

UNIVERSIDADE DE SÃO PAULO

ESCOLA DE ENGENHARIA DE SÃO CARLOS

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Desenvolvimento de um Sistema de
Controle via Rede Híbrido utilizando
CAN e ZigBee

Autor: Fernando Scorzoni

Orientador: Prof. Tit. Arthur José Vieira Porto e
Prof. Dr. Eduardo Paciência Godoy

São Carlos

2011

FERNANDO SCORZONI

Desenvolvimento de um Sistema de Controle via Rede Híbrido utilizando CAN e ZigBee

Trabalho de Conclusão de Curso
apresentado à Escola de Engenharia de
São Carlos, da Universidade de São
Paulo

Curso de Engenharia Elétrica com ênfase
em Automação e Sistemas de Energia

Orientador: Prof. Tit. Arthur José Vieira Porto e
Prof. Dr. Eduardo Paciência Godoy

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

S423d Scorzoni, Fernando.
Desenvolvimento de um sistema de controle híbrido utilizando *CAN* e *ZigBee*. / Fernando Scorzoni ; orientador Arthur José Vieira Porto -- São Carlos, 2011.

Monografia (Graduação em Elétrica com ênfase em Automação e Sistemas de Energia) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2011.

1. Sistemas de controle via rede (NCS). 2. *ZigBee*. 3. *CAN*. 4. Controle baseado em modelo (MBC). I. Título.

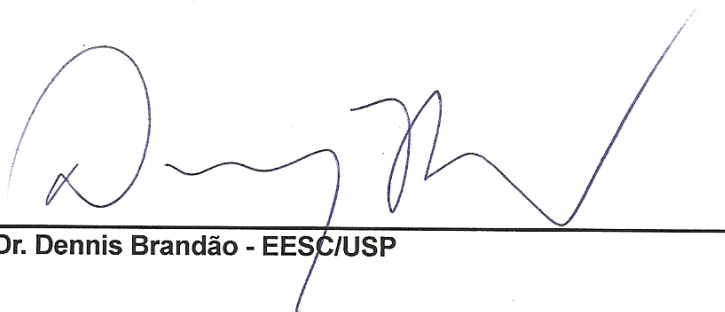
FOLHA DE APROVAÇÃO

Nome: Fernando Scorzoni

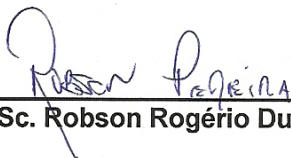
Título: “Desenvolvimento de um Sistema de Controle Via Rede Híbrido Utilizando CAN e ZigBee”

Trabalho de Conclusão de Curso defendido e aprovado
em 24 / 11 / 2011;

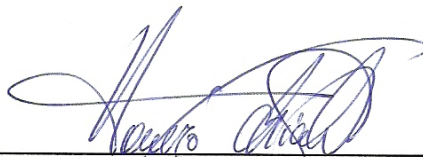
com NOTA 9,2 (nove, dois), pela comissão julgadora:



Prof. Dr. Dennis Brandão - EESC/USP



MSc. Robson Rogério Dutra Pereira - EESC/USP



Prof. Associado Homero Schiabel
Coordenador da CoC-Engenharia Elétrica
EESC/USP

Resumo

Sistemas de controle via rede (NCS – Networked Control Systems) representam uma inovadora abordagem para aplicação de sistemas de controle que utilizam redes industriais. NCS é um tipo de 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. Este projeto visa a análise, caracterização e implementação de sensores conectados a módulos ZigBee para sua implementação em um NCS baseado em rede CAN (Controller Area Network), formando assim um NCS híbrido. Analisou-se uma estratégia de controle baseado em modelo que reduz os problemas causados pela utilização de protocolos de comunicação sem fio em controle como a perda de dados. Esta estratégia utilizará um modelo matemático do sistema para fornecer os dados perdidos durante os ensaios garantindo assim o desempenho do controle do NCS. Experimentos foram realizados em uma plataforma de pesquisa sobre NCS que validaram o objetivo deste projeto.

Palavras-Chave: Sistemas de controle via rede (NCS); ZigBee; CAN; Controle Baseado em Modelo.

Abstract

Networked Control Systems (NCS) represent an innovative approach to application of control systems using industrial networks. NCS is a type of distributed control system where sensors, actuators and controllers are physically located in separate locations and are connected through a network of industrial communication. This project concerns the analysis, characterization and implementation of sensors connected to ZigBee modules for implementation on an NCS based on CAN (Controller Area Network), thus forming a hybrid NCS. Consideration will be given a control strategy based on a model that reduces the problems caused by the use of wireless communication protocols in control as the loss of data. This strategy uses a mathematical model of the system to provide data lost during testing thereby ensuring the control performance of the NCS. Experiments were performed on a platform of NCS research that validated the objective of this project.

Key Words: Networked Control Systems (NCS); ZigBee; CAN; Model-Based Control (MBC).

Índice de Figuras

FIGURA 1 - ESTRUTURA DE UM SISTEMA DE CONTROLE VIA REDES (NCS) [2].....	15
FIGURA 2 – NÍVEIS ELÉTRICOS DO PROTOCOLO CAN (MODIFICADO DE [12])	18
FIGURA 3 – FORMATO DO QUADRO DE DADOS DO CAN 2.0A [12]	19
FIGURA 4 - FORMATO DO QUADRO DE DADOS DO CAN 2.0B [12]	19
FIGURA 5 – DEMONSTRAÇÃO DO HOP	20
FIGURA 6 – PILHA BLUETOOTH SIMPLIFICADA	22
FIGURA 7 – PILHA ZIGBEE.....	24
FIGURA 8 – TOPOLOGIAS DE UMA PAN ZIGBEE	26
FIGURA 9 – ESTRUTURA PARA ENVIO DE MENSAGEM ZB	27
FIGURA 10 – DIAGRAMA DA ECU [12]	29
FIGURA 11 – FOTO DA ECU UTILIZADA [12]	30
FIGURA 12 – FOTO DA INTERFACE BLUETOOTH UTILIZADA	31
FIGURA 13 – FOTO DA INTERFACE BLUETOOTH UTILIZADA	32
FIGURA 14 – HISTOGRAMA COM DOS TEMPOS DE ATRASO A 10 METROS COM O BLUETOOTH	34
FIGURA 15 – HISTOGRAMA COM DOS TEMPOS DE ATRASO A 10 METROS COM O ZIGBEE	35
FIGURA 16 – FOTO DA ESTRUTURA DA NCS DE CONTROLE DE VELOCIDADE	37
FIGURA 17 – ESQUEMA DE UMA NCS	39
FIGURA 18 - ESTRUTURA DE SOFTWARE DA PLATAFORMA DE NCS COM REDE CAN.....	41
FIGURA 19 - ESTRUTURA DO CONTROLADOR PID DISCRETO [10].....	43
FIGURA 20 – ESQUEMA SIMPLIFICADO DA ESTRATÉGIA DO CONTROLE BASEADO EM MODELO	45
FIGURA 21 – RESPOSTA AO DEGRAU NCS VELOCIDADE	47
FIGURA 22 – EFEITO DA PERDA DE PACOTES NO NCS VELOCIDADE	48
FIGURA 23 – RESPOSTA AO DEGRAU COM CINCO LEITURAS PERDIDAS	48
FIGURA 24 – RESPOSTA AO DEGRAU COM DEZ LEITURAS PERDIDAS.....	49
FIGURA 25 – RESPOSTA AO DEGRAU COM VINTE LEITURAS PERDIDAS.....	49

Índice de Tabelas

TABELA 1: COMPARAÇÃO ENTRE OS PROTOCOLOS SEM FIO MAIS COMUNS [5]	16
TABELA 2: DESEMPENHO DO BLUETOOTH A DIFERENTES DISTÂNCIAS	34
TABELA 3: DESEMPENHO DO ZIGBEE A DIFERENTES DISTÂNCIAS.....	35

Siglas e Abreviaturas

A/D	ANALÓGICO/DIGITAL
ACK	ACKNOWLEDGE
API	APPLICATION PROGRAMMING INTERFACE
ARQ	AUTOMATIC REPEAT REQUEST
CAN	CONTROLLER AREA NETWORK
CAN_H	REFERENTE AO FIO CAN_HIGH DE UM BARRAMENTO CAN
CAN_L	REFERENTE AO FIO CAN_LOW DE UM BARRAMENTO CAN
CPU	CENTRAL PROCESSING UNIT (UNIDADE DE PROCESSAMENTO CENTRAL)
CRC	CYCLIC REDUNDANCY CHECK – VERIFICAÇÃO CÍCLICA DE REDUNDÂNCIA
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
FEC	FORWARD ERROR CORRECTION
FHSS	FREQUENCY HOPPING SPREAD-SPECTRUM – SALTO DE FREQUÊNCIA COM ESPALHAMENTO ESPECTRAL
IEEE	INSTITUTO DE ENGENHEIROS ELETRICISTAS E ELETRÔNICOS
ISO	INTERNATIONAL ORGANIZATION FOR STANDARDIZATION – ORGANIZAÇÃO INTERNACIONAL PARA PADRONIZAÇÃO
MAC	MEDIUM ACCESS CONTROL – CONTROLE DE ACESSO AO MEIO FÍSICO
MBC	MODEL BASED CONTROL – CONTROLE BASEADO EM MODELO
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)
NDBA	NON-DESTRUCTIVE BITWISE ARBITRATION – ARBITRAGEM NÃO DESTRUTIVA POR OPERAÇÃO LÓGICA BIT-A-BIT
OEM	ORIGINAL EQUIPMENT MANUFACTURER – EQUIPAMENTO ORIGINAL DE FABRICA
OSI	OPEN SYSTEMS INTERCONNECTION - INTERCONEXÃO DE SISTEMAS ABERTOS
PAN	PERSONAL AREA NETWORK – REDE PESSOAL
PCI	PERIPHERAL COMPONENT INTERCONNECT
PID	PROPORCIONAL INTEGRAL DERIVATIVO
PWM	PULSE WIDTH MODULATION – MODULAÇÃO POR LARGURA DE PULSO
QoS	QUALITY OF SERVICE – QUALIDADE DE SERVIÇO
RFCOMM	RADIO FREQUENCY COMMUNICATION – COMUNICAÇÃO POR RADIOFREQUÊNCIA
SPI	SERIAL PERIPHERAL INTERFACE – INTERFACE PARA PERIFÉRICO SERIAL
TTL	TRANSISTOR-TRANSISTOR LOGIC – LÓGICA TRANSISTOR-TRANSISTOR

USART UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER
USB UNIVERSAL SERIAL BUS
USP UNIVERSIDADE DE SÃO PAULO

Sumário

1. INTRODUÇÃO	13
1.1 OBJETIVOS DO TRABALHO	14
2. REVISÃO BIBLIOGRÁFICA	15
2.1 SISTEMAS DE CONTROLE VIA REDES (NCS)	15
2.2 SISTEMAS DE CONTROLE VIA REDE SEM FIO (WNCS)	16
2.3 CONTROLE BASEADO EM MODELO (MBC)	16
3. REVISÃO CONCEITUAL	18
3.1 PROTOCOLO CAN (CONTROLLER AREA NETWORK)	18
3.1.1 Descrição	18
3.1.2 Formato das Mensagens	19
3.1.3 Detecção e Tratamento de Erros	19
3.2 BLUETOOTH	20
3.2.1 Descrição	20
3.2.2 Pilha do Protocolo Bluetooth	21
3.2.3 Detecção e Tratamento de Erros	22
3.3 ZIGBEE	23
3.3.1 Descrição	23
3.3.2 Tipos de Dispositivo e topologias de rede	25
3.3.3 Detecção e Tratamento de Erros	27
4. MATERIAIS E MÉTODOS.....	29
4.1 DESCRIÇÃO DOS MATERIAIS UTILIZADOS	29
4.1.1 Electronic Control Unit (ECU)	29
4.1.2 Interface Bluetooth	31
4.1.3 Interface ZigBee	31
4.2 DEFINIÇÃO DA INTERFACE A SER UTILIZADA.....	32
4.2.1 Testes de desempenho com o Bluetooth	33
4.2.2 Teste de desempenho com o ZigBee	34
4.2.3 Conclusão dos testes de desempenho	36
4.3 DESENVOLVIMENTO	36
4.3.1 NCS Velocidade.....	36
4.3.1.1 Estrutura da NCS	36

4.3.1.2	Modelagem do NCS.....	38
4.3.2	Controlador	39
4.3.2.1	Hardware do Controlador	39
4.3.2.2	Software do Controlador.....	40
4.4	ESTRATÉGIA DO CONTROLE BASEADO EM MODELO	44
5.	RESULTADOS	46
5.1	VALIDAÇÃO DO MODELO	46
5.2	VALIDAÇÃO DA ESTRATÉGIA DE CONTROLE BASEADO EM MODELO.....	47
6.	CONCLUSÃO	51
7.	TRABALHOS FUTUROS	52
8.	REFERÊNCIAS BIBLIOGRÁFICAS.....	53

1. Introdução

As arquiteturas tradicionais de comunicação e controle ponto a ponto implementadas nos sistemas de manufatura industriais nos últimos tempos são compostas por cabeamento, conectando o computador ou dispositivo central de controle a cada sensor e atuador do sistema. Este tipo de controle tradicional e centralizado, não fornece os novos requisitos de desenvolvimento de sistemas de controle como modularidade, controle descentralizado, facilidade de diagnóstico e baixo custo [1].

Desenvolvimentos recentes de sistemas de controle distribuído, cujos sensores, atuadores e controladores são conectados por uma rede de comunicação, demonstram o surgimento de uma nova abordagem para a utilização de redes industriais. Nessa abordagem, o controlador e a planta ficam fisicamente alocados em locais separados e são conectados 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 [2]. Este tipo de implementação em sistemas industriais, onde as malhas de controle são fechadas sob uma rede de comunicação tem sido denominada de Sistema de Controle via Redes (NCS - Networked Control System) [1].

Como em um NCS a única forma do controlador atuar no sistema é através da rede de comunicação industrial, o protocolo de comunicação utilizado necessita de alta confiabilidade. O protocolo CAN (Controller Area Network) foi originalmente desenvolvido para aplicações automotivas, no entanto já é muito utilizado na área agrícola [3] e em aplicações industriais devido a sua robustez. Este protocolo prevê algoritmos de controle de erro que garantem a integridade do dado [4].

Em algumas aplicações existe a necessidade do sensor ser colocado em locais onde fica inviável ou impossível a instalação de cabos para a integração desse sensor na rede. ZigBee [5] é um protocolo de comunicação sem fio baseado na norma IEEE 802.15.4 e capaz de manter a QoS (Quality of Service) mesmo em ambientes hostis a equipamentos de RF (Radio Frequência) como o industrial [6]. Este protocolo, assim como o CAN, possui algoritmos que garantem a integridade do dado, no entanto possui uma latência que em alguns sistemas de controle pode ser considerada alta e inviável.

1.1 Objetivos do trabalho

Este trabalho tem como objetivo a integração de uma rede de sensores ZigBee em um NCS baseado no protocolo CAN, formando assim uma rede híbrida com dois protocolos de comunicação muito pesquisados atualmente. Esta integração está dividida em duas partes, sendo uma focada no estudo e implementação de uma rede de sensores ZigBee e outra focada no desenvolvimento do sistema de controle a fim de validar o projeto.

Um estudo sobre os protocolos ZigBee e Bluetooth são apresentados, bem como trabalhos em baixo nível, tais como a criação de bibliotecas para os dispositivos utilizados, que elevarão o nível de programação e facilitarão aplicações futuras. Experimentos são realizados buscando avaliar o desempenho dos dispositivos ZigBee e Bluetooth para aplicações de controle via rede.

Redes de sensores sem fio apresentam dois importantes problemas que são as variações da ordem de milissegundos no tempo de transmissão e a perda de informações transmitidas na rede. Como esses problemas podem comprometer o controle do NCS, uma metodologia de controle baseado em modelo é desenvolvida buscando superá-los. Para isso são criados modelos dos NCS a fim de prover dados simulados na ausência dos reais, que auxiliem na obtenção de um sistema de controle mais preciso e estável para o NCS.

2. Revisão Bibliográfica

2.1 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 [7], também chamados de sistemas de controle via redes (NCS). Na Figura 1 temos um exemplo na qual 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.

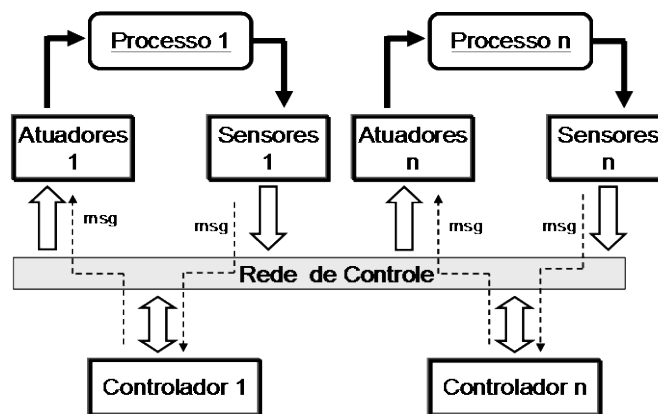


Figura 1 - Estrutura de um sistema de controle via redes (NCS) [2]

Os atuadores e sensores podem ser conectados à mesma unidade de controle eletrônico (ECU) ou serem independentes na qual cada dispositivo fica responsável por disponibilizar o dado na rede de controle. Para um bom resultado no controle via rede o protocolo de comunicação é determinante. Um grande número de protocolos de comunicação têm sido desenvolvido e aplicado em NCS [8], como o Profibus, CAN e Foundation Fieldbus. Dentre esses protocolos, o CAN [4] originalmente desenvolvido para aplicações em na área automotiva, se encontra atualmente muito difundido e tem sido utilizado em muitas outras aplicações industriais de controle [9].

2.2 Sistemas de controle via rede sem fio (WNCS)

Pesquisas em sistemas de controle via rede que utilizam protocolos sem fio na rede de controle vem aumentando devido a facilidade de implementação e manutenção além da inexistência de conexões por cabeamento. Apesar da tendência e das facilidades, a utilização de protocolos sem fio implica em alta latência, jitter (variação entre as latências de sucessivos pacotes), perda e colisão de pacotes e como consequência uma baixa amostragem quando comparada com protocolos com fio. Esta ultima (taxa de amostragem) foi mostrado por [10] como sendo o determinante para um bom desempenho e estabilidade do sistema de controle.

Um protocolo sem fio que vem sendo muito pesquisado recentemente é o ZigBee [5] que é baseado na norma IEEE 802.15. Foi concebido a principio para monitoramento a atuação com baixo consumo de energia priorizando a integridade do dado. Assim a latência é considerada alta comparada com outros protocolos sem fio como é mostrado na tabela 1.

Tabela 1: Comparação entre os protocolos sem fio mais comuns [5]

Wi-Fi 802.11 b/g	Bluetooth	ZigBee
~3ms	~10ms	~30ms

Para alguns sistemas de controle esta latência é considerada inviável, no entanto existem métodos para que se consiga contornar essa escassez de dados como a utilização de modelos em paralelo com o processo de forma a prover dados simulados na ausência dos reais.

2.3 Controle baseado em modelo (MBC)

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 [11]. 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*¹, 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 estimativa deste modelo pode ser realizada através de dados experimentais.

¹ *Multi-rate* – Esta estratégia intercala dados reais dos sensores com dados simulados de forma periódica a fim de aumentar a taxa de amostragem.

3. Revisão Conceitual

3.1 Protocolo CAN (Controller Area Network)

O protocolo CAN, como desenvolvido por Robert Bosch, foi padronizado e documentado internacionalmente pela ISO, gerando a Norma ISO 11898, para aplicações de alta velocidade, e a Norma ISO 11519, para aplicações de baixa velocidade [12]

3.1.1 Descrição

O CAN é um protocolo de comunicação serial síncrono no qual o sincronismo dos módulos conectados à rede se dá com relação ao início de cada mensagem.

As mensagens são enviadas em regime multicast, ou seja, toda e qualquer mensagem é enviada para todos os da rede. Os módulos da rede verificam o estado do barramento, analisando se outro módulo está ou não enviando mensagens com maior prioridade. Caso isto seja constatado, o módulo cuja mensagem tiver menor prioridade cessará sua transmissão e o de maior prioridade continuará enviando sua mensagem deste ponto, sem ter que reiniciá-la, tal característica é fundamentada no conceito CSMA/CD *with NDA* (*Carrier Sense Multiple Access / Collision Detection with Non-Destructive Arbitration*).

No CAN os dados são representados por bits dominantes e bits recessivos (nível lógico “0” e “1” respectivamente) e são definidos por sinais diferenciais formados pelo fios CAN_H e CAN_L.

O nós da rede devem detectar uma condição de bit recessivo quando a tensão do CAN_H não é 0,5V maior que a tensão do CAN_L. Caso a tensão do CAN_H e de pelo menos 0,9 V superior ao CAN_L temos um bit dominante. Em um estado dominante a tensão do CAN_H de 3,5 V e para o CAN_L de 1,5 V, como pode ser observado na Figura 2.

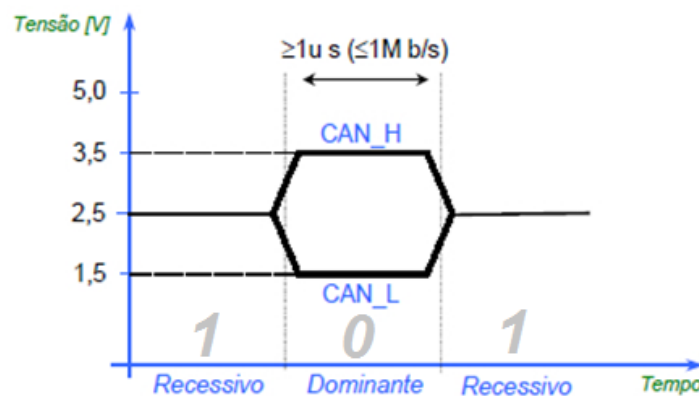


Figura 2 – Níveis elétricos do Protocolo CAN (modificado de [12])

Os níveis de tensão podem variar de acordo com o protocolo baseado em CAN utilizado. Porém o mais importante de ser observado nesta forma de transmissão de dado é o fato do dado que está sendo transmitido ter alta imunidade a interferências eletromagnéticas externas já que ambos os fios estão expostos ao mesmo ruído, logo o sinal diferencial se mantém inalterado.

3.1.2 Formato das Mensagens

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 de 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). Ambos estão representados nas figuras 3 e 4 respectivamente.

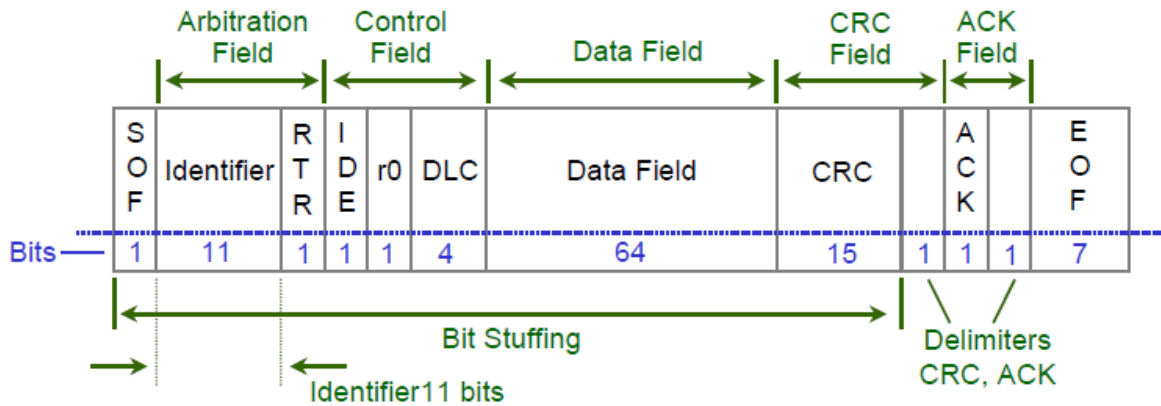


Figura 3 – Formato do quadro de dados do CAN 2.0A [12]

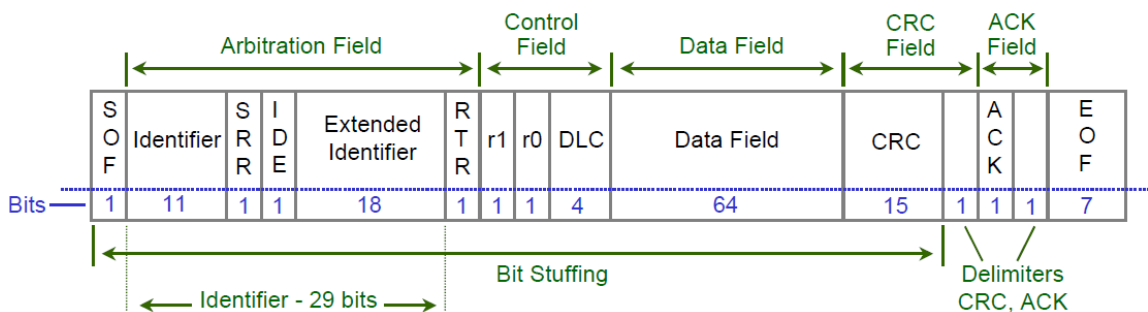


Figura 4 - Formato do quadro de dados do CAN 2.0B [12]

3.1.3 Detecção e Tratamento de Erros

A verificação de erros é principalmente feita por duas técnicas chamada *Bit Stuffing* e o CRC (Cyclic Redundancy Check). Na primeira após cada conjunto de cinco

bits iguais é inserido um sexto bit de forma a mostrar que tais bits não são provenientes de algum ruído o erro na transmissão. No CRC, a partir de uma função *hash*, são feitos cálculos de seis em seis bits de forma que o resultado seja colocado num campo de dois bytes da mesma mensagem. Caso a mensagem seja recebida com erro é enviado uma mensagem de erro ao transmissor. Caso não tenha erro o dispositivo receptor sobrescreve o campo ACK da mensagem com bits dominantes de forma que o transmissor tenha a informação que a mensagem chegou com sucesso.

3.2 Bluetooth

Bluetooth é essencialmente um protocolo para conectividade sem fio criado pela Ericsson em 1994. foi originalmente concebida como uma alternativa sem fio para cabos RS-232 de dados, mas hoje a sua utilização em diversas aplicações por causa de sua segurança e taxa de dados grande (em comparação com outras tecnologias sem fio).

3.2.1 Descrição

O sistema Bluetooth adota uma comunicação baseada em salto de frequência com espalhamento espectral (FHSS). Trata-se de uma técnica que espalha o dado por uma banda muito maior do que a necessária para a sua transmissão. Para tal, FHSS divide a banda total em vários canais de pequena largura de banda. Desta forma, transmissor e receptor saltam por estes canais de acordo com uma sequência pseudo-aleatória conhecida por ambos e obtida por um algoritmo baseado no endereço físico de cada dispositivo.

No caso do Bluetooth são utilizadas até 79 frequências diferentes com largura de banda de 1MHz entre 2400 MHz e 2483,5 MHz. Durante a operação são realizados 1600 saltos por segundo. Tais saltos estão exemplificados na figura 5.

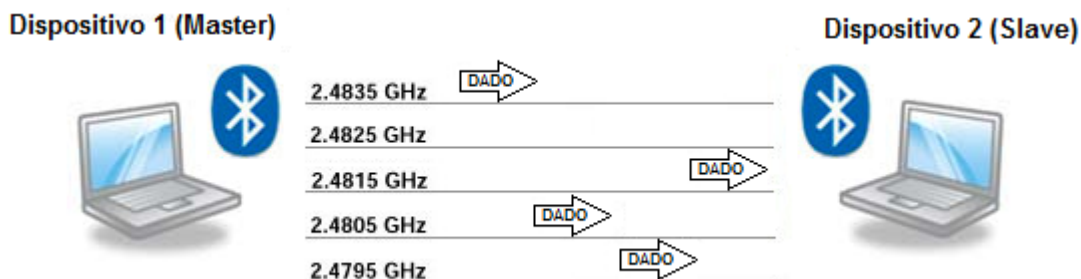


Figura 5 – Demonstração do hop

Isto dá pacote de comprimento relativamente curto e boa proteção contra interferência [13]. Bluetooth utiliza uma comunicação mestre-escravo baseado protocolo MAC (Media Access Control) com até oito unidades de Bluetooth podem ser interligados em um assim chamado Piconet². A distância máxima entre os dispositivos varia de 1 a 100 metros de acordo com a classe dos mesmos.

3.2.2 Pilha do Protocolo Bluetooth

O protocolo bluetooth é composto basicamente por:

-Radio: é o responsável pela conversão do sinal de radio frequência em sinal elétrico e vice-versa.

-Baseband: é a camada física do Bluetooth que gerencia canais físicos e links para além de outros serviços como a correção de erros, seleção de hop e segurança Bluetooth.

-Link Manager: Fornece serviços como autenticação, controle de criptografia, controle de energia e fornece capacidades de QoS. Ele também gerencia os dispositivos em diferentes modos de operação.

-L2CAP (Logical Link Control and Adaptation Layer): Encontra-se na camada de enlace de dados e tem como função a segmentação e remontagem de pacotes, qualidade do serviços (QoS) para protocolos de camada superior, multiplexação de dados entre diferentes protocolos de camada superior.

Tais protocolos podem ser organizados na forma de pilha como na figura 6. Note que esta pilha esta resumida apenas aos componentes necessários para esta aplicação já que a pilha Bluetooth completa é relativamente complexa. Nesta figura podemos observar também a distribuição de tais protocolos no modelo ISO OSI (Open Systems Interconnection).

² Piconet - Uma rede do tipo *ad hoc* de dispositivos conectados usando a tecnologia Bluetooth.

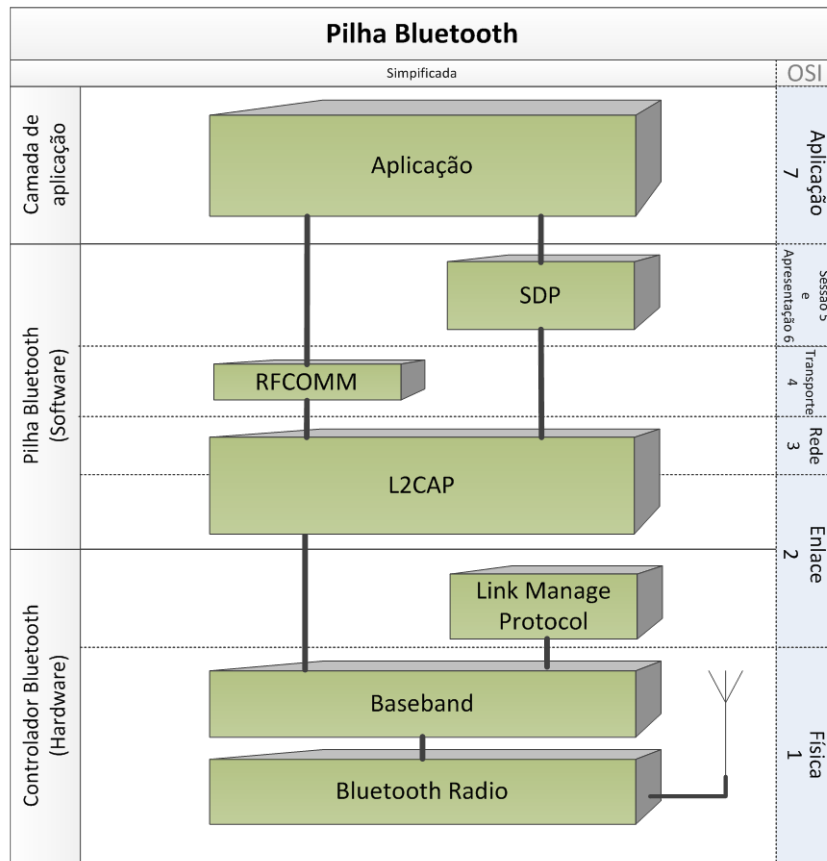


Figura 6 – Pilha Bluetooth simplificada

Acima do L2CAP está situado o SDP (Service Discovery Protocol) que tem como função descobrir quais serviços estão disponíveis para assim determinar as características dos serviços que o dispositivo irá fornecer na piconet. Nessa mesma camada encontra-se o RFCOMM (Radio Frequency Communication) que trata-se de um simples conjunto de protocolos de transporte que emula portas RS232 nos dispositivos (até sessenta conexões simultâneas a um dispositivo Bluetooth de cada vez). Utilizado como “substituidor de cabo”, tal protocolo promove uma comunicação transparente entre os pertencentes à piconet, permitindo que um dado seja transmitido via RS232 sem a utilização de fios e a uma distância maior que poderia se por cabos comuns.

3.2.3 Detecção e Tratamento de Erros

Além do FHSS existem outros três métodos para aumentar a imunidade a ruído do sinal. Tais métodos além de detectar o erro, o corrigem. São eles: “1/3 rate FEC”, “2/3 rate FEC” and “ARQ scheme”. Em “1/3 rate FEC” cada bit é repetido três vezes para a

redundância, em “2/3 rate FEC” um polinômio gerador é usado para codificar código de 10 bits a um código de 15 bits, e no “ARQ scheme” um pacote é retransmitido até que uma confirmação (*acknowledge* - ACK) é recebida (ou tempo limite for excedido). Este ultimo só atua após a verificação dos dois anteriores diminuindo assim o numero de retransmissões.

Embora bem sucedido em reduzir o número de retransmissões, a utilização FEC diminui significativamente as transmissões (devido ao tempo demandado para os calculos) e não é, portanto, utilizados em ambientes relativamente livres de ruídos. Os cabeçalhos dos pacotes são exceção a isso, ja que eles sempre fazem uso de FEC. Os pacotes trocados sobre o protocolo RFCOMM exigem do dispositivo remoto o ACK e este é reconhecido transmissor. No entanto na ausência do ACK o dispositivo transmissor não retransmite a mensagem. Dessa forma o “ARQ scheme” não atua no protocolo RFCOMM [14].

Portanto, RFCOMM deve exigir L2CAP para fornecer canais com o máximo de confiabilidade a fim de garantir que todos os pacotes sejam entregues corretamente, e sem duplicatas. Caso o L2CAP não fornecer isso o RFCOMM vai esperar uma notificação perda de link, que deve tratado pelo mesmo.

3.3 ZigBee

O protocolo ZigBee foi desenvolvido pela ZigBee Alliance para aplicações *wireless* de baixo custo e baixo consumo de energia. Tais características tornaram o ZigBee um protocolo muito pesquisado [6] por apresentar características inovadoras e aplicáveis em diversas áreas como automação industrial e residencial, brinquedos, sensores médicos, sensoriamento agrícola, dentre outros [15].

3.3.1 Descrição

A norma IEEE 802.15.4 define as duas camadas inferiores:

-Camada Física (PHY): É onde está o rádio responsável pela transmissão e recepção dos sinais nas frequências 2,4GHz, 915MHz ou 868MHz dependendo da região.

-Medium Access Control (MAC): É a camada de enlace de dados.

Utilizando essas camadas como base, a ZigBee Alliance desenvolveu:

-Camada de Rede (NWK): Responsável pelo gerenciamento da segurança, de rede e do roteamento, enfim, garantir o funcionamento das camadas inferiores e fornecer suporte às superiores.

-Sub-camada de Suporte à de Aplicação (APS): É responsável pela interface entre a camada de Aplicação (APL) e a Camada de Rede.

-ZigBee Device Objects (ZDO): Responsável por intermediar a atuação da camada de aplicação na APS e na NWK promovendo uma melhor integração entre a camada de aplicação e inferiores.

A organização dessas camadas pode ser observada na figura 7.

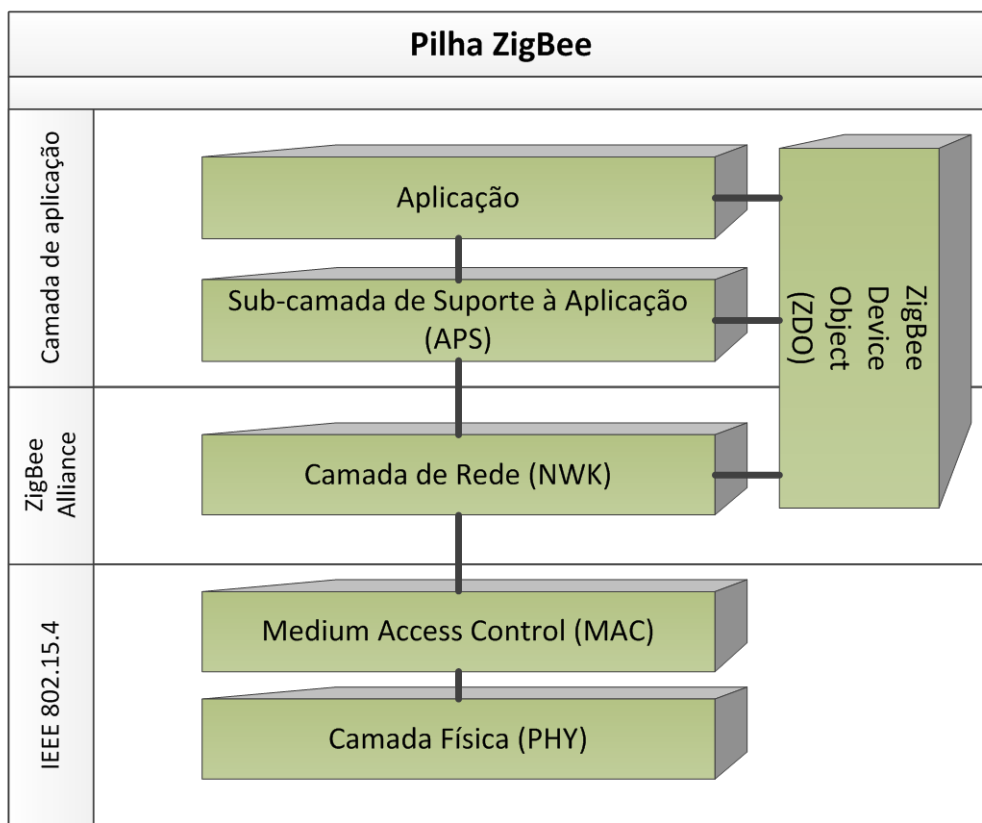


Figura 7 – Pilha ZigBee

O que diferencia o ZigBee das demais tecnologias de rede sem fio é o baixo consumo de energia e a possibilidade de hospedar até 54 mil dispositivos em uma única PAN (Public Area Network) com diversas topologias. Gotemburgo na Suécia possui a maior rede baseada em ZigBee do mundo [16] com 262.000 medidores de energia elétrica divididos em 8900 PAN que se conectam a central por fibra óptica ou GPRS.

3.3.2 Tipos de Dispositivo e topologias de rede

Existem três classes que um dispositivo ZigBee pode assumir:

- Coordenador: O coordenador é responsável pela inicialização, distribuição de endereços, manutenção da Rede, reconhecimento de todos os Nós, entre outras funções podendo servir como ponte entre várias outras Redes ZigBee. Uma PAN ZigBee necessita obrigatoriamente de apenas um coordenador para atribuir os endereços de todos os dispositivos encontrados e distribui tabelas de roteamento aos roteadores. A rede é capaz de manter o serviço caso o coordenador seja retirado depois de estabelecidos os endereços e as tabelas de roteamento. Para isso os roteadores se rearranjam de forma que as mensagens que passavam pelo coordenador encontrem outros caminhos para chegar ao seu destino.

- Roteador: Funciona como um nó na rede, é responsável rotear ou apenas repetir os dados que chegam a ele. Quanto este dispositivo ingressa numa rede ele reconhece todos os dispositivos que estão próximos dele. Dessa forma o dispositivo pode escolher o melhor caminho para o dado percorrer.

- Dispositivo Final: É onde estão conectados os sensores e atuadores. Um dispositivo final necessita de um “pai” que pode ser um coordenador ou um roteador. Todas as mensagens que entram e saem do dispositivo final apadrinhado terá obrigatoriamente que passar pelo seu “pai”. Esta classe é a única em que é permitido a função “dormir”, na qual o dispositivo pode ficar em um estado de baixo consumo de energia (dormindo) por um determinado período (máximo de 45 segundos e mínimo de 5 segundos no caso dos módulos da Digi) e apenas “acordar” para enviar ou receber as mensagens para/do seu “pai”. É importante citar que o dispositivo final não “acorda” ao receber uma mensagem, ele acorda ciclicamente com o período determinado. Enquanto o dispositivo final estiver dormindo seu “pai” é responsável por armazenar os pacotes recebidos por um determinado tempo que é configurado e, por lógica, tem que ser maior que o período em que o dispositivo estiver dormindo. Esta funcionalidade permite que um dispositivo final funcione por até anos com bateria.

O protocolo ZigBee define uma estrutura de rede que incorpora os conceitos de redes *ad hoc*, as características de conexão em malha e *multi-hop*. As topologias que podemos encontrar em uma rede Zigbee são a topologia em estrela, em árvore e em malha.

A figura 8 mostra as possíveis topologias:.

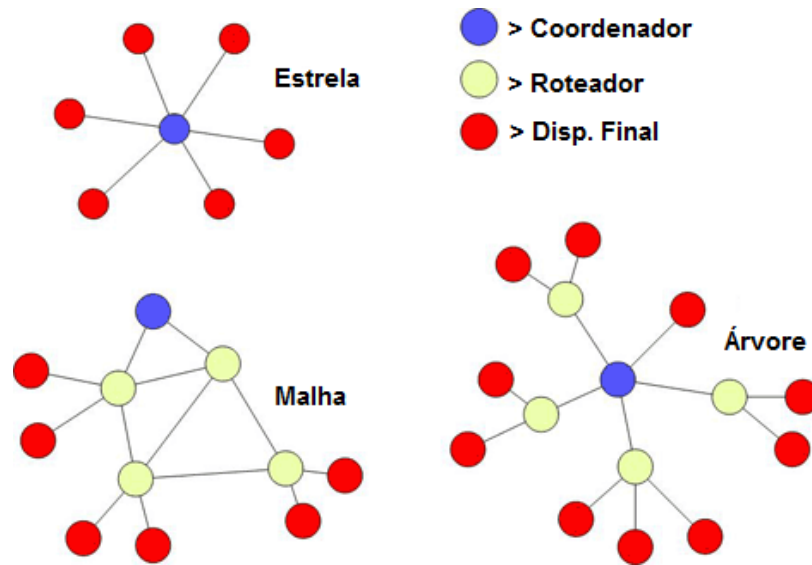


Figura 8 – Topologias de uma PAN ZigBee

Existem dispositivos ZigBee de vários fabricantes: Atmel, Freescale, Texas Instrument, Microchip, Digi, etc. A única diferença entre eles está na camada de aplicação onde cada fabricante desenvolve procedimentos específicos para o controle do dispositivo.

Os módulos da Digi, chamados de XBee, possuem duas formas de operação (API e AT) que possuem suas vantagens e desvantagens de acordo com a aplicação. Um módulo XBee pode ser AT ou API simplesmente mudando o *firmware*.

Com um firmware AT o dispositivo é capaz de fazer uma comunicação transparente sempre e apenas com um dispositivo. Para se comunicar com outro é necessário a reconfiguração do mesmo. Este modo de operação é bem semelhante ao RFCOMM do Bluetooth, funciona muito bem como substituidor de cabo. No entanto, é impossível saber a origem do dado, se o pacote chegou ou não ao destino (ACK), por que o pacote não chegou, por onde o pacote passou (no caso de uma rede em malha), dentre outras funções. Enfim, no modo AT o usuário tem poucas opções de interagir com as funcionalidades que a ZigBee Alliance disponibiliza.

O firmware API, diferentemente do AT, utiliza mensagens estruturadas de forma que para enviar uma mensagem para algum dispositivo da PAN, por exemplo, é preciso enviar um frame cujos campos podem ser vistos na figura 9.

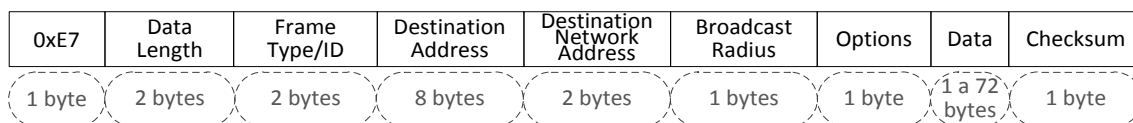


Figura 9 – Estrutura para envio de mensagem ZB

Data Length é o numero de bytes entre o campo seguinte e o *Checksum*. *Frame Type* é o tipo de frame, por exemplo este é para enviar uma mensagem a algum dispositivo ZigBee na PAN. Existe o frame para receber mensagem ZigBee, para receber status da transmissão (uma espécie de *acknowledge* de alto nível) , para executar comandos AT em dispositivos com este *firmware*, para receber amostragem das portas I/O e conversores A/D dentre outros. *Destination Address* é o campo que corresponde a um endereço do dispositivo de destino. Tal endereço é único e definido na sua fabricação. *Destination Network Address* é o endereço do dispositivo na PAN. O do coordenador é sempre uma sequencia de quatro zeros e o endereço dos outros dispositivos são definidos pelo coordenador. *Broadcast Radius* define o máximo de dispositivos que uma mensagem em *broadcast* poderá atingir. *Data* é o dado a ser transmitido e *checksum* é o *byte* menos significativo da soma de todos os *bytes* do *frame*.

Note que a complexidade do modo de operação API é maior, no entanto suas possibilidades são bem maiores. Os módulos XBee possuem para utilização cinco portas I/O dentre as quais três podem ser entradas para conversor A/D. O módulo pode ser configurado para amostrar a leitura dessas portas a um período estabelecido de 50 a 65535ms.

Outra vantagem dos módulos Xbee é o fato de todas as configurações relativas a software (atualização de firmware e configuração de parametros) pode ser feita através da própria rede ZigBee por qualquer dispositivo da PAN. Basta conectar este a um computador e executar os procedimentos necessários.

3.3.3 Detecção e Tratamento de Erros

O principal método de correção de erros utilizado no protocolo ZigBee para correção de erros é o ARQ que funciona de forma similar ao do Bluetooth. O transmissor retransmite um pacote até que o receptor responda com um ACK confirmando a correta recepção. Esse processo é controlado pela camada de rede, no entanto com o firmware API é possível ter acesso a essa e outras informações como recebimento ou não da mensagem e por que. O dispositivo, ao enviar uma mensagem na rede, retorna a seu controlador uma mensagem estruturada com o status da transmissão. Com esta

mensagem é possível saber se a mensagem transmitida chegou corretamente, quantas tentativas foram necessárias para que isso ocorresse e caso não tenha chegado, o motivo da falha.

4. Materiais e Métodos

4.1 Descrição dos Materiais Utilizados

Para a implementação dos dispositivos sem fio no sistema de controle via rede foi necessário a utilização de ECU (*Eletronic Control Unit*) além do desenvolvimento de bibliotecas para os dispositivos utilizados.

4.1.1 Electronic Control Unit (ECU)

A unidade de controle eletrônico ECU [12] será responsável pela aquisição de dados (sensor), atuação (atuador) e pela comunicação nos protocolos escolhidos (CAN, ZigBee Bluetooth). Como os dispositivos ZigBee e Bluetooth que utilizamos se comunicam por RS232, um controlador CAN é a única característica especial necessária. A figura 10 mostra o diagrama da ECU utilizada.

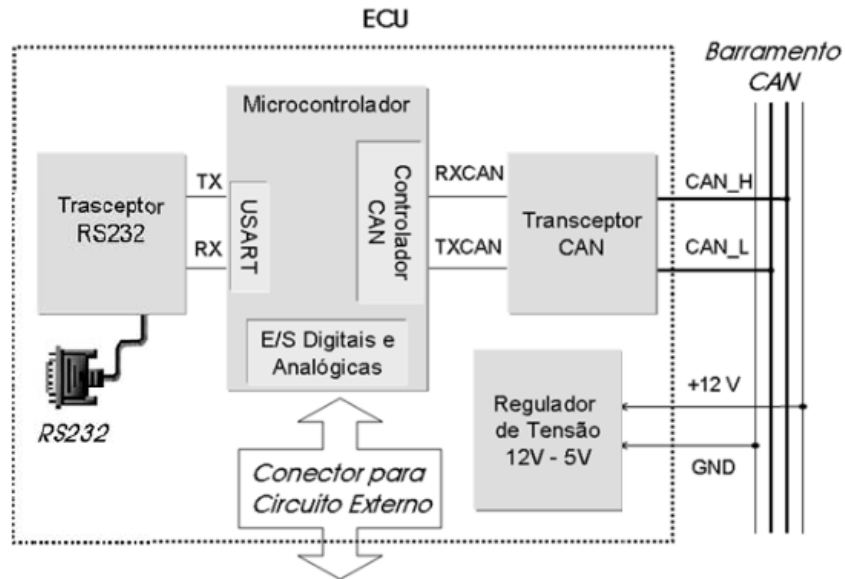


Figura 10 – Diagrama da ECU [12]

Podemos ver na figura 11 uma foto da ECU com legenda em seus principais componentes.

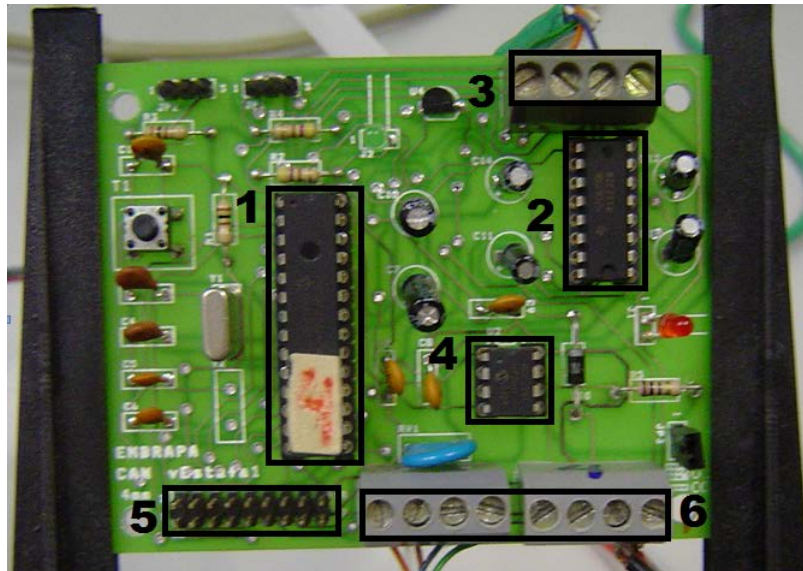


Figura 11 – Foto da ECU utilizada [12]

1. Microcontrolador com controlador CAN (PIC 18F2580 da Microchip [17]).
2. Circuito integrado MAX232 que faz a conversão do sinal TTL (do microcontrolador) para ± 12 do RS232.
3. Conexão RS232.
4. Circuito integrado MCP2551 que condiciona o sinal para que o microcontrolador possa se comunicar com o barramento CAN.
5. Conexões para acesso aos pinos do microcontrolador (Entradas analógicas, I/O, SPI, I²C, PWM entre outras)
6. Conexões do barramento CAN a 4 fios (GND, VCC, CAN_H, CAN_L)

O microcontrolador (PIC18F2580) utilizado é um PIC de baixo custo cuja principal característica é o controlador CAN, no entanto este apresenta conversor A/D de 10 bits integrado e disponível em oito portas, memória Flash de (32kb), uma porta CCP (Capture/Compare/PWM) dentre outras funcionalidades que tornam este PIC uma ótima opção para a aplicação pretendida.

A fim de tornar elevar o nível de programação e facilitar aplicações futuras foram desenvolvidas bibliotecas para o microcontrolador e LabView. Dessa forma foi possível a realização de testes o observar o funcionamento e as limitações dos módulos Bluetooth e ZigBee.

A programação de todos os micrcontroladores foi feita utilizando o ambiente de desenvolvimento MPLAB com o compilador C18, ambos da Microchip [17].

4.1.2 Interface Bluetooth

A interface Bluetooth utilizada pode ser vista na figura 12.



Figura 12 – Foto da interface bluetooth utilizada

Vendida no Brasil pela Naxos Tecnologia, a interface Bluetooth trata-se de um sistema embarcado para substituição de cabo no padrão RS232 através do protocolo RFCOMM. O fabricante forneceu um programa onde é possível fazer configurações como o modo de operação (mestre ou escravo), endereço do mestre, senha para conexão dentre outras.

O módulo cumpre o seu papel como substituidor de cabo RS232, no entanto, o fato desse ser projetado apenas para tal, trouxe uma carência de informações que seriam importantes no desenvolvimento do projeto. Informações com o recebimento ou não do *acknowledge*, QoS e status do *buffer* estão disponíveis para acesso num modem Bluetooth, sem eletrônica embarcada, que não é o caso.

Como a comunicação é transparente, não é possível identificar o endereço do dispositivo que esta enviando o dado. Dessa forma foi criado uma biblioteca para o microcontrolador utilizando (PIC18F2580) e para o LabView de forma que eles se comuniquem através de mensagens estruturadas com campos pré-estabelecidos onde é possível identificar os dispositivos.

4.1.3 Interface ZigBee

Já a interface ZigBee foi utilizado o XBee da Digi [18] mostrado na figura 13.

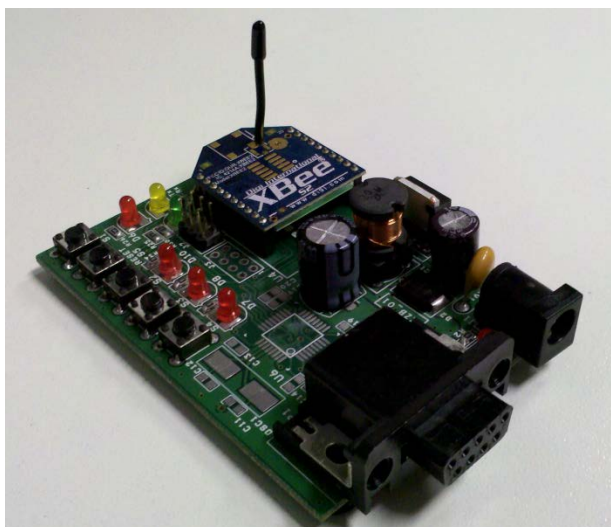


Figura 13 – Foto da interface bluetooth utilizada

Como já foi dito na revisão conceitual XBee trata-se de um módulo de um módulo ZigBee com um firmware que oferece recursos extras. Dentre os dois firmwares oferecidos pela Digi (API ou AT), optamos pelo API que fornece melhores funcionalidades.

Para o modelo em questão são oferecidos três tipos de firmware API, o coordenador, roteador e dispositivo final. Como nosso objetivo é controle, que exige menor atraso, utilizamos apenas conexões ponto-a-ponto entre coordenador e roteador. Assim como no Bluetooth foram criadas bibliotecas para o microcontrolador PIC18F2580 e para o LabView. Tal biblioteca foi criada aproveitando das funcionalidades que o firmware dispõe. Dentre as funcionalidades podemos citar como principal a confirmação de recebimento de uma mensagem (*acknowledge*) que é disponibilizado para o microcontrolador tal qual o motivo do não recebimento, ficando a critério do programador o tratamento deste problema.

4.2 Definição da Interface a ser utilizada

A fim de conhecer as limitações e características de cada dispositivo foram feitos testes de desempenho na qual foram observadas as características que mais afetam o controle via rede tais como a latência, o jitter e a amostragem. De acordo com [19] as métricas principais que caracterizam o desempenho do NCS são tempo de atraso médio e jitter.

O mais crítico destas métricas é o tempo que decorre entre o envio de dados em um nó e recebê-la em outro ou o tempo de atraso.

É difícil medir experimentalmente o intervalo de tempo levado por um nó para processar uma mensagem e transmiti-lo. Consequentemente, a única forma de medir esse atraso é através do tempo de ida e do dado. Se um cálculo estatístico é realizado a partir de uma quantidade de atrasos coletados, podemos obter o tempo de atraso médio de tempo entre as transmissões de mensagens na rede.

A variação nos valores medidos do tempo de atraso é o jitter (o desvio padrão dos valores). Dependendo do jitter o desempenho NCS pode ser significativamente afetado. O pior valor da distribuição de atrasos dará o valor mínimo da amostragem que poderia ser usado para o controle da NCS em malha fechada.

4.2.1 Testes de desempenho com o Bluetooth

Como metodologia deste teste, utilizaremos como as métricas de desempenho a metade do tempo de atraso médio de ida e volta do dado e a metade do jitter. Outro módulo Bluetooth foi conectado na mesma piconet para aumentar o tráfego de mensagens e compartilhar a largura de banda.

Para a realização do experimento foi utilizado um conector *loop-back* que não faz nada além de curto-circuitar os pinos Rx (pino utilizado para receber os dados) e Tx (pino utilizado para transmitir os dados). Dessa forma quando um dado é recebido, ele é imediatamente enviado de volta.

Foi feito um programa no LabView que envia bytes aleatórios e aguarda o recebimento do mesmo registrando o tempo entre o envio e recebimento. Caso não houvesse retorno do byte, este era contado como dado perdido. Foram feitas coletas de mil amostras e a distribuição pode ser vista no gráfico da figura 14. Observarmos que a maioria das coletas se aproximam do valor médio que é de 75,62ms.

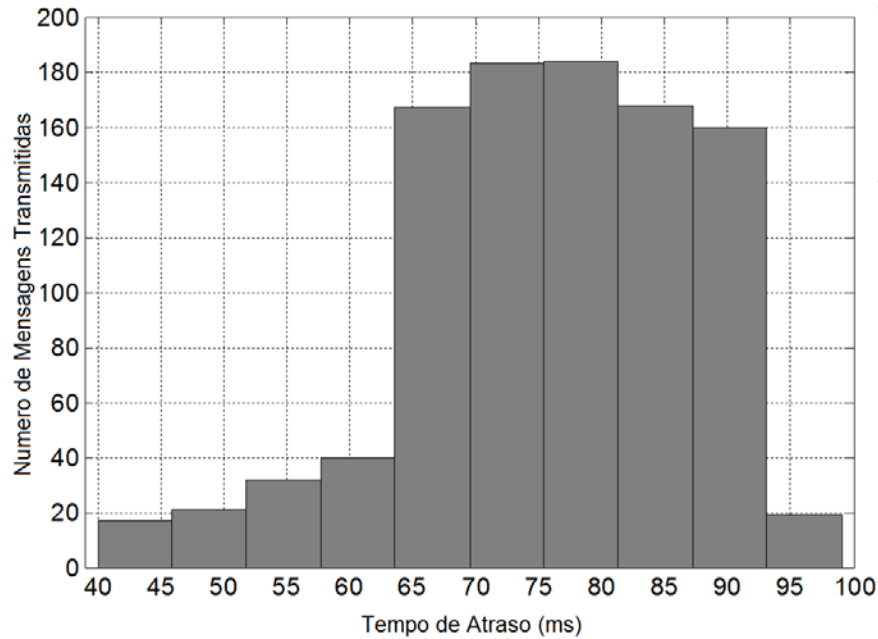


Figura 14 – Histograma com dos tempos de atraso a 10 metros com o Bluetooth

Note que nessa figura os valores são do atraso *round-trip* (Ida e volta) do dado, já na tabela 2 são os valores de “ida”. A Tabela 2 apresenta um resumo do desempenho para diferentes distâncias do módulo Bluetooth utilizado.

Tabela 2: Desempenho do Bluetooth a diferentes distâncias

Distância (m)	Atraso (ms)	Jitter (ms)	Pior (ms)	Mensagens Perdidas (%)
5	37,80	5,23	49,5	0,9%
10	37,81	5,43	49,5	1,2
20	38,33	6,38	50	4,6

De acordo com os dados da Tabela 2, os atrasos médios do módulo Bluