t) - VP_{simulada}(t)| dt}{\int_0^T VP_{real}(t) dt} \quad (47)$$

Com os parâmetros: T = período de execução do experimento, VP_{real} = variável de processo real e $VP_{simulada}$ = variável de processo simulada.

O procedimento de estimação consiste na utilização, repetidamente, da estrutura proposta e do índice de desempenho relacionado à comparação entre os resultados até a validação do respectivo modelo matemático do NCS analisado. É importante citar que a estrutura de identificação proposta não representa um algoritmo recursivo de identificação, portanto as alterações no modelo matemático testadas a cada repetição da estrutura são realizadas pelo usuário. O modelo identificado para cada NCS representa o modelo matemático cujo índice de desempenho de estimação foi o menor encontrado entre as repetições realizadas na estrutura proposta.

5.6. CONSIDERAÇÕES FINAIS

Os principais conceitos referentes ao projeto de estratégias de controle para NCS foram discutidos neste capítulo, destacando uma estrutura padrão para o desenvolvimento de NCS com controladores projetados no tempo discreto. Foi observado que um dos grandes desafios relacionados ao desenvolvimento de estratégias de controle para NCS se encontra na sua

aplicabilidade prática. Muitas vezes, as estratégias de controle acabam se tornando complexas e requerendo alto poder de processamento, prejudicando sua aplicação real em NCS.

Buscando lidar com este problema, foi investigada e proposta uma estratégia de controle baseado numa estrutura modificada de um controlador PID discreto. Ao final do capítulo, o desenvolvimento de uma nova estratégia de controle para NCS com redes CAN é apresentado, focando numa técnica de controle adaptativo cujo objetivo é o gerenciamento automático do período de amostragem das mensagens de forma a reduzir a utilização da rede CAN e ao mesmo tempo garantindo os requisitos de desempenho do NCS. Essa nova estratégia de controle adaptativo é discutida, enfatizando suas principais diferenças, vantagens e inovações desenvolvidas em relação às abordagens disponíveis na literatura que usam técnicas definição dinâmica do período de execução do controlador.

Uma simples e eficaz técnica para estimação de modelos matemáticos de NCS foi apresentada. Essa técnica faz uso de um índice de desempenho de estimação que mede o grau de semelhança entre as curvas de resposta real e simulada obtidas. A principal vantagem dessa técnica se encontra na utilização da estrutura de NCS, obtendo todas as informações necessárias para a estimação diretamente através de mensagens disponíveis na rede e sem a necessidade de acesso físico a planta controlada.

6. APLICAÇÕES DA PLATAFORMA DE PESQUISA NO DESENVOLVIMENTO DE NCS COM REDES CAN

6.1. CONSIDERAÇÕES INICIAIS

Neste capítulo são apresentados e discutidos os resultados das aplicações da plataforma de pesquisa construída para o desenvolvimento de NCS com redes CAN. Divididos em subseções, são investigados a realização de tarefas de simulação, análise e controle de NCS com redes CAN.

O projeto dos NCS que compõe a plataforma de pesquisa é realizado utilizando as ferramentas de simulação de NCS avaliadas no capítulo 3. As principais vantagens da utilização dessas ferramentas para o projeto de NCS com redes CAN são discutidas, focando na análise dos parâmetros que exercem influência sobre o desempenho dos NCS. Experimentos realizados demonstram a aplicação das estratégias de controle desenvolvidas para NCS com redes CAN e o desenvolvimento de uma técnica de identificação de NCS. Uma verificação da robustez da estratégia de controle adaptativo é realizada através da avaliação do desempenho dos NCS para diversas configurações e taxas de ocupação da rede CAN.

6.2. PROJETO DE NCS UTILIZANDO FERRAMENTAS DE SIMULAÇÃO

A utilização de ferramentas de simulação e projeto de NCS é necessária para o desenvolvimento da plataforma de NCS, pois permite ao projetista avaliar a dinâmica do

sistema de controle antes de sua implementação real, reduzindo tempo e custos de projeto e desenvolvimento. No entanto, a escolha de qual ferramenta utilizar no projeto de um NCS depende das especificações do sistema e dos resultados e informações requeridas sobre o mesmo (GODOY, PORTO & INAMASU, 2009). Para o projeto da plataforma de NCS proposta neste trabalho, as ferramentas avaliadas, e selecionadas de acordo com a sistematização de informações da seção 3.3, para serem aplicadas foram a *TrueTime* e a *Jitterbug*.

Esse projeto consistiu nas tarefas de análise da influência de parâmetros no desempenho dos NCS e no desempenho global da plataforma, na simulação de operação da plataforma e no desenvolvimento de controladores para todos os NCS da plataforma de forma a se atingir requisitos de controle requeridos. A ferramenta *TrueTime* foi aplicada para analisar os efeitos dos atrasos de comunicação da rede CAN, para obter parâmetros desempenho e para projetar estratégias de controle para todos os NCS que compõe a plataforma. A ferramenta *Jitterbug* foi utilizada para avaliar a sensibilidade dos NCS de acordo com várias condições temporais impostas pela dinâmica de toda a operação da plataforma como presença de atrasos de comunicação, *jitter* e diferentes períodos de amostragem.

Primeiramente o modelo completo da plataforma de NCS foi desenvolvido na ferramenta *TrueTime*. A Figura 66 exemplifica o modelo obtido no ambiente *Matlab/Simulink*. Nesta etapa utilizou-se dos modelos matemáticos de cada NCS levantados anteriormente na seção 4.3. O NCS de controle de movimentação da esteira não foi incluso neste modelo por se tratar de um sistema de controle on/off (e não um sistema contínuo). Sua influência na operação da plataforma de NCS foi incluída através da inserção de uma ECU com capacidade de enviar e receber mensagens na rede CAN, gerando assim tráfego de mensagens não periódico na rede CAN.

Figura 66. Esquemático do Modelo Completo da Plataforma de NCS com Rede CAN na Ferramenta

TrueTime

Conforme pode ser visto na Figura 66, para cada NCS que compõe a plataforma de pesquisa, o modelo na ferramenta consiste de quatro blocos principais. Um bloco na parte de baixo representando o controlador, separado dos três blocos restantes (na parte de cima) que representam o atuador, sensor e planta ou processo.

A partir dos modelos de cada NCS, dado pela função de transferência que representa a dinâmica do sistema, os respectivos controladores puderam ser projetados. Esses controladores levaram em consideração todas as particularidades presentes no NCS como saturação de atuadores, tempo de resposta e períodos de amostragem compatíveis com o *hardware* utilizado. Para os NCS da plataforma, foi utilizada a estratégia de controle PID discreto apresentada na seção 5.3.4.

Nas simulações realizadas foram utilizados os seguintes parâmetros para o controlador PID: valor de $N = 10$ para a constante de filtragem da ação derivativa, $B = 1$ e $C = 0$ para as constantes de ponderação de referência, período de amostragem $h = 100\text{ms}$ e o parâmetro Tt ,

da constante de *Anti-Windup*, igual a $\sqrt{T_i.T_d}$ para controladores PID e igual a T_i para controladores PI (ÅSTRÖM & HÄGGLUND, 1995).

Além desses parâmetros, foi utilizada uma velocidade de transmissão de dados da rede CAN de 250Kbit/s (de acordo com padrões do CAN como DeviceNet, CANOpen, J1939 e ISO11783).

As simulações da plataforma de NCS foram realizadas para obtenção de informações sobre o desempenho de cada NCS e sobre a operação global da plataforma, além do projeto dos controladores que cumprissem os requisitos de controle definidos para a operação de cada NCS da plataforma. Por exemplo, um requisito de controle definido para o NCS de controle de posição de motor seria não apresentar *overshoot*.

Os melhores resultados obtidos através das simulações realizadas para os NCS da plataforma são sintetizados na Tabela 3.

Tabela 3. Informações sobre Controladores Projetados e Desempenho dos NCS

Plataforma de NCS	Parâmetros do Controlador PID			Medidas de Desempenho dos NCS			
	K	Ti	Td	Máximo de pico (%)	Tempo de Pico (s)	Tempo de Subida (s)	Tempo de Estabilização (s)
Controle de Velocidade de Motor	0,07	0,65	0	6,5	0,91	0,52	1,52
Controle de Posição de Motor DC	3	0	0,001	-	-	1,9	2,9
Controle de Temperatura	2	7	0,01	-	-	10,7	13,4
Controle de Nível	10	0,55	0	9,03	21,71	18,48	33,1

Figura 67 até Figura 70 apresentam os resultados finais das simulações dos NCS da plataforma após sintonia dos controladores. Esses gráficos mostram curvas obtidas da resposta a uma entrada degrau e do sinal de controle aplicado na planta para cada um dos NCS da plataforma.

Figura 67. Desempenho do NCS para Controle de Velocidade de Motor obtido por Simulação: Resposta a Degrau e Sinal de Controle

Figura 68. Desempenho do NCS para Controle de Posição de Motor obtido por Simulação: Resposta a Degrau e Sinal de Controle

Figura 69. Desempenho do NCS para Controle de Temperatura obtido por Simulação: (a) Resposta a Degrau e Sinal de Controle

Figura 70. Desempenho do NCS para Controle de Nível obtido por Simulação: Resposta a Degrau e Sinal de Controle

O projeto da plataforma utilizando ferramentas de simulação de NCS fornece um leque de informações sobre cada um dos NCS e também sobre a operação global da plataforma, obtendo-se assim uma base sólida para a implementação real dos NCS e dos controladores projetados na plataforma, de forma a reduzir principalmente tempo de desenvolvimento.

6.3. IMPLEMENTAÇÃO DE CONTROLADORES PID PARA APLICAÇÕES EM NCS

Com o auxílio das informações obtidas com a realização do projeto e simulação da plataforma de NCS, partiu-se para a implementação real dos controladores projetados. A aplicação de controladores PID requer a definição dos parâmetros e ganhos do controlador necessários para atingir os requisitos de operação desejados. De acordo com Eriksson (2008), ainda não foram propostos métodos de sintonia que possam ser aplicados a todos os tipos de NCS, o que é explicado principalmente pela variabilidade imposta ao NCS pela utilização de diferentes redes de comunicação.

Godoy et al. (2010c) estudaram a viabilidade de se aplicar o método de sintonia de controladores PID baseado na curva de resposta do sistema definido por *Ziegler-Nichols* (ZN) para NCS, e após uma parametrização inicial, realizar manualmente uma nova sintonia até a obtenção dos resultados requeridos. Figura 71 apresenta a comparação dos resultados entre as respostas do NCS para controle de velocidade para os métodos de sintonia utilizados.

Figura 71. Desempenho do NCS de Controle de Velocidade para Métodos de Sintonia do PID

A partir da análise do gráfico da Figura 71 percebe-se que o método de sintonia ZN pode ser utilizado para projeto e sintonia do controlador PID discreto definido para aplicação em NCS como base inicial para posterior refinamento manual, conforme realizado para obtenção dos parâmetros finais mostrados na Tabela 3.

Figura 72 até Figura 75 apresentam os resultados finais obtidos para os NCS da plataforma utilizando-se do controlador PID projetado com a ferramenta *TrueTime* (usando os parâmetros definidos na Tabela 3). Esses gráficos mostram as curvas reais obtidas da resposta a uma entrada degrau aplicada na planta para cada um dos NCS da plataforma.

Figura 72. Desempenho do NCS para Controle de Velocidade obtido da Plataforma

Figura 73. Desempenho do NCS para Controle de Posição obtido da Plataforma

Figura 74. Desempenho do NCS para Controle de Temperatura obtido da Plataforma

Figura 75. Desempenho do NCS para Controle de Nível obtido da Plataforma

O controlador PID discreto definido para aplicação em NCS foi testado na plataforma, se mostrando uma solução menos complexa, porém eficiente e de grande versatilidade. Isso foi verificado através da análise dos gráficos que demonstram que o controlador pode ser aplicado para o controle de todos os NCS da plataforma (diferentes tipos de plantas e dinâmicas) apresentando desempenho aceitável em todos os casos (cumprindo os respectivos requisitos de controle).

6.4. VERIFICAÇÃO E VALIDAÇÃO DAS FERRAMENTAS DE SIMULAÇÃO PARA O PROJETO DE NCS COM REDES CAN

O desenvolvimento de ferramentas de simulação de NCS tem crescido juntamente com a grande difusão da tecnologia de NCS. Atualmente, existe uma gama de ferramentas disponíveis na literatura que facilitam a simulação e análise de NCS (TORNGREN et al., 2006; AL-HAMMOURI, BRANICKY & LIBERATORE, 2008). No entanto, é importante verificar a confiabilidade dos dados obtidos com essas ferramentas e assim validar sua aplicação, antes de utilizar dados de simulações em aplicações reais de NCS.

Com o intuito de se avaliar a eficiência das ferramentas selecionadas para utilização neste projeto (*TrueTime* e Jitterbug), foi realizado uma comparação entre resultados simulados obtidos com as ferramentas e dados reais obtidos com a operação do NCS na plataforma (ZANATA, GODOY & PORTO, 2010). A Figura 76 e Figura 77 sintetizam os resultados obtidos para dois NCS da plataforma, indicando que a ferramenta *TrueTime* se mostrou precisa na simulação da rede CAN e do NCS.

Figura 76. Comparação entre os Resultados Simulados na Ferramenta *TrueTime* e Reais Obtidos para o NCS de Controle de Posição utilizando Controlador PD com $K=3$ e $T_d=0,001$

Figura 77. Comparação entre os Resultados Simulados na Ferramenta *TrueTime* e Reais Obtidos para o NCS de Controle de Nível utilizando Controlador PI com $K=10$ e $T_i=0,55$

Nos experimentos realizados visando a validação do uso das ferramentas, foram utilizados o mesmo controlador PID projetado, os mesmos parâmetros do controlador e os mesmos parâmetros de configuração da rede CAN, tanto nas simulações com a ferramenta quanto na implementação real no NCS da plataforma. Analisando o gráfico da Figura 76, que apresenta os resultados do NCS de controle de posição, e o gráfico da Figura 77, que apresenta os resultados do NCS de controle de nível, nota-se claramente a grande aderência entre as curvas de resposta “real” e “simulada” dos NCS. Esta semelhança entre as curvas de resposta deve-se a acuracidade dos modelos matemáticos estimados para os NCS e principalmente ao uso da ferramenta *TrueTime*, que incorpora corretamente as parametrizações e dinâmicas relacionadas a inserção da rede CAN no NCS simulado. A análise dos resultados apresentados permite verificar a confiabilidade dos resultados simulados em relação aos resultados reais e, portanto validar a utilização da ferramenta *TrueTime* no desenvolvimento de NCS com redes CAN (ZANATA, GODOY & PORTO, 2010).

6.5. ANÁLISE DA QUALIDADE DE CONTROLE DE NCS COM REDES CAN

Um dos maiores desafios no desenvolvimento de NCS com rede CAN se encontra no fato de que diversos parâmetros de configuração podem influenciar no desempenho e na estabilidade do NCS (LIAN, MOYNE & TILBURY, 2002). A partir dos dados obtidos de simulações e experimentos realizados na plataforma de pesquisa, foi possível verificar que vários parâmetros podem afetar o desempenho e a estabilidade de um NCS com redes CAN. Entre esses parâmetros estão:

- ✓ velocidade de transmissão de dados da rede;
- ✓ esquema de prioridades das mensagens CAN;
- ✓ atrasos de comunicação da rede;
- ✓ perda de pacotes e mensagens transmitidas na rede CAN
- ✓ período de amostragem das mensagens.

No entanto, o parâmetro que se mostrou mais importante para o projeto de NCS com redes CAN e que relativamente mais afetou a operação dos NCS da plataforma foi o período de amostragem das mensagens.

O gráfico da Figura 78 apresenta resultados que permitem observar tal constatação, num experimento no qual foi alterado o valor do período de amostragem (h), avaliando a influência dessas alterações no desempenho de um NCS da plataforma. Na Figura 78, que sintetiza os resultados do NCS de controle de velocidade, pequenas variações no período de amostragem (relativamente ao valor inicial $h=100\text{ms}$) afetam fortemente o desempenho do NCS, tornando-o mais oscilatório.

Figura 78. Desempenho do NCS de Controle de Velocidade para Diferentes Períodos de Amostragem de Mensagens

De forma a avaliar de forma mais sistemática essa grande influência do período de amostragem de mensagens no desempenho e estabilidade de NCS com redes CAN, a ferramenta *Jitterbug* foi utilizada para calcular um índice de desempenho (função custo quadrática) relacionado com a qualidade de controle do NCS. De acordo com a ferramenta, altos valores para este índice geralmente indicam sistemas mais oscilatórios e menos estáveis, podendo até tornarem-se instáveis. A função custo utilizada foi definida pela soma dos quadrados dos sinais de controle (u) e das saídas da planta (y) durante o tempo de simulação (T) e é dada pela Equação (48).

$$J = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [y^2(t) + u^2(t)] dt \quad (48)$$

Os atrasos de comunicação dos NCS da plataforma foram modelados a partir de duas variáveis randômicas τ_1 e τ_2 , de forma a incorporar corretamente as características não determinísticas de rede CAN no NCS. O atraso total entre a amostragem e a atuação no NCS

é dado então por $\tau_{total} = \tau_1 + \tau_2$. Os modelos das malhas de controle e os controladores PID foram definidos para a aplicação da ferramenta *Jitterbug* conforme simulados e projetados na ferramenta *TrueTime*.

Figura 79 apresenta os resultados das simulações com a ferramenta *Jitterbug* para o NCS de controle de velocidade de motor DC. O gráfico de superfície apresenta valores da função custo definida (eixo z), relacionado à qualidade de controle do NCS, calculados de acordo com diferentes condições de períodos de amostragem (h) (eixo y), e presença de atrasos de comunicação (em porcentagem de h) (eixo x). Um período de amostragem (h) de até 0,5s (até 5 vezes o projetado de 100ms) foi utilizado nas simulações realizadas para o NCS de controle de velocidade de motor DC.

Figura 79. Gráfico obtido com a Utilização da Ferramenta Jitterbug para o NCS de Controle de Velocidade de Motor DC da Plataforma

De acordo com o gráfico da Figura 79 e conforme definido na ferramenta *Jitterbug*, altos valores para o índice de desempenho definido pela função custo indicam sistemas mais oscilatórios e menos estáveis, podendo tornar-se até instáveis. No gráfico da Figura 79, a cor avermelhada representa altos valores da função custo definida enquanto que a cor azulada

representa baixos valores, sendo que a cor da superfície varia gradativamente de acordo com o valor da função custo.

É importante verificar que este tipo de gráfico fornece informações suficientes para o projeto do NCS em questão, apresentando assim o impacto do período de amostragem e do atraso de comunicação sob a qualidade do controle ou a sensibilidade do NCS projetado, incluindo o controlador desenvolvido, para a alteração desses parâmetros.

Para verificar as informações contidas no gráfico obtido com a ferramenta *Jitterbug*, foram feitas diversas simulações com a ferramenta *TrueTime*. Essas diversas simulações cobriram todas as configurações de parâmetros avaliadas e apresentadas no gráfico de superfície mostrado na Figura 79. Essas simulações estão sintetizadas no gráfico da Figura 80, que apresenta e comprova a sensibilidade do NCS projetado em relação a alteração dos parâmetros período de amostragem e atraso de comunicação na rede de comunicação.

De acordo com a Figura 80, para valores de $h = 0,5s$ e $\text{atraso} = 0,5s$ (corresponde a 100% de h), verifica-se que a função-custo apresenta um valor representado na superfície através de uma cor avermelhada. Este fato indica um sistema mais oscilatório e menos estável para o NCS de controle de velocidade (para estes valores de h e atraso de comunicação), que pode ser comprovado pelo gráfico da simulação a resposta ao degrau apresentado que mostra um sistema tendendo a instabilidade.

Ainda de acordo com a Figura 80, para valores de $h=0,5s$ e $\text{atraso} = 0,25s$ (50% de h), verifica-se que o valor da função custo apresenta uma cor verde. Este fato significa que o NCS apresenta um desempenho marginalmente estável, que pode ser verificado através respectivo gráfico de simulação apresentado que mostra um sistema bastante oscilatório, porém estável.

Figura 80. Sensibilidade do NCS para Controle de Velocidade de Motor DC da Plataforma sob Diferentes Condições de Período de Amostragem e Atraso de Comunicação

Finalmente, para valores de $h = 0,1s$ e atraso muito pequeno representados na Figura 80, que foi a configuração inicial utilizada no projeto, verifica-se que a função-custo apresenta uma cor azulada. Este fato significa que o NCS apresenta um desempenho estável, que pode ser verificado através do seu respectivo gráfico de simulação apresentado.

Devido a grande relevância e importância do gráfico final apresentado, que combina os resultados obtidos com simulações realizadas nas ferramentas *TrueTime* e *Jitterbug*, para a análise e projeto de NCS, os mesmos foram obtidos para as outras malhas de controle que compõe a plataforma de NCS com redes CA. A partir desses gráficos podem-se obter as

mesmas informações descritas anteriormente para cada um dos NCS que compõe a plataforma.

Figura 81 apresenta o gráfico final das simulações das ferramentas *TrueTime* e *Jitterbug* para o NCS de controle de posição de motor DC. Para a construção desse gráfico foi utilizado um período de amostragem (h) de até 1s (até 10 vezes o projetado de 100ms) nas simulações realizadas.

Figura 81. Sensibilidade do NCS para Controle de Posição de Motor DC da Plataforma sob Diferentes Condições de Período de Amostragem e Atraso de Comunicação

A partir da análise dos gráficos das Figura 80 e Figura 81, é possível verificar a grande influência do período de amostragem (h) no desempenho dos NCS de controle de velocidade

e posição de motor DC. Pequenas variações nesse parâmetro (relativamente aos valores iniciais projetados de $h = 0,1$ s) tendem a tornar o sistema mais oscilatório. Atrélado a variação do período de amostragem, a variação do atraso de comunicação na rede presente no NCS tende a acentuar essa degeneração do desempenho do NCS, podendo em alguns casos levá-lo a instabilidade.

Figura 82 apresenta o gráfico final das simulações das ferramentas *TrueTime* e *Jitterbug* para o NCS de controle de temperatura. Para a construção desse gráfico foi utilizado um período de amostragem (h) de até 2s (até 20 vezes o projetado de 100ms) nas simulações realizadas.

Figura 82. Sensibilidade do NCS para Controle de Temperatura da Plataforma sob Diferentes Condições de Período de Amostragem e Atraso de Comunicação

Figura 83 apresenta o gráfico final das simulações das ferramentas *TrueTime* e *Jitterbug* para o NCS de controle de nível de reservatório. Para a construção desse gráfico foi utilizado um período de amostragem (h) de até 2s (até 20 vezes o projetado de 100ms) nas simulações realizadas.

Figura 83. Sensibilidade do NCS para Controle de Nível da Plataforma sob Diferentes Condições de Período de Amostragem e Atraso de Comunicação

A partir da análise dos gráficos das Figura 82 e Figura 83, é possível verificar que a influência do período de amostragem (h) no desempenho dos NCS de controle de temperatura e nível de reservatório é bem menor. Para esses NCS, são necessárias grandes variações (relativamente aos valores iniciais projetados de $h = 0,1$ s) nesse parâmetro para tornar o

sistema mais oscilatório. O mesmo pode ser visto em relação ao parâmetro atraso de comunicação da rede. Portanto verifica-se que os NCS de controle de nível e de temperatura são menos sensíveis a variação dos parâmetros período de amostragem (h) e atraso de comunicação na rede.

Assim sendo, a análise dos gráficos apresentados permitiu verificar que mesmo sendo o período de amostragem o parâmetro que mais afeta o desempenho de NCS, sua influência está relacionada com o tipo do sistema. Nos gráficos das Figura 80 e Figura 81, os NCS de controle de velocidade e de controle de posição de motor DC representam sistemas de dinâmica (resposta) rápida. Nesse tipo de sistema a influência do período de amostragem é grande. De acordo com a Figura 80 pequenas variações no valor do período de amostragem, de 0,1 a 0,5 s, afetam fortemente o desempenho do NCS. Essa conclusão também foi verificada para a simulação do NCS de controle de posição (com período de amostragem variando até 1s), mostrada na Figura 81. Consequentemente, nesse tipo de sistema a presença de atrasos de comunicação na rede de comunicação também apresenta grande influência o desempenho do NCS.

No entanto, a influência do período de amostragem em NCS com dinâmica lenta é menor. Este fato é verificado de acordo com a Figura 82, que apresenta os resultados para a simulação do NCS de controle de temperatura com períodos de amostragem de 0,1 a 2s. Os efeitos degenerativos no desempenho do NCS foram menores para maiores variações do período de amostragem. Esta conclusão também pode ser verificada através da análise do gráfico da Figura 83, que apresenta as simulações realizadas para o NCS de controle de nível (com período de amostragem variando até 2s), que também representa um sistema de dinâmica lenta.

6.6. APLICACAO DA TÉCNICA DE MODELAGEM E IDENTIFICAÇÃO DE NCS

A técnica de modelagem e identificação de NCS desenvolvida neste projeto e descrita na seção 5.5 visa fornecer uma solução simples e eficaz para a estimação de modelos matemáticos que representem a dinâmica de operação de um NCS analisado. Para demonstrar a funcionalidade da estrutura de modelagem e estimação desenvolvida para NCS, são apresentados experimentos realizados com os NCS que compõe a plataforma de pesquisa com redes CAN desenvolvida.

Nesses experimentos, inicialmente foram utilizadas informações de *datasheet* e teoria de modelagem clássica de sistemas dinâmicos para obter os modelos matemáticos iniciais (de acordo com seção 4.3) necessários para utilização da estrutura. Os resultados obtidos para os experimentos realizados na plataforma de pesquisa para o NCS de controle de velocidade e para o NCS de controle de nível são discutidos a seguir.

A Tabela 4 sintetiza as informações do experimento de identificação do NCS de controle de velocidade. O modelo matemático clássico obtido a partir de informações de *datasheet* foi inicialmente utilizado para o procedimento de estimação e a partir dele foram feitas atualizações até a obtenção do modelo matemático estimado para o NCS.

Tabela 4. Informação dos Modelos Matemáticos para o NCS de Controle de Velocidade de Motor
Utilizado no Experimento de Identificação

Forma de Obtenção	Modelo do NCS - Função de Transferência
Obtido a partir de informações de <i>datasheet</i> e modelagem clássica	$\frac{w_m}{V}(s) = \frac{0,0886}{0,007982s + 0,008144}$
Estimado através da estrutura de identificação desenvolvida	$\frac{w_m}{V}(s) = \frac{0,085}{0,003982s + 0,008144} e^{-0,5s}$

De forma a verificar a exatidão do modelo matemático estimado para o NCS de controle de velocidade, são apresentados experimentos com o mesmo operando em malha aberta e em malha fechada com controlador PID discreto ($K = 0,07$ e $T_i = 0,65$). Os resultados, mostrados na Figura 84 e Figura 85, apresentam uma comparação entre as saídas obtidas para o modelo simulado e o processo real controlado.

Visualizando os gráficos da Figura 84 é possível observar a grande aderência entre as curvas de resposta em malha aberta do modelo simulado e do processo real. Isso é comprovado pelos baixos valores do índice de desempenho obtido para estes casos, respectivamente $I = 0,0069$ e $I = 0,0074$.

Figura 84. Comparação entre os Resultados Reais e Simulados para Experimentos de Estimação do Modelo do NCS para Controle de Velocidade em Malha Aberta: (a) Degrau de 6V, (b) Degrau de 9V

A grande aderência entre as curvas de resposta do modelo simulado e do processo real também pode ser verificado pela análise da Figura 85, que apresenta os resultados para o NCS operando em malha fechada. O índice de desempenho (I) calculado para o experimento da Figura 85 utilizando o modelo estimado final foi $I = 0,0082$, ou seja, a resposta do modelo estimado para o NCS difere em 0,82% da resposta real do NCS obtida da plataforma. Essa

pequena diferença enfatiza a eficiência da estrutura de identificação de NCS desenvolvida neste projeto.

Figura 85. Comparação entre os Resultados Reais e Simulados para o Experimento de Estimação do Modelo do NCS para Controle de Velocidade em Malha Fechada com Controlador PI

A Tabela 5 sintetiza as informações utilizadas no experimento de identificação do NCS de controle de nível. Seguindo os passos descritos, o modelo matemático clássico obtido a partir de informações de *datasheet* foi inicialmente utilizado para o procedimento de estimação e a partir dele foram feitas atualizações até a obtenção do modelo matemático estimado para o NCS.

No entanto, como o NCS de controle de nível representa um sistema não linear, seu modelo matemático é linearizado de acordo com o ponto de operação (nível h em cm) escolhido e seu modelo matemático tem de ser, portanto obtido para cada ponto de operação simulado.

Tabela 5. Informação dos Modelos Matemáticos para o NCS de Controle de Nível Utilizado no Experimento de Identificação

Forma de Obtenção	Modelo do NCS	Função de Transferência –Exemplo para Operação no Ponto h = 9 cm
Obtido a partir de informações de datasheet e modelagem clássica	$\frac{h(s)}{q_i(s)} = \frac{R/2}{A.R.s/2 + 1}$	$\frac{h(s)}{q_i(s)} = \frac{0,375}{67,3.s + 1}$
Estimado através da estrutura de identificação desenvolvida	$\frac{h(s)}{q_i(s)} = \frac{R/2}{1,05.A.R.s/2}$	$\frac{h(s)}{q_i(s)} = \frac{0,375}{70,5.s + 1}$

De forma a verificar a exatidão do modelo matemático não linear estimado para o NCS de controle de nível foi realizado um experimento com o NCS operando com diferentes *setpoint* e com o controlador PID discreto ($K = 10$ e $T_i = 0,55$). A comparação entre as saídas obtidas para o modelo simulado e para o processo real controlado pode ser visualizada na Figura 86.

Figura 86. Comparação entre os Resultados Reais e Simulados para o Experimento de Estimação do Modelo do NCS para Controle de Nível em Malha Fechada com Controlador PI

Analisando o gráfico da Figura 86, novamente pode-se verificar grande aderência entre as curvas de resposta real e simulada do NCS de controle de nível. O valor do índice de desempenho (I) calculado foi de $I = 0,0182$, o qual representa um pequeno valor, ainda que um pouco maior que o obtido para o NCS de controle de velocidade ($I = 0,0082$). No entanto, essa diferença pode ser explicada pelo impacto no modelo estimado da não linearidade do NCS de controle de nível.

Portanto, a grande similaridade entre os resultados obtidos do processo real e do modelo simulado, mostrados na Figura 85 e Figura 86, permite validar os modelos matemáticos estimados para o NCS de controle de velocidade e para o NCS de controle de nível. Adicionalmente, os resultados apresentados confirmam a viabilidade de aplicação da estrutura desenvolvida para modelagem e identificação de NCS com redes CAN (GODOY et al., 2010d).

6.7. APLICAÇÃO DA ESTRATÉGIA DE CONTROLE ADAPTATIVO DO PERÍODO DE AMOSTRAGEM

A estratégia de controle adaptativo do período de amostragem desenvolvida neste projeto e descrita na seção 5.4 visa fornecer uma solução para o controle de NCS que consiga superar automaticamente o problema da definição do período de amostragem das mensagens e seu efeito sobre o desempenho do NCS. Para a avaliação do controlador adaptativo desenvolvido em termos de aplicabilidade, versatilidade e robustez para diferentes configurações de NCS, foi realizada uma série de experimentos utilizando-se da plataforma de pesquisa de NCS com redes CAN. A estratégia de controle adaptativo será chamada nesta seção de ECA para fins de simplificação.

Em todos os testes de operação realizados, os cinco NCS da plataforma de pesquisa foram mantidos em operação, de forma a se obter uma situação com um tráfego de mensagens (periódico e não periódico) significativo na rede CAN, condizente com a realidade de aplicações industriais em rede. Um período de amostragem (PA) inicial de 10ms foi configurado para os sensores de quatro dos NCS, sendo que o NCS de controle da esteira opera baseado em eventos (presença de peças).

Os parâmetros configurados para a ECA e para o controlador PID discreto utilizados em cada um dos NCS da plataforma de pesquisa são apresentados na Tabela 6. Os parâmetros $N=10$, $B=1$ e $C=0$ foram definidos para todos os NCS e utilizados nos experimentos.

Tabela 6. Resumo dos Parâmetros Utilizados nos NCS da Plataforma de Pesquisa para Experimentos de Aplicação da Estratégia de Controle Adaptativo do Período de Amostragem (PA)

NCS	Velocidade	Posição	Nível	Temperatura
<i>Parâmetros da Estratégia de Controle Adaptativo</i>				
PA Máximo (ms)	200	200	400	800
Passo de Alteração do PA (ms)	5	5	5	10
Erro de Saída (%)	10	5	5	10
<i>Parâmetros do Controlador PID Discreto</i>				
Ganho Proporcional - K	0.07	3	10	15
Tempo Integral – Ti	0,65	-	0,55	7,5
Tempo Derivativo - Td	-	0,001	0	0,1

Inicialmente, de forma a explicar melhor os gráficos que serão apresentados e utilizados posteriormente, foi realizado um experimento onde a estratégia de controle adaptativo (ECA) desenvolvida foi aplicada somente para o NCS de controle de velocidade. Dessa forma, consegue-se demonstrar nitidamente os resultados obtidos com a aplicação do controlador adaptativo através da comparação direta entre a variação do período de amostragem (PA) do NCS de controle de velocidade e a variação da taxa de ocupação da rede CAN. Esses primeiros resultados podem ser vistos na Figura 87.

Figura 87. Estratégia de Controle Adaptativo no NCS de Controle de Velocidade: Resposta ao Degrau e Variações do Período de Amostragem e da Ocupação da Rede CAN durante a Operação do NCS

É importante entender que a ECA começa a operar automaticamente com uma mudança no *setpoint* do NCS. A Figura 87 apresenta, no gráfico superior, o desempenho de saída do NCS durante a operação da ECA e, no gráfico inferior, as variações do período de amostragem (PA) e da taxa de ocupação da rede CAN obtidas. De acordo com a Figura 87, a ocupação da rede CAN alcançava inicialmente quase 21%. A partir do momento em que a ECA desenvolvida começou a operar, o PA do NCS de controle de velocidade foi automaticamente incrementado até o valor de 200ms. Com esse incremento do PA do NCS, a taxa de ocupação da rede CAN foi diminuída para aproximadamente 12%, enquanto que o desempenho de saída do NCS era mantido constante, conforme pode ser visto no gráfico superior da Figura 87 que mostra a resposta ao degrau do NCS de controle de velocidade durante a operação da ECA.

Com a aplicação da ECA para mais de um NCS ao mesmo tempo, o gráfico referente à variação da taxa de ocupação da rede CAN irá mostrar as alterações ocorridas durante o tempo do experimento. Assim, teremos que cada alteração ocorrida estará relacionada ao momento em que o controlador adaptativo começou a operar em um determinado NCS da plataforma, automaticamente gerenciando o período de amostragem (PA) desse NCS. Isso será importante para visualização dos resultados apresentados a partir desse ponto.

De forma a avaliar o desempenho e a versatilidade de aplicação da estratégia de controle adaptativo (ECA) para diferentes tipos de NCS, foi realizado um experimento com a aplicação simultânea da mesma para os quatro NCS da plataforma de pesquisa. Os parâmetros utilizados nesse experimento foram os descritos na Tabela 6. Os resultados obtidos são apresentados de forma conjunta na Figura 88 e demonstram a eficácia da ECA desenvolvida através de uma significativa redução na taxa de ocupação da rede CAN.

Figura 88. Aplicação Simultânea da Estratégia de Controle Adaptativo nos NCS da Plataforma: Respostas ao Degrau dos NCS e Variações dos Períodos de Amostragem e da Ocupação da Rede CAN

A Figura 88 apresenta para os quatro NCS da plataforma, um gráfico com o desempenho de controle do NCS a uma entrada degrau (gráficos superiores), um gráfico demonstrando a variação do período de amostragem (PA) realizada pela ECA (gráficos intermediários) e mais um gráfico (inferior maior) mostrando a variação da ocupação total da rede CAN durante a execução do experimento. Verifica-se através da análise dos gráficos das respostas ao degrau da Figura 88, a manutenção do bom desempenho de controle dos NCS (em relação a resultados obtidos anteriormente como na seção 6.3), os quais não são afetados pela alteração de seu PA. No entanto, essas alterações do PA dos NCS resultam num resultado ainda mais impactante e significativo, a diminuição da taxa de utilização da rede CAN de aproximadamente 37% para 2%, sem sacrificar o desempenho de controle ou a confiabilidade dos NCS.

Observa-se no gráfico da taxa de ocupação da rede CAN durante o experimento da Figura 88, quatro grandes alterações na taxa de utilização da rede. Conforme explicado, essas quatro alterações estão relacionadas aos instantes em que a ECA começa a gerenciar o PA daquele respectivo NCS:

- ✓ Instante $t=1,5s$ para o NCS de controle de velocidade;
- ✓ Instante $t=2,5s$ para o NCS de controle de posição;
- ✓ Instante $t=37s$ para o NCS de controle de temperatura;
- ✓ Instante $t=40s$ para o NCS de controle de nível.

A Figura 89 apresenta uma versão amplificada do gráfico do NCS de controle temperatura da Figura 88 e demonstra uma situação relacionada à operação da ECA. Observando o gráfico de alteração do PA da Figura 89, verifica-se que após o início do gerenciamento do PA (tornando-o mais lento) no instante $t=37s$, ocorre aproximadamente no instante $t=38s$, a reinicialização do PA para seu valor inicial. De acordo com a ECA, isso

ocorre devido ao valor da saída do NCS atingir valores fora do erro ou da área de operação definidos para o controlador. Nesse instante ($t=38s$) verifica-se um aumento da taxa de ocupação da rede CAN relacionada ao PA mais rápido definido para o NCS. Após esse instante, o controlador adaptativo novamente começa a gerenciar o PA, aumentando seu valor e tornando-o mais lento até o PA máximo de $0,8s$ definido previamente.

Figura 89. Versão Ampliada dos Resultados da Figura 88 para o NCS de Controle de Temperatura:

Situação de Reinicialização do PA do NCS

A Figura 90 apresenta uma comparação entre o desempenho de controle obtido para o NCS de controle de velocidade utilizando diferentes valores de PA fixos e com a aplicação da ECA implementada. O objetivo deste gráfico é verificar através das curvas a diferença de resultados entre as abordagens de PA fixo e variável e se existe vantagem na utilização da abordagem variável implementada (ECA).

Figura 90. Comparação do Desempenho de Controle do NCS de Velocidade Utilizando Período de Amostragem (PA) Constante e Variável (ECA)

Analisando as curvas da Figura 90, é possível verificar que não existe desvantagem ou degradação de desempenho com a aplicação da ECA em relação à utilização de PA constantes. Pelo contrário, uma pequena melhoria no tempo de subida do NCS pode ser notada para os casos onde foi usado o controle adaptativo com PA máximo de 200ms, principalmente quando foi configurado o passo de alteração do PA = 1ms. Os resultados de comparação para os outros NCS forneceram a mesma conclusão verificada, no entanto os mesmos foram omitidos como forma de simplificação.

Os resultados obtidos e apresentados anteriormente provam a eficácia da estratégia de controle adaptativo (ECA) para aplicação em NCS com redes CAN, além de demonstrar sua aplicabilidade prática e a grande versatilidade de utilização para diferentes tipos de NCS.

A aplicação da ECA é importante pelo fato que geralmente em NCS as características da rede de comunicação como, por exemplo, os atrasos (*delays*) de comunicação e de controle, são variantes no tempo e dependentes de parâmetros como o tráfego de mensagens e o número de dispositivos conectados na rede, entre outros. Por causa disso, NCS operando

com períodos de amostragem muito altos (ou lentos) podem em algum momento ter seu desempenho deteriorado, e assim não cumprirem seus requisitos de controle e se tornarem mais oscilatórios. Por outro lado, se o PA do NCS é muito pequeno (ou rápido), a rede de comunicação apresenta alta taxa de ocupação e fica sobrecarregada, induzindo maiores atrasos (tempo de espera para contenção de mensagens) e até mesmo impedindo a conexão de mais dispositivos na rede.

Para demonstrar a importância de aplicação da ECA foi realizado um experimento no qual foi configurado um PA fixo de 10ms (rápido) para todos os NCS da plataforma e avaliados seus respectivos desempenhos para essa situação de maior ocupação da rede CAN (37%). Os resultados são apresentados na Figura 91 e demonstram que uma maior taxa de utilização da rede CAN (e consequentemente maiores atrasos) pode deteriorar o desempenho dos NCS com PA fixo. Isso é comprovado no gráfico do NCS de controle de nível, operando nestas condições, se torna mais oscilatório e tende a instabilidade após o instante $t=80s$.

Figura 91. Desempenho de Controle dos NCS da Plataforma Obtidos Utilizando PA=10ms (Ocupação Total da Rede CAN de 37%)

Posteriormente, de forma a avaliar o desempenho dos NCS para operação numa situação extrema, um microcomputador foi conectado a rede CAN da plataforma e configurado para enviar, repetidamente, mensagens extras na rede a uma taxa que elevasse a ocupação da rede para valores máximos. Conforme medido no experimento, essas mensagens extras originavam uma ocupação de aproximadamente 55% da rede CAN, levando a uma ocupação total de aproximadamente 92% para a realização do experimento. Os resultados desse experimento com ocupação máxima da rede CAN são apresentados na Figura 92.

Figura 92. Desempenho dos NCS da Plataforma Obtidos Utilizando PA=10ms e Ocupação Extra da Rede CAN (Ocupação Total de 92%)

Conforme pode ser observado na Figura 92, que mostram os resultados do experimento com ocupação máxima (92%) da rede CAN, a degradação do desempenho de controle dos NCS é mais acentuada neste caso de maior ocupação da rede CAN. Isso pode ser explicado por causa dos maiores atrasos, originados principalmente pelo tempo de bloqueio de acesso a rede CAN causado pela contenção de mensagens. De acordo com os gráficos, verifica-se uma

deterioração do desempenho dos NCS de controle de velocidade e de temperatura, que não haviam sido afetados no experimento anterior. O NCS de controle de nível, da mesma forma que no experimento anterior, torna-se mais oscilatório e tende a instabilidade.

Para avaliar a robustez de aplicação da estratégia de controle adaptativo (ECA) desenvolvida, o mesmo experimento com ocupação máxima de 92% da rede CAN foi utilizado. Este experimento representaria o pior caso de aplicação da ECA. No entanto nesse experimento, a ECA foi aplicada buscando automaticamente gerenciar os PA dos NCS e garantir requisitos de desempenho requeridos para os NCS. Os resultados do experimento de pior caso da ECA (com ocupação máxima da rede CAN) são apresentados na Figura 93.

Figura 93. Resultados do Experimento de Pior Caso de Aplicação da Estratégia de Controle Adaptativo nos NCS da Plataforma: Respostas ao Degrau dos NCS e Variações dos Períodos de Amostragem e da Ocupação da Rede CAN

Observando-se os gráficos da Figura 93, é possível verificar que com a aplicação da ECA consegue-se alcançar os desempenhos de controle requeridos para os NCS da

plataforma, superando a influência dos efeitos da rede CAN operando com ocupação máxima inicial de 92%. Esses resultados demonstram a robustez de aplicação da ECA, mesmo para o pior caso de aplicação considerando a rede CAN. De acordo com o gráfico da variação da ocupação da rede CAN, o gerenciamento dos PA dos NCS (que pode ser visto nos gráficos de alteração dos PA) acarretou uma diminuição da ocupação da rede CAN de 92% para 57% (representando os 55% de tráfego extra, mais 2% referente ao tráfego de controle dos NCS).

Para facilitar a discussão e a avaliação dos resultados obtidos com esses experimentos realizados, foram implementados gráficos de comparação das curvas de saída obtidas em todos os experimentos para cada um dos NCS da plataforma. Esses gráficos podem ser visualizados nas Figura 94 até Figura 97. Para os gráficos mostrados, os termos “com” ou “sem ocupação extra” referem-se ao experimento no qual foi inserido tráfego extra de mensagens representando uma ocupação de 55% da rede CAN. De forma a permitir uma melhor visualização das curvas, as figuras apresentam um gráfico (A) que representa uma ampliação da região (A) delimitada no gráfico principal de comparação.

A Figura 94 apresenta os resultados da comparação de desempenho para o NCS de controle de velocidade.

Figura 94. Comparação de Desempenho do NCS de Velocidade e Avaliação de Aplicação da ECA

Analisando a Figura 94, verifica-se conforme citado anteriormente, uma pequena degradação do desempenho do NCS, tornando-o mais oscilatório, quando da utilização do PA fixo e com aumento da ocupação da rede CAN. No entanto, tais problemas são superados com a aplicação da ECA.

A Figura 95 apresenta os resultados da comparação de desempenho para o NCS de controle de posição. É importante notar que os resultados obtidos para o NCS de posição diferenciam-se dos obtidos para os outros NCS. Conforme pode ser visto no gráfico, o NCS não apresentou significativa degradação de desempenho relacionada a maior ocupação da rede CAN. Acredita-se que isso esteja relacionado ao fato de esse NCS ter suas mensagens CAN configuradas com maior prioridade de acesso a rede, o que significa menores atrasos. A configuração de prioridades utilizada neste trabalho foi: (consecutivamente da maior para menor) NCS de Posição, de Velocidade, de Temperatura, de Nível e da Esteira. Outro fator interessante diz respeito à aplicação da ECA para este NCS. Observado o gráfico com a versão ampliada das curvas, verifica-se que a aplicação da ECA acarreta num maior tempo de subida para o NCS.

Figura 95. Comparação de Desempenho do NCS de Posição e Avaliação de Aplicação da ECA

As Figura 96 e Figura 97 apresentam os resultados da comparação de desempenho para os NCS de controle de temperatura e controle de nível.

Figura 96. Comparação de Desempenho do NCS de Temperatura e Avaliação de Aplicação da ECA

Figura 97. Comparação de Desempenho do NCS de Nível e Avaliação de Aplicação da ECA

Também conforme discutido anteriormente, observa-se nas Figura 96 e Figura 97 uma maior deterioração do desempenho de controle desses NCS, o que acarreta altas oscilações para o NCS de controle de nível, levando-o a instabilidade.

Os resultados obtidos, e discutidos nas Figura 93 até Figura 97, comprovam a viabilidade e os benefícios de aplicação da estratégia de controle adaptativo (ECA) desenvolvida. Adicionalmente, os resultados permitem confirmar a versatilidade e robustez da ECA utilizadas em NCS com redes CAN. Com a utilização da ECA em NCS com redes CAN, conseguiu-se diminuir a ocupação da rede CAN durante a operação dos NCS, mitigando possíveis problemas com atrasos de comunicação e perdas de informações transmitidas na rede que podem afetar o desempenho dos NCS conectados.

Outra vantagem importante da ECA é sua total compatibilidade com sistemas industriais já implantados. Novos NCS podem ser desenvolvidos utilizando a ECA em sistemas que utilizam NCS com período de amostragem fixo ou até em sistemas distribuídos em rede que não possuem soluções baseadas na tecnologia de NCS.

6.8. CONSIDERAÇÕES FINAIS

É importante citar que diversos outros experimentos, que apesar não terem sido alvo de estudos neste trabalho, mas que representam outros focos de pesquisa em NCS, também são capazes de serem realizados na plataforma de pesquisa sobre NCS com redes CAN desenvolvida. Entre esses se podem citar a avaliação do desempenho do NCS sob a influência da perda de mensagens na rede CAN, da variação da taxa de utilização do processador do controlador do NCS e do esquema de prioridades de mensagens da rede CAN.

Este capítulo apresentou uma série de experimentos realizados na plataforma de pesquisa de NCS que permitiu consolidar novos conhecimentos adquiridos sobre o desenvolvimento de NCS com redes CAN e validar novos e inovadores desenvolvimentos para a área de NCS. Os resultados apresentam a avaliação dos fatores que afetam o

desempenho de NCS com redes CAN e permitem concluir que o período de amostragem é o parâmetro que mais afeta o desempenho desses sistemas. Uma técnica de identificação de NCS, que utiliza das vantagens da conexão totalmente distribuída de NCS, foi desenvolvida e apresentou bons resultados permitindo a estimação de modelos matemáticos de NCS com redes CAN. Uma nova estratégia de controle para NCS que busca lidar com os problemas da correta definição do período de amostragem das mensagens através de uma abordagem adaptativo foi proposta. A versatilidade e robustez dessa nova estratégia de controle adaptativo foram comprovadas através da avaliação do desempenho dos NCS para diversas configurações e taxas de ocupação da rede CAN.

7. CONCLUSÕES

Este trabalho de doutorado contribuiu com o projeto e o desenvolvimento de NCS baseado no uso da simulação e na proposta de uma estratégia de controle que possibilitou contornar os efeitos degenerativos causados por fatores como o período de amostragem e permitiu garantir requisitos de desempenho e estabilidade para o NCS. As ideias e propostas apresentadas neste trabalho não estão restritas à aplicação em NCS com redes CAN, podendo ser utilizadas para o desenvolvimento de NCS baseados em outros tipos de redes de comunicação industrial.

O projeto e o desenvolvimento de um NCS é essencialmente um problema multidisciplinar. Por exemplo, definições feitas no projeto da rede de comunicação afetam o projeto do controle do NCS e vice-versa. Por esta razão, a utilização de ferramentas que forneçam suporte a simulação de NCS se torna muito interessante, senão indispensável. Diante dessa constatação, uma revisão geral sobre o desenvolvimento e a utilização de ferramentas de análise, simulação e projeto de NCS foi realizada. A partir dessa revisão, foram sistematizadas, na forma de um estudo comparativo, as informações pesquisadas sobre características, requisitos de *software*, vantagens e usabilidade de diversas ferramentas disponíveis na literatura.

A análise desse estudo comparativo e qualitativo das ferramentas possibilitou avaliar e selecionar as ferramentas *TrueTime* e *Jitterbug* como as melhores opções para utilização neste trabalho. De acordo com esse levantamento, entre as ferramentas analisadas, a *TrueTime* foi a melhor avaliada para utilização na simulação e projeto de NCS. Essa conclusão foi obtida diante de características como possibilitar o desenvolvimento total (modelo da planta, projeto do controlador e definição de parâmetros da rede), ser de fácil utilização e apresentar grande

respaldo e aplicação no meio acadêmico. Uma comparação entre resultados teóricos, obtidos através de simulação com a ferramenta *TrueTime*, e resultados reais coletados de experimentos utilizando NCS com redes CAN permitiu validar o uso das ferramentas para a análise e projeto de NCS realizada.

Para suportar o teste e a verificação das estratégias e tarefas desenvolvidas neste trabalho, foi proposto o projeto de uma plataforma de pesquisa sobre NCS com redes CAN. O grande benefício dessa plataforma de NCS foi fornecer conhecimento (vantagens, problemas e formas de implementação) sobre o desenvolvimento de NCS com redes CAN, enfatizando os principais conceitos desta tecnologia de sistemas de controle distribuído. A funcionalidade e versatilidade da plataforma foram comprovadas através da realização de tarefas de análise, modelagem, simulação e controle de NCS com redes CAN, permitindo analisar os benefícios da estratégia de controle e da técnica de identificação desenvolvidas e fornecendo dados para validação de resultados teóricos obtidos através de simulações.

A arquitetura desenvolvida para a plataforma forneceu grande flexibilidade para o desenvolvimento de NCS, permitindo que experimentos fossem conduzidos para diferentes configurações da plataforma e sob diferentes parâmetros de configuração. Considerando-se a implementação de *hardware* definida, o desenvolvimento de novos NCS se torna mais simples, fácil e de baixo custo. A utilização da teoria de máquinas de estado na implementação de *software* facilitou o desenvolvimento de código e forneceu uma configuração bastante flexível e expansível para os NCS. Essa configuração possibilita a expansão do controlador (adição de mais NCS), a inserção de novos algoritmos de controle para serem testados em NCS e a replicação do código desenvolvido para utilização em outros controladores (*desktop*) da plataforma.

As ferramentas de simulação de NCS estudadas e selecionadas foram aplicadas para o projeto e simulação da plataforma de pesquisa de NCS proposta. A ferramenta *TrueTime* foi

utilizada para analisar os efeitos dos atrasos de comunicação da rede CAN e avaliar a influência de parâmetros de projeto como período de amostragem no desempenho dos NCS. A aplicação dessa ferramenta permitiu projetar controladores PID em tempo discreto de acordo com os requisitos reais de implementação da plataforma como: velocidade de transmissão de dados da rede e período de amostragem dos *hardwares* utilizados nas ECUs das malhas de controle da plataforma. A utilização da ferramenta *Jitterbug* no projeto da plataforma permitiu evidenciar que o período de amostragem das mensagens é um dos principais parâmetros que afetam o desempenho de NCS. Permitiu também verificar a resposta dos NCS de acordo com a operação sob diferentes períodos de amostragem e atrasos de comunicação, o que acabou resultando na constatação de que a influência do período de amostragem depende da dinâmica do NCS. NCS com dinâmica mais rápida são relativamente mais afetados por variações no período de amostragem enquanto que NCS com dinâmica lenta são menos relativamente afetados. Portanto a utilização das ferramentas auxiliou no projeto da plataforma de pesquisa de NCS desenvolvida, permitindo simular a operação global da plataforma antes de sua implementação e reduzindo principalmente tempo de desenvolvimento.

Outra constatação observada foi a dificuldade de se obter modelos de NCS, que representassem de forma adequada a dinâmica do sistema, para aplicações em desenvolvimentos recentes na área. Nessas soluções, um modelo do NCS é utilizado em estratégias de controle baseadas em modelo e diagnóstico de falhas que usam informações provenientes do modelo matemático para utilização no processo real. Para superar esse problema, um método de modelagem e estimação de NCS com redes CAN foi proposto e desenvolvido. O funcionamento desse método demonstrou uma importante vantagem, relacionada à utilização da arquitetura totalmente distribuída do NCS, na qual todos os dados necessários estão disponíveis na rede de comunicação. Com a utilização desse método, o

processo de estimação e obtenção do modelo é realizado sem a necessidade de conexão de *hardware* adicional e sem a necessidade de acesso físico ao processo controlado do NCS. Os resultados obtidos comprovam a eficácia do método através da estimação de modelos matemáticos precisos para os NCS da plataforma.

Controladores PID baseados em tempo discreto para aplicação em NCS foram projetados e testados na plataforma, se mostrando uma solução menos complexa e bastante versátil para cumprir os requisitos de controle de todos os NCS implementados na plataforma. A estrutura definida para esse controlador foi uma solução que unia algumas modificações bastante utilizadas em controladores PID industriais como *Anti-Windup*. Essa mesma estrutura foi utilizada também no desenvolvimento de uma nova estratégia de controle para NCS com redes CAN.

O desenvolvimento dessa nova estratégia foi norteado pela constatação neste trabalho de que o período de amostragem das mensagens representava o principal fator que afetava o desempenho e a estabilidade de um NCS com redes CAN. De forma a lidar com este problema, foi então desenvolvida uma estratégia de controle adaptativo (ECA) que gerenciava automaticamente o período de amostragem do NCS, reduzindo a ocupação da rede CAN e garantindo requisitos de desempenho e estabilidade do sistema. Com a aplicação dessa estratégia na plataforma de pesquisa de NCS, pode-se diminuir consideravelmente a ocupação da rede CAN durante a operação dos NCS, diminuindo possíveis problemas com atrasos de comunicação e perdas de informações transmitidas na rede que poderiam afetar o desempenho dos NCS conectados. A versatilidade de utilização da ECA desenvolvida foi verificada através da análise de desempenho conjunta dos diferentes tipos de NCS da plataforma operando com a ECA. Adicionalmente, a confiabilidade e robustez de aplicação da ECA foram comprovadas através de experimentos realizados sob condições extremas (pior caso) de

ocupação da rede CAN, onde se comprovou a eficácia da ECA em manter o desempenho e a estabilidade dos NCS ainda que diminuindo a ocupação da rede CAN.

Finalmente, pode-se concluir que a realização deste trabalho contribuiu com o desenvolvimento da pesquisa em NCS, através da apresentação de uma solução para o desenvolvimento de NCS com redes CAN que utiliza simulação, análise e controle para lidar com os problemas relacionados ao projeto e a implementação de NCS. A sistematização das informações obtidas nesta tese de doutorado permitiu a publicação de uma série de trabalhos sobre o tema de NCS, os quais se espera que sirvam como referência para alavancar uma maior difusão da área de pesquisa relacionada à NCS e para promover o desenvolvimento de grupos nacionais de pesquisa nesta área.

7.1. LISTA DE PUBLICAÇÕES REALIZADAS

Os trabalhos científicos publicados relacionados ao desenvolvimento desta tese de doutorado foram:

Artigos em Revistas Científicas

✓ GODOY, E. P.; LOPES, W. C.; SOUSA, R. V.; PORTO, A. J. V.; INAMASU, R. Y. Modelagem e Simulação de Redes de Comunicação Baseadas no Protocolo CAN (Controller Area Network). *Controle & Automação* (Impresso), 2010.

✓ GODOY, E. P. ; SOUSA, R. V.; PORTO, A. J. V.; INAMASU, R. Y. Design of CAN-Based Distributed Control Systems with Optimized Configuration. *Journal of the Brazilian Society of Mechanical Sciences and Engineering* (Impresso), 2010.

✓ GODOY, E. P.; PORTO, A. J. V.; INAMASU, R. Y. Sistemas de Controle via Redes: Uma Nova Abordagem para Utilização de Redes Industriais. Revista C & I. Controle & Instrumentação, v. 139, p. 82-86, 2008.

Capítulos de livros publicados

✓ GODOY, E. P.; PORTO, A. J. V.; INAMASU, R. Y. Using Simulation Tools in the Development of a Networked Control Systems Research Platform. In: Aceito para publicação - Victor Juliano De Negri. (Org.). ABCM Symposium Series in Mechatronics. Rio de Janeiro: ABCM, 2010, v. 4.

✓ GODOY, E. P.; SOUSA, R.V., PORTO, A. J. V.; INAMASU, R. Y. Verification and Validation of an Analysis Model of CAN-Based Networked Control Systems. In: Paulo Eigi Miyagi, Oswaldo Horikawa, José Mauricio Motta. (Org.). ABCM Symposium Series in Mechatronics. 1 ed. Rio de Janeiro: ABCM, 2008, v. 3, p. 386-394.

Trabalhos completos publicados em anais de congressos

✓ GODOY, E. P.; PORTO, A. J. V.; INAMASU, R. Y. Applied Simulation to Evaluate the Quality of Control of Networked Control Systems. In: 2010 IEEE International Conference on Networking, Sensing, and Control, 2010, Chicago. Proceedings of the 2010 IEEE ICNSC 2010. Chicago: IEEE, 2010. p. 435-440.

✓ ZANATA, A. F.; GODOY, E. P.; PORTO, A.J.V. Simulação, Análise e Controle de Motor de Corrente Contínua Utilizando um Sistema de Controle via Rede CAN. In: 9th Brazilian Conference on Dynamics, Control and Their Applications, 2010, Bauru. Proceedings of the DINCON 2010. Bauru: ABCM, 2010.

✓ GODOY, E. P.; BRAGATO, B. N.; LULIO, L. C.; PORTO, A. J. V.; INAMASU, R. Y. Implementação e Avaliação de Sistemas de Controle via Redes Baseadas no Protocolo CAN Controller Area Network. In: 9th Brazilian Conference on Dynamics,

Control and their Applications, 2010, Bauru. Proceedings of the DINCON 2010. Bauru: ABCM, 2010.

✓ GODOY, E. P.; PEREIRA, R.R.D.; SCORZONI, F.; PORTO, A. J. V.; INAMASU, R. Y. CAN-Based Platform for the Study and Experimentation on Networked Control Systems (NCS). In: 3rd Annual Dynamic Systems and Control Conference and 5th IFAC Symposium on Mechatronic Systems, 2010, Boston. Proceedings of the IFAC MECH 2010. Boston: IFAC, 2010.

✓ GODOY, E. P.; PORTO, A. J. V.; INAMASU, R. Y. Sampling Time Adaptive Control Methodology for CAN-Based Networked Control Systems. In: 9th IEEE/IAS International Conference on Industry Applications, 2010, São Paulo. Proceedings of the IEEE/IAS INDUSCON 2010. São Paulo: IEEE, 2010.

✓ GODOY, E. P.; PORTO, A. J. V.; INAMASU, R. Y. Using Simulation Tools in the Development of a Networked Control Systems Research Platform. In: 20th International Congress of Mechanical Engineering, 2009, Gramado. Proceedings of COBEM 2009. Rio de Janeiro: ABCM, 2009.

✓ GODOY, E. P.; INAMASU, R.Y.; PORTO, A.J.V. Proposta de Utilização de Hardware-in-the-Loop no Desenvolvimento de Sistemas de Controle via Redes CAN. In: VIII Conferencia Internacional de Aplicações Industriais, 2008, Poços de Caldas. Anais do INDUSCON 2008. Poços de Caldas: IEEE, 2008.

✓ GODOY, E. P.; INAMASU, R.Y.; PORTO, A.J.V. Ferramentas de Análise, Simulação e Projeto de Sistemas de Controle via Redes: Uma Revisão. In: V Congresso Nacional de Engenharia Mecânica, 2008, Salvador. Anais do CONEM 2008. Rio de Janeiro: ABCM, 2008.

✓ GODOY, E. P.; PORTO, A. J. V.; INAMASU, R. Y. Desenvolvimento de Controladores Fuzzy para Sistemas de Controle via Redes CAN - Controller Area Network.

In: XVII Congresso Brasileiro de Automática, 2008, Juiz de Fora. Anais do CBA 2008. Campinas: SBA, 2008.

✓ GODOY, E. P.; SOUSA, R.V.; PORTO, A. J. V.; INAMASU, R. Y. Avaliação da Variabilidade de Desempenho de Redes de Controle CAN - Controller Area Network. In: 6th Brazilian Conference on Dynamics, Control and Their Applications, 2007, São José do Rio Preto. Proceedings of the DINCON 2007. São José do Rio Preto: ABCM, 2007.

✓ GODOY, E. P.; SOUSA, R.V.; PORTO, A. J. V.; INAMASU, R. Y. Verification and Validation of an Analysis Model of CAN-Based Networked Control Systems. In: 19th International Congress of Mechanical Engineering, 2007, Brasília. Proceedings of COBEM 2007. Brasília: ABCM, 2007.

7.2. TRABALHOS FUTUROS

Entre as perspectivas desse trabalho, espera-se que a quantidade de informações compilada na revisão bibliográfica e os conhecimentos (constatações e conclusões) apresentados durante o trabalho, bem como a extensa avaliação de ferramentas de simulação de NCS possa ser utilizada como base para confecção de trabalhos sobre a tecnologia NCS por outras instituições de pesquisa e ensino. Além disso, também é almejado que os novos desenvolvimentos propostos no trabalho (técnica de identificação e estratégia de controle adaptativo) possam ser utilizados para auxiliar o desenvolvimento e a implementação de NCS com redes CAN.

Os trabalhos futuros relacionados a esta tese de doutorado podem ser divididos em duas partes. Uma parte relacionada com a ampliação e aperfeiçoamento da plataforma de pesquisa de NCS, bem como o desenvolvimento de material pedagógico para utilização da mesma para

fins didáticos. Outra parte relacionada com as expectativas de novos desenvolvimentos relacionados á área de NCS, que atualmente tem recebido grande demanda para o estudo da utilização de redes sem fio (*wireless*).

Na parte relacionada com a ampliação e aperfeiçoamento da plataforma de NCS com redes CAN, um NCS composto por um braço robótico com três graus de liberdade esta atualmente em desenvolvimento. A migração da plataforma para um sistema operacional de tempo real (*LabVIEW Real Time*) também é planejada para fornecer capacidade real de estudo e aplicação de técnicas avançadas para o desenvolvimento de NCS como simulação *hardware-in-the-loop* e *rapid control prototyping*. Além disso, soluções de *hardware*, compatíveis com a utilização de microcontroladores, para possibilitar a implementação de comunicação e controle via redes sem fio (*Wi-Fi*, *ZigBee* e *Bluetooth*) têm sido pesquisadas e estudadas.

A parte relacionada com as expectativas de novos desenvolvimentos está focada principalmente no desenvolvimento e investigação de HNCS (sistemas de controle via redes híbridas) e WNCS (sistemas de controle via redes *wireless*), que atualmente são um dos principais focos de pesquisa na área de automação industrial. HNCS representam sistemas de controle distribuídos que utilizam redes industriais com fio e sem fio simultaneamente, enquanto WNCS buscam estudar a viabilidade de aplicação de NCS que somente utilizam redes de comunicação sem fio. O desenvolvimento de interfaces de redes sem fio baseadas em *Bluetooth* e *ZigBee* estão atualmente em desenvolvimento e visam permitir a pesquisa nos tópicos citados acima.

Outro foco de pesquisa a ser estudado será o desenvolvimento e aplicação de técnicas de controle baseado em modelo (*MPC - Model Based Control*) para controle e diagnóstico de falhas. Para isso, serão utilizados os modelos obtidos através da técnica de identificação de NCS desenvolvida neste trabalho. Também será realizado o estudo de estratégias de controle

utilizando estampas de tempo (*time stamping*) e sincronização de tempo (*time synchronization*).

8. REFERÊNCIAS*

ACTON, K., ANTOLOVIC, M., KALAPPA, N., LUNTZ, J., MOYNE, J. AND TILBURY, D., (2008). Practical Metrics for Evaluating Network System Performance. In: ANNUAL UM-ERC/RMS NETWORK PERFORMANCE WORKSHOP, 3, 2008, University of Michigan, USA. **Proceedings**. Disponível em: <<http://erc.engin.umich.edu/publications/NPWPaper.pdf>>. Acesso em: Janeiro, 2008.

ALBERT, A., PIETSCH, B.; VOETZ, F. (2005). Simulation Environment for Investigating the Impacts of Time-Triggered Communication on a Distributed Vehicle Dynamics Control System. In: INTERNATIONAL ECRTS WORKSHOP ON REAL-TIME AND CONTROL, 1, 2005, Palma de Mallorca, Spain. **Proceedings**. July 5.

ALLDREDGE, G. W. (2007). **PID and Model Predictive Control in a Networked Environment**. 130p. Master's Thesis, Department of Electrical Engineering and Computer Science, CASE Western Reserve University, August, 2007.

ÅSTRÖM, K. J., HÄGGLUND, T. (1995). **PID Controllers: Theory, Design, and Tuning**, 2nd ed., Instrument Society of America.

ÅSTRÖM, K. J., WITTENMARK, B. (1995). **Adaptive Control**, 2nd ed., Addison Wesley Publishing Company.

AL-HAMMOURI, A., BRANICKY, M. S.; LIBERATORE, V. (2008). Co-simulation Tools for Networked Control Systems. In: EGERSTEDT, M.; MISHRA, B. (Org.). **Hybrid Systems: Computation and Control**, Lecture Notes in Computer Science. Berlin: Springer, v. 4981, p. 16-29.

BAILLIEUL, J.; ANTSAKLIS, P.J. (2007). Control and Communication Challenges in Networked Real-Time Systems. **IEEE Proceedings of the Technology of Networked Control Systems**, v. 95, n. 1, p. 9-28, January.

BERBRA, C.; SIMON, D.; GENTIL, S.; LESECQ, S. (2009). Hardware in the loop networked control and diagnosis of a quadrotor drone. In: IFAC INTERNATIONAL SYMPOSIUM ON FAULT DETECTION, SUPERVISION AND SAFETY OF TECHNICAL PROCESSES, 7, 2009, Barcelona, Spain. **Proceedings**. June 30 - July 3.

BLAZEWICZ, J.; ECKER, K.; SCHMIDT, G.; WEGLARZ, J. (2001). **Scheduling Computer and Manufacturing Processes**, 2nd ed., Springer, 485p.

* De acordo com: ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação: referências: elaboração. Rio de Janeiro, 2002.

BJÖRKBOM, M.; NETHI, S.; KOHTAMÄKI, T. (2010). **PiccSIM Manual**. Department of Communications and Networking, Aalto University, Finland, October. Disponível em: <http://wsn.tkk.fi/en/software/piccsim/piccsim_manual_1.0.pdf>. Acesso em: Janeiro. 2010.

BOSCH. (2006). **CAN 2.0 Specification Version**. Disponível em: <<http://www.can.bosch.com>>. Acesso em: Setembro. 2006.

CAMACHO, A.; MARTÍ, P.; VELASCO, M.; LOZOYA, C.; VILLÀ, R.; FUERTES, J.M.; GRIFUL, E. (2010). Self-triggered networked control systems: An experimental case study. In: IEEE International Conference on Industrial Technology (ICIT), 2010, Vina del Mar, Spain. **Proceedings**. March 14-17, p.123-128.

CASANOVA, V.; SALT, J.J.; CUENCA, A.; MASCAROS, V. (2006). Networked Control Systems over Profibus-DP: Simulation Model. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL APPLICATIONS, 2006, Munich, Germany. **Proceedings**. October 4-6, p.1337-1342.

CAVALIERI, S.; DI STEFANO, A.; LO BELLO, L.; MIRABELLA, O. (1996). CAN Assessment in Time-Critical Cyclic Applications through Petri Net Model, **IEEE Control Systems Magazine**, p. 922-927.

CERVIN, A. (2003). Using Jitterbug to Derive Control Loop Timing Requirements. In: CO-DESIGN OF EMBEDDED REAL-TIME SYSTEMS WORKSHOP (CERTS'03), 2003, Porto, Portugal. **Proceedings**. July, p. 1-6.

CERVIN, A.; ARZEN, K. E.; HENRIKSSON, D.; LLUESMA, M.; BALBASTRE, P.; RIPOLL, I.; CRESPO, A. (2006). Control Loop Timing Analysis Using TrueTime and Jitterbug. In: IEEE CONFERENCE ON COMPUTER AIDED CONTROL SYSTEMS DESIGN, 2006. **Proceedings**. p. 1194-1199, 4-6 October.

CERVIN, A.; HENNINGSSON, T. (2008). Scheduling of Event-Triggered Controllers on a Shared Network. In: IEEE CONFERENCE ON DECISION AND CONTROL, 47, 2008, Cancun, Mexico. **Proceedings**. 9-11 December.

CERVIN, A.; HENRIKSSON, D.; LINCOLN, B.; EKER, J.; ARZEN, K. (2003). How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime. **IEEE Control System Magazine**, p. 16-30, June.

CERVIN, A.; LINCOLN, B. (2006). **Jitterbug 1.21 Reference Manual**. Department of Automatic Control, LTH, Lund University, Lund, Sweden, February. Disponível em: <<http://www.control.lth.se/~lincoln/jitterbug/manual.pdf>>. Acesso em: Setembro. 2008.

CERVIN, A.; OHLIN, M.; HENRIKSSON, D. (2007). Simulation of Networked Control Systems Using TrueTime. In: INTERNATIONAL WORKSHOP ON NETWORKED CONTROL SYSTEMS: TOLERANT TO FAULTS, 3, 2007, Nancy, France. **Proceedings**. June.

CERVIN, A.; VELASCO, M.; MARTÍ, P.; CAMACHO, A. (2010). Optimal Online Sampling Period Assignment: Theory and Experiments. **IEEE Transactions on Control Systems Technology**, v. 99, p.1-9. July.

CHANG, X. (1999). Network simulations with OPNET. In: CONFERENCE ON WINTER SIMULATION: SIMULATION –A BRIDGE TO THE FUTURE, 31, 1999, Arizona, USA. **Proceedings**. v. 1, p. 307-314.

CHEN, C. H.; LIN, C. L.; HWANG, T. S. (2007). Stability of Networked Control Systems with Time-Varying Delays. **IEEE Communications Letters**, v. 11, n. 3, March.

CHRISTENSEN, S., JORGENSEN, J.B.; KRISTENSEN, L. (1997). Design/CPN - A computer tool for Colored Petri Nets. In: INTERNATIONAL WORKSHOP ON TOOLS AND ALGORITHMS FOR CONSTRUCTION AND ANALYSIS OF SYSTEMS, 3, 1997. **Proceedings**. v. 1217, p. 209 – 223.

CIA. (2005). CAN in Automation. Disponível em: <[http:// www.can-cia.org/](http://www.can-cia.org/)>. Acesso em: Novembro. 2005.

CLOOSTERMAN, M.; WOUW, N. V.; HEEMELS, M.; NIJMEIJER, H. (2006). Robust Stability of Networked Control Systems with Time-varying Network-induced Delays, In: IEEE CONFERENCE ON DECISION & CONTROL, 45, 2006. **Proceedings**. p. 4980-4985, December.

COUTINHO, M. M. (2007). **NETWORK SIMULATOR Guia Básico para Iniciantes**. Disponível em: < <http://www.cci.unama.br/margalho/networksimulator/>>. Acesso em: Julho. 2008.

DE BIASI, M. (2008). **Implementation of a wireless HART simulator and its use in studying packet loss compensation in networked control**. 130p. Master's Thesis, XR-EE-RT 2008:010, School of Electrical Engineering, KTH - The Royal Institute of Technology Stockholm, Sweden, February, 2008.

DIAS, A. L.; PEDRO, L. M.; CAURIN, G. A. P. (2008). Using Fieldbus in Realtime control system for robotic applications. In: CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA (CONEM), 5, 2008, Salvador. **Anais**. 25 - 28 Agosto.

DIOURI, I.; GEORGES, J. P.; RONDEAU, E. (2007). Accommodation of delays for NCS using classification of service. In IEEE INTERNATIONAL CONFERENCE ON NETWORKING, SENSING AND CONTROL (ICNSC), 2007, London, UK. **Proceedings**. 15-17 April.

EL-KHOURY, J.; TÖRNGREN, M. (2001). Towards a Toolset for Architectural Design of Distributed Real-Time Control Systems. In: IEEE INTERNATIONAL REAL-TIME SYSTEMS SYMPOSIUM, 22, 2001. **Proceedings**. v. 0, n. 0, p. 267-276.

ERIKSSON, L. (2008) **PID Controller Design and Tuning in Networked Control Systems**, 118p., PhD Thesis, Department of Automation and Systems Technology, Helsinki University of Technology, Finland, 2008.

ESTRADA, T.; ANTSAKLIS, P. A. (2009). Performance of Model-Based Networked Control Systems with discrete-time plants. In: MEDITERRANEAN CONFERENCE ON CONTROL AND AUTOMATION, 17, 2009, Greece. **Proceedings**. p. 628-633, 24-26 June.

FADAEI, A.; SALAHSHOOR, K. (2008). Design and implementation of a new fuzzy PID controller for networked control systems. **ISA Transactions**, v.47, p. 351–361.

FARIAS, G.; CERVIN, A.; ÅRZÉN, K.E.; DORMIDO, S.; ESQUEMBRE, F. (2010). Java Simulations of Embedded Control Systems. **Sensors**, v.10, n. 9, p. 8585-8603.

FRIDMAN, E.; SHAKED, U. (2003). Delay-dependent stability and H_∞ control: constant and time varying delays. **International Journal of Control**, v. 76, n. 1, p. 48-60.

FUMMI, F., MARTINI, S., MONGUZZI, M., PERBELLINI, G.; PONCINO, M. (2004). Software/Network Co-Simulation of Heterogeneous Industrial Networks Architectures. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER DESIGN, 2004, San Jose, California. **Proceedings**. 11-13 October.

GAID, M. B.; CELA, A.; DIALLO, S.; KOCIK, R.; HAMOUCHE, R.; REAMA, A. (2006). Performance Evaluation of the Distributed Implementation of a Car Suspension System. In IFAC WORKSHOP ON PROGRAMMABLE DEVICES AND EMBEDDED SYSTEMS, 2006, Czech Republic. **Proceedings**. February.

GEORGES, J. P.; VATANSKI, N.; RONDEAU, E.; AUBRUN, C.; JAMSA-JOUNELA, S. L. (2006). Control compensation based on upper bound delay in networked control systems. In: INTERNATIONAL SYMPOSIUM ON MATHEMATICAL THEORY OF NETWORKS AND SYSTEMS, 17, 2006, Kyoto. **Proceedings**. July.

GHUDE, S. (2007) Design a PID Controller with Missing Packets in a Networked Servo-System. 80p., Masters Thesis, Faculty of Engineering and Surveying, University of Southern Queensland, Australia, 2007.

GIRAULT, C.; VALK, R. (2003). **Petri nets for systems engineering**: A guide for modeling, verification and application. 1st ed., Springer, 607p.

GODOY, E. P. (2007). **Desenvolvimento de uma Ferramenta de Análise de Desempenho de redes CAN para Aplicações em Sistemas Agrícolas**, 93p. Dissertação (Mestrado em Engenharia Mecânica), Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2007.

GODOY, E. P.; BRAGATO, B. N.; LULIO, L. C.; PORTO, A. J. V.; INAMASU, R. Y. (2010c). Implementação e avaliação de sistemas de controle via redes baseados no protocolo CAN – Controller Area Network. In: BRAZILIAN CONFERENCE ON DYNAMICS,

CONTROL AND THEIR APPLICATIONS (DINCON), 9, 2010. Bauru, SP, Brasil. **Proceedings**. ABCM, Junho 07-11.

GODOY, E. P.; INAMASU, R. Y.; PORTO, A. J. V. (2008a). Ferramentas de Análise, Simulação e Projeto de Sistemas de Controle via Redes: Uma Revisão. In: CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA (CONEM), 5, 2008, Salvador. **Anais**. ABCM.

_____. (2008b). Proposta de Utilização de Hardware-in-the-Loop no Desenvolvimento de Sistemas de Controle via Redes CAN. In: CONFERENCIA INTERNACIONAL DE APLICACIONES INDUSTRIALES (INDUSCON), 7, 2008, Poços de Caldas. **Anais**. IEEE.

GODOY, E. P.; PEREIRA, R. R. D.; SCORZONI, F.; PORTO, A. J. V.; INAMASU, R. Y. (2010d). CAN-based Platform for the Study and Experimentation on Networked Control Systems (NCS). In: IFAC Symposium on Mechatronic Systems, 5, 2010. Cambridge, MA, USA. **Proceedings**. September 13-15, 2010.

GODOY, E. P.; PORTO, A. J. V.; INAMASU, R. Y. (2009). Using simulation tools in the development of a networked control systems research platform. In: INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING (COBEM), 20, 2009. Gramado, RS, Brazil. **Proceedings**. ABCM, November 15-20, 2009.

GODOY, E. P.; PORTO, A. J. V.; INAMASU, R. Y. (2010a). Applied Simulation to Evaluate the Quality of Control of Networked Control Systems. In: 2010 IEEE INTERNATIONAL CONFERENCE ON NETWORKING, SENSING, AND CONTROL (ICNSC), Chicago, IL, USA. **Proceedings**. April 11-13, 2010.

_____. (2010b). Sampling Time Adaptive Control Methodology for CAN-Based Networked Control Systems. In: IEEE/IAS INTERNATIONAL CONFERENCE ON INDUSTRY APPLICATIONS, 9, 2010. São Paulo, SP, Brazil. **Proceedings**. November 8 - 10, 2010.

GODOY, E. P.; SOUSA, R. V.; PORTO, A. J. V.; INAMASU, R. Y. (2007). Avaliação da Variabilidade de Desempenho de Redes de Controle CAN - Controller Area Network. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS (DINCON), 6, 2007, São José do Rio Preto. **Proceedings**. ABCM.

GODOY, E. P.; SOUSA, R. V.; PORTO, A. J. V.; INAMASU, R. Y. (2010e). Design of CAN-Based Distributed Control Systems with Optimized Configuration. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, v. 32, p. 420-426.

GOODWIN, G. C.; QUEVEDO, D. E.; SILVA, E. I. (2008). Architectures and coder design for networked control systems, **Automatica**, v. 44, p. 248–257, January.

GU, F.; HARRISON, W.S.; TILBURY, D.M.; YUAN, C. (2007). Hardware-In-The-Loop for Manufacturing Automation Control: Current Status and Identified Needs”, In: ANNUAL IEEE INTERNATIONAL CONFERENCE ON AUTOMATION SCIENCE AND ENGINEERING, 3, 2007. **Proceedings**. p. 1105-1110.

GUPTA, R.A.; CHOW, M.Y. (2010). Networked Control System: Overview and Research Trends. **IEEE Transactions on Industrial Electronics**, v. 57, n. 7, p. 2527-2535, July.

HASAN, M. S.; YU, H.; GRIFFITHS, A.; YANG, T.C. (2008). Co-simulation framework for Networked Control Systems over multi-hop mobile ad-hoc networks. In: WORLD CONGRESS THE INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL, 17, 2008, Seoul, Korea. **Proceedings**. July 6-11, 2008, p. 12552-12557.

HENRIKSSON, D; CERVIN, A., (2003). **TrueTime 1.13 Reference Manual**. Internal Report, Department of Automatic Control, Lund University, Sweden, October 2003.

HESPANHA, J. P.; NAGHSHTABRIZI, P.; XU, Y. (2007). A Survey of Recent Results in Networked Control Systems, **IEEE Proceedings of the Technology of Networked Control Systems**, v. 95, n. 1, p. 138-162, January.

HETEL, L.; CLOOSTERMAN, M.; VAN DE WOUW, N.; HEEMELS, M.; DAAFOUZ, J.; NIJMEIJER, H. (2009). Comparison of Stability Characterizations for Networked Control Systems. In: IEEE CONFERENCE ON DECISION AND CONTROL AND CHINESE CONTROL CONFERENCE, 48, 2009, Shanghai, China. **Proceedings**. 16-18 December.

HOKAYEM, P. F.; ABDALLAH, C. T. (2004). Inherent Issues in Networked Control Systems: A Survey. In: AMERICAN CONTROL CONFERENCE (ACC), 2004, Massachusetts, Boston. **Proceedings**. p. 4897-4902, June 30 - July 2.

HUO, Z.; FANG, H.; MA, C. (2004). Networked control system: state of the art. In: WORLD CONGRESS ON INTELLIGENT CONTROL AND AUTOMATION (WCICA), 50, 2004. **Proceedings**. v. 2, p. 1319- 1322, June.

ISAKSSON, A.J.; GRAEBE, S.F. (2002). Derivative filter is an integral part of PID design. **IEE Proceedings Control Theory and Applications**, p. 61. 149, n. 1, January.

ISERMANN, R. (2007) Mechatronic systems — Innovative products with embedded control, **Control Engineering Practice**, v. 16, p. 14–29, June.

JIANYONG, Y.; SHIMIN, Y.; HAIQING, W. (2004). Survey on the performance analysis of networked control systems. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, 2004, **Proceedings**. 10-13 October, v. 6, p. 5068- 5073.

JUN, T.; ZHAN-LIN, W. (2005). Analysis of PROFIBUS.DP Network Delay and Its Influence on the Performance of Control Systems. **Wuhan University Journal of Natural Sciences**, v. 10, n. 5, p. 877-882.

KAO, C. Y.; LINCOLN, B. (2004). Simple stability criteria for systems with time-varying delays. **Automatica**, v. 40, p. 1429-1434.

KHAN, A. A.; TILBURY, D. M.; MOYNE, J.R. (2008). Favorable effect of time delays on tracking performance of type-I control systems. **IET Control Theory Applications**, the Institution of Engineering and Technology, v. 2, n. 3, p. 210–218.

KIM, D. S. (2003). **A Scheduling Method for Networked Control Systems in Fieldbus Environments**. PhD Thesis, Seoul National University, February, 2003.

KOLLA, S. (2007). CAN-based Fieldbus Experiments. In: ASEE ANNUAL CONFERENCE & EXPOSITION, 2007, Honolulu, Hawaii. **Proceedings**. American Society of Engineering Education, 20-24 June.

KUSNADI, D. (2007). **TrueTime in Scicos**. 72p., Masters Thesis, ISSN 0280-5316, Department of Automatic Control, Lund University, Sweden, June, 2007.

KUTIL, M.; SUCHA, P.; SOJKA, M.; HANZALEK, Z. (2007). **TORSCHÉ Scheduling Toolbox for Matlab User's Guide** (Release 0.4.0). Department of Control Engineering, Czech Technical University. Disponível em: <<http://rttime.felk.cvut.cz/scheduling-toolbox/>>. Acesso em: Setembro. 2008.

LEE, K. C.; LEE, S.; LEE, M. H. (2003). Remote fuzzy logic control of networked control system via Profibus-DP. **IEEE Transactions on Industrial Electronics**, v. 50, n. 4, p. 784-792, August.

LEE, K. C.; LEE, S.; LEE, M. H. (2004). Implementation and PID tuning of network-based control systems via Profibus polling network. **Computer Standards & Interfaces**, v. 26, n. 3, p. 229–240, May.

LI, Y.; FANG, H. (2005). Control methodologies of large delays in networked control systems. In: INTERNATIONAL CONFERENCE ON CONTROL AND AUTOMATION (ICCA), 2005. **Proceedings**. v. 2, p. 1225–1230.

LIAN, F. L. (2001). **Analysis, Design, Modeling, and Control of Networked Control Systems**. 194p., PhD Thesis. University of Michigan, USA, 194p.

LIAN, F. L.; MOYNE, J. R.; TILBURY, D. M. (2001). Performance evaluation of control networks: Ethernet, ControlNet and Device Net. **IEEE Control Systems Magazine**, p.66-83.

_____. (2002). Network Design Consideration for Distributed Control Systems. **IEEE Transactions on Control Systems Technology**, v. 10(2), p. 297-307, March.

LIAN, F. L.; YOOK, J. K.; OTANEZ, P.; TILBURY, D. M.; MOYNE, J. R. (2004). Design of Sampling and Transmission Rates for Achieving Control and Communication Performance in Networked Agent Systems. In: AMERICAN CONTROL CONFERENCE (ACC), 2004, Denver, Colorado. **Proceedings**. June 4-6, p. 3329-3334.

LIAN, F. L.; YOOK, J. K.; TILBURY, D. M.; MOYNE, J. R. (2006). Network architecture and communication modules for guaranteeing acceptable control and communication

performance for networked multi-agent systems. **IEEE Transactions on Industrial Informatics**, v. 2(1), p. 12-24, February.

LIMA, F.S.; GUEDES, L.A.; SALAZAR, A.O.; MAITELLI, A.L. (2004). Hybrid Environment for Tests and Training in Fieldbuses, In: CONFERÊNCIA INTERNACIONAL DE APLICAÇÕES INDUSTRIAIS (INDUSCON), 6, 2004. **Proceedings**.

LIU, L.; FREY, G. (2008). Feasibility Analysis for Networked Control Systems by Simulation in Modelica. In: INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION, 13, 2008. **Proceedings**. p. 729-732.

LLUESMA, M.; CERVIN, A.; BALBASTRE, P.; RIPOLL, I.; CRESPO, A. (2006). Jitter Evaluation of Real-Time Control Systems. In: IEEE INTERNATIONAL CONFERENCE ON EMBEDDED AND REAL-TIME COMPUTING SYSTEMS AND APPLICATIONS (RTCSA), 12, 2006, Sydney. **Proceedings**. p. 257-260.

LOPES, W. C. (2007). **Análise de Desempenho do Protocolo CAN para Aplicação na Área Agrícola utilizando Rede de Petri Colorida**. 121p., Dissertação (Mestrado em Engenharia Mecânica), Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2007.

LUSTOSA, H. D.; SOUZA, M. L. DE O. (2008). Influence of databus types and some high level characteristics in networked control systems. In: ARTIST2 SOUTH-AMERICAN SCHOOL FOR EMBEDDED SYSTEMS, 2008, Florianopolis, Brasil. **Proceedings**. Disponível em: < <http://urlib.net/sid.inpe.br/mtc-m17@80/2009/03.17.01.32>>. Acesso em: Outubro, 2009.

MARTÍ, P.; CAMACHO, A.; VELASCO, M.; GAID, M. E. M. B. (2010). Runtime Allocation of Optional Control Jobs to a Set of CAN-Based Networked Control Systems. **IEEE Transactions on Industrial Informatics**, v. 6, n. 4, p. 503-520. November.

MARTÍN, J. V.; TADEO, F. (2006). Learning fieldbus technology using open and flexible educational equipment, In: IFAC SYMPOSIUM ON ADVANCES IN CONTROL EDUCATION, 7, 2006. **Proceedings**. v. 7(1).

MELLIOS, T.; KANSOULIDOU, C.; HASSAPIS, G. (2005). Evaluation of Industrial Networked Control Systems Using Hardware-In-The-Loop Simulation, In: INTERNATIONAL CONFERENCE ON TECHNOLOGY AND AUTOMATION – ICTA, 5, 2005. **Proceedings**. p. 235-241.

MOYNE, J. R.; TILBURY, D. M. (2007). The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data, **IEEE Proceedings of the Technology of Networked Control Systems**, v. 95, n. 1, p. 29-47, January.

MURATA, T. (1989). Petri nets: Properties, analysis and applications. **IEEE Proceedings**, v. 77, p. 541–580.

NAIR, G. N.; FAGNANI, F.; ZAMPIERI, S.; EVANS, D R. J. (2007). Feedback control under data rate constraints: An overview. **IEEE Proceedings of the Technology of Networked Control Systems**, v. 95, n. 1, p. 108-137, January.

NETHI, S.; POHJOLA, M.; ERIKSSON, L.; JANTTI, R. (2007a). Platform for Emulating Networked Control Systems in Laboratory Environments. In: IEEE INTERNATIONAL SYMPOSIUM ON WORLD OF WIRELESS, MOBILE AND MULTIMEDIA NETWORKS (WOWMOM), 2007, Helsinki, Finland. **Proceedings**. 18-21 June, p. 1-8.

_____. (2007b). Simulation Case Studies of Wireless Networked Control Systems, In: ACM/IEEE INTERNATIONAL SYMPOSIUM ON MODELING, ANALYSIS AND SIMULATION OF WIRELESS AND MOBILE SYSTEMS, 10, 2007, Chania, Greece. **Proceedings**. 22-26 October, p. 100 – 104.

NI (2005). National Instruments. Disponível em: <<http://www.ni.com/>>. Acesso em: Novembro. 2005.

NICULESCU, S. I.; GU, K.; ABDALLAH, C.T. (2003). Some remarks on the delay stabilizing effect in SISO systems. In: AMERICAN CONTROL CONFERENCE (ACC), 2003. **Proceedings**. v. 3, p. 2670–2675

NILSSON, J. (1998). **Real-time control systems with delays**, PhD Thesis, Lund Institute of Technology.

NS-2. (2007). The Network Simulator. Disponível em: < <http://www.isi.edu/nsnam/ns/>>. Acesso em: Novembro. 2007.

OHLIN, M.; HENRIKSSON, D.; CERVIN, A. (2010). **TrueTime 2.0 Reference Manual**. Internal Report, Department of Automatic Control, Lund University, Sweden, June 2010, Disponível em: <<http://www.control.lth.se/truetime/report-2.0-beta1.pdf>>. Acesso em: Janeiro. 2010

OKANO, R.; OHTANI, T.; NAGASHIMA, A. (2008). Networked Control Systems by PID Controller Improvement of Performance Degradation Caused by Packet loss. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS (INDIN), 2008, Korea. **Proceedings**. July, p. 1126-1132.

ONAT, A.; NASKALI, T.; PARLAKAY, E. (2008). Model Based Predictive Networked Control Systems. In: WORLD CONGRESS OF THE INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL, 17, 2008, Seoul, Korea. **Proceedings**. July 6-11, p. 13000-13005.

OTHMAN, H. F.; AJI, Y. R.; FAKHREDDIN, F. T.; AL-ALI, A. R. (2006). Controller Area Networks: Evolution and Applications, In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES (ICTTA), 2, 2006. **Proceedings**. v. 2, p. 3088- 3093, 24-28 April.

PALOPOLI, L.; LIPARI, G.; LAMASTRA, G.; ABENI, L. (2002). An object-oriented tool for simulating distributed real-time control systems. **Software - Practice and Experience**, v. 32, p. 907-932.

PANG, Y.; YANG, S. H.; NISHITANIC, H. (2006). Analysis of control interval for foundation fieldbus-based control systems. **ISA Transactions**, v. 45, n. 3, p. 447-458, July.

PARK, T. J.; HAN, C. S.; LEE, S. H. (2005). Development of the electronic control unit for the rack-actuating steer-by-wire using the hardware-in-the-loop simulation system, **Mechatronics**, v. 15, p.899-918.

PEREZ, D. A. (2006). Propostas de Co-Projeto para Sistemas de Controle via Rede. 104p., Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis, 2006.

PEREZ, D. A.; MORENO, U. F.; MONTEZ, C. B. (2006). CAN Networked Control Systems with Remote Controllers using Jitter Margin. In: IEEE INTERNATIONAL CONFERENCE OF INDUSTRIAL ELECTRONICS SOCIETY (IECON), 2006, Paris. **Proceedings**. p. 252-257. IEEE

PEREZ, D. A.; MORENO, U. F.; MONTEZ, C. B.; SANTOS, T. L. M. (2007). Sampling period assignment for networked control systems based on the plant operation mode. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (SBAI), 7, 2007, Florianópolis. **Anais**. SBA.

PINOTTI Jr, M.; BRANDAO, D. (2005). A Flexible Fieldbus Simulation Platform for Distributed Control Systems Laboratory Courses. **International Journal of Engineering**, v. 21, n. 6, p. 1050-1058.

POHJOLA, M. (2006). **PID Controller Design in Networked Control Systems**, 91p., Masters Thesis, Department of Automation and Systems Technology, Helsinki University of Technology, 2006.

POHJOLA, M. (2008). Adaptive Jitter Margin PID Controller. In: IEEE CONFERENCE ON AUTOMATION SCIENCE AND ENGINEERING (CASE), 4, 2008, Key Bridge, Marriott, Washington DC, USA. **Proceedings**. August 23-26, p. 534-539.

POHJOLA, M.; ERIKSSON, L.; HÖLTTÄ, V.; OKSANEN, T. (2005). Platform for Monitoring and Controlling Educational Laboratory Process over Internet. In: IFAC TRIENNIAL WORLD CONGRESS, 16, 2005, Prague, Czech Republic. **Proceedings**. p.55-60.

POHJOLA, M.; ERIKSSON, L.; KOIVO, H. (2006). Tuning of PID Controllers for Networked Control Systems, In: ANNUAL CONFERENCE OF THE IEEE INDUSTRIAL ELECTRONICS SOCIETY (IECON), 32, 2006, Paris, France. **Proceedings**. November 6-10.

PUNNEKKAT, S.; HANSSON, H. A.; NORSTROM, C. (2000). Response Time Analysis under Errors for CAN. In: IEEE SYMPOSIUM OF REAL-TIME TECHNOLOGY AND APPLICATIONS, 2000, Washington, USA. **Proceedings**. p. 258-266.

QUEVEDO, D. E.; GOODWIN, G. C. (2005). An improved architecture for networked control systems. In IFACWORLD CONGRESS, 6, 2005, Prague, Czech Republic. **Proceedings**.

QUEVEDO, D. E.; SILVA, E. I.; GOODWIN, G. C. (2008). Control over Unreliable Networks Affected by Packet Erasures and Variable Transmission Delays. **IEEE Journal on Selected Areas in Communications**, v. 26, n. 4, p. 672-685.

RAJU, B. (2009). **Time Delay Compensation Schemes with Application to Networked Control Systems**, 104p., Masters Thesis, Department of Electrical Engineering , National Institute of Technology, Rourkela, India, 104p.

REDELL, O.; EL-KHOURY, J.; TORNGREN, M. (2004). The AIDA tool-set for design and implementation analysis of distributed real-time control systems. **Journal of Microprocessors and Microsystems**, v. 28:4, p. 163.182.

REUTERSWÄRD, P.; ÅKESSON, J.; CERVIN, A.; ÅRZÉN, K.E. (2009). TrueTime Network—A Network Simulation Library for Modelica. In: INTERNATIONAL MODELICA CONFERENCE, 7, 2009, September. **Proceedings**. Modelica Association

RICHARD, J. P. (2003). Time-delay systems: an overview of some recent advances and open problems. **Automatica**, v. 39, p. 1667–1694

RODRIGUEZ, R. V. (2007). **Network-induced delay models for CAN-based Networked Control Systems Evaluation**. 128p., Masters Thesis, Instituto Tecnológico Y de Estudios Superiores de Monterrey, México, 2007.

SALA, A.; CUENCA, A.; SALT, J. (2009). A retunable PID multi-rate controller for a networked control system. **Information Sciences**, v. 79, n. 14, p. 2390-2402, June.

SANTOS, M. M. D. (2004). **Metodologias de Projeto para Sistemas de Controle via Redes**. 110p., Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis, 2004.

SANTOS, M. M. D.; VASQUES, F.; STEMMER, M. R. (2003). Avaliação das propriedades temporais de duas redes de controle: CAN e PROFIBUS, **Acta Scientiarum Technology**, v. 25, n. 2, p. 193-201.

SANTOS, M. M., OLIVEIRA, B. C.; VASQUEZ, F., STEMMER, M. R. (2006). Projeto de Sistemas de Controle via Redes baseado no Tempo de Resposta de Mensagens, In: CONGRESSO LATINO-AMERICANO DE CONTROLE AUTOMÁTICO (CLCA), 12, 2006, Salvador. **Anais**. p.266-269.

SANTOS, M. M., VASQUEZ, F., STEMMER, M. R. (2004). Performance Analysis of Networked Control Systems over CAN and Token Passing Networks. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL APPLICATIONS (INDUSCON), 6, 2004, Itajaí. **Proceedings**. v. 1. p. 120-128.

SANTOS, M. M., VASQUEZ, F., STEMMER, M. R. (2006). The Jitter margin applied to the design of networked control systems. In: CONGRESSO BRASILEIRO DE AUTOMATICA (CBA), 2006, Salvador. **Anais**. p.1363-1368.

SCILAB. (2008). Scilab Home Page. Disponível em: < <http://www.scilab.org/>>. Acesso em: Abril. 2008.

SEILER, P.; SENGUPTA, B. (2001). Analysis of communication losses in vehicle control problems. In: AMERICAN CONTROL CONFERENCE (ACC), 2001. **Proceedings**. v. 2, p. 1491–1496.

SEURET, A.; MICHAUT, F.; RICHARD, J. P.; DIVOUX, T. (2006). Networked control using GPS synchronization. In: IEEE AMERICAN CONTROL CONFERENCE, 25, 2006, Minneapolis, Minnesota, USA. **Proceedings**. June.

SHIN, M.; SUNWOO, M. (2007). Optimal Period and Priority Assignment for a Networked Control System Scheduled by a Fixed Priority Scheduling System. **International Journal of Automotive Technology**, v. 8, n. 1, p. 39-48, February.

SILVA, E. I.; GOODWIN, G. C.; QUEVEDO, D. E. (2008). On Networked Control Architectures for MIMO Plants. In: WORLD CONGRESS OF THE INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL, 17, 2008, Seoul, Korea. **Proceedings**. July 6-11, p. 8044-8049.

SO, J. K. C. (2003). **Delay Modeling and Controller Design for Networked Control Systems**. 147p., Masters Thesis, University of Toronto, Toronto, 2003.

SOGLO, A. B.; XIANHUI, Y. (2006). Networked Control System Simulation Design and Its Application. **Tsinghua Science and Technology**, v. 11, n. 3, p. 287-294, June.

SOREL, Y. (2005): From modeling/simulation with Scilab/Scicos to optimized distributed embedded real-time implementation with SynDEx. In: INTERNATIONAL WORKSHOP ON SCILAB AND OPEN SOURCE SOFTWARE ENGINEERING (SOSSE), 2005, Wuhan, China. **Proceedings**.

SOUSA, R. V. (2002). **CAN (Controller Area Network): uma abordagem para automação e controle na área agrícola**. Dissertação (Mestrado em Engenharia Mecânica), Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2002.

SUCHA, P.; KUTIL, M.; SOJKA, M.; HANZALEK, Z. (2006). TORSCHÉ Scheduling Toolbox for Matlab. In: IEEE INTERNATIONAL SYMPOSIUM ON COMPUTER-AIDED CONTROL SYSTEMS DESIGN, 2006. **Proceedings**. p. 1181-1186, 4-6 October.

TIAN, X.; WANG, X.; CHENG, Y. (2007). A Self-tuning Fuzzy Controller for Networked Control System. **International Journal of Computer Science and Network Security**, v. 7, n. 1, p. 97-102.

TINDELL, K.; BURNS, A., WELLINGS, A. (1995). Calculating Controller Area Network (CAN) Message Response Time. **Control Engineering Practice**, v. 3, n. 8, p. 1163-1169.

TIPSUWAN, Y. (2003). **Gain Scheduling for Networked Control System**. 175p., PhD Thesis, Department of Electrical and Computer Engineering, University of North Carolina State, 2003.

TIPSUWAN, Y.; CHOW, M. Y. (2003). Control Methodologies in Networked Control Systems. **Control Engineering Practice**, v. 11, n. 3, p. 1099-1111.

TORNGREN, M.; HENRIKSSON, D.; REDELL, O.; KIRSCH, C.; EL-KHOURY, J.; SIMON, D.; SOREL, Y.; HANZALEK, Z.; ARZEN, K. E. (2006). **Co-design of Control Systems and their real-time implementation – A Tool Survey**. Technical Report, Mechatronics Lab, Department of Machine Design, Royal Institute of Technology, Sweden.

VALDIVIA, R. H. V.; SOUZA, M. L. DE O. (2007). An study on the influence of some characteristics of communication protocols on the performance of networked control systems. In: INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING (COBEM), 17, 2007, Brasilia, Brazil. **Proceedings**.

VATANSKI, N.; GEORGES, J. P.; AUBRUN, C.; RONDEAU, E.; JAMSA-JOUNELA, S. L. (2009). Networked control with delay measurement and estimation. **Control Engineering Practice**, v. 17, p. 231–244.

VOGLAUER, B.; GARCIA, R.; JORGL, H.P. (2005). Improvements of a three-tank-system operated in real time with MATLAB in a PLC-Profibus-network, In: IFAC WORLD CONGRESS, 16, 2005. **Proceedings**. v. 16, p. 43-48.

WALSH, G. C.; YE, H. (2001). Scheduling of networked control systems. **IEEE Control Systems Magazine**. V. 21 p. 57-65.

WEI, R.; JIN, M.H.; XIA, J.J.; XIE, Z.W.; LIU, H. (2006). Distributed Hardware-in-the-Loop Simulation for Space Robot on CAN Bus, In: ANNUAL CONFERENCE OF THE IEEE INDUSTRIAL ELECTRONICS SOCIETY (IECON), 32, 2006. **Proceedings**. p. 3928-3933.

WU, W.; HE, Y.; SHE, J. H.; LIU, G. P. (2004). Delay-dependent criteria for robust stability of time varying delay systems, **Automatica**, v. 40, p. 1435-1439.

XIA, F.; WANG, Z.; SUN, Y. (2004). Simulation based performance analysis of networked control systems with resource constraints. In: ANNUAL CONFERENCE OF IEEE INDUSTRIAL ELECTRONICS SOCIETY (IECON), 30, 2004. **Proceedings**. v. 3, p. 2946-2951, November.

YANG, T. C. (2006a). **NCS_Simu User Guide**. Disponível em: <<http://www.sussex.ac.uk/Users/taiyang/>>. Acesso em: Agosto. 2008.

_____. (2006b). Networked control system: a brief survey. **IEEE Proceedings of Control Theory and Applications**, v. 153, n 4, July, p. 403–412.

YANG, T. C. (2007). **Networked and Embedded Control Systems**: An introduction, applications in automotive industry, and a new simulation package: NCS_Simu. Selected presentations. Disponível em: <http://www.sussex.ac.uk/Users/taiyang/presentation/Intro_NCS_simu.pdf>. Acesso em: Julho. 2007.

YUE, D.; HAN, Q. L.; PENG, C. (2004). State feedback controller design for networked control systems. **IEEE Transactions on Circuit Systems**, v. 51, n. 11, p. 640–644.

ZAMPIERI, S. (2008). Trends in Networked Control Systems. In: WORLD CONGRESS THE INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL, 17, 2008, Seoul, Korea. **Proceedings**. July 6-11, 2008, p. 2886-2894.

ZANATA, A. F.; GODOY, E. P.; PORTO, A. J. V. (2010). Simulação, Análise e Controle de Motor de Corrente Contínua Utilizando um Sistema de Controle via Rede CAN. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS (DINCON), 9, 2010, Bauru. **Proceedings**. Bauru : ABCM, 2010.

ZHANG, W. (2001). **Stability analysis of Networked Control Systems**. 179p., PhD Thesis, Department of Electrical Engineering and Computer Science, Case Western Reserve University, USA, August, 2001.

ZHANG, L.; HUA-JING, F. (2006). A novel controller design and evaluation for networked control systems with time-variant delays. **Journal of the Franklin Institute**. v. 343 p. 161–167.

ZHANG, W. J.; DENG, Y. F.; WANG, L. M.; ZHANG, H. W. (2008). Fuzzy PID Control Method for Networked Control System with Constant Delays. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS, 7, 2008, Kunming. **Proceedings**. 12-15 July 2008, p. 1968-1963.

ZHANG, W.; BRANICKY, M. S.; PHILLIPS, S. M. (2001). Stability of Networked Control Systems. **IEEE Control System Magazine**, v. 21(1) p. 84–99, February.

ZHAO, W.; XIA, F. (2006a). A Neural Network Approach to QoS Management in Networked Control Systems over Ethernet. **Lecture Notes in Control and Information Sciences**, n. 344, p. 444-449.

_____. (2006b). Design and Simulation of Function Block Based Networked Control Systems. In: IEEE INTERNATIONAL CONFERENCE ON INNOVATIVE COMPUTING,

INFORMATION AND CONTROL (ICICIC), 1, 2006, Beijing, China. **Proceedings**. August 30 - September 1.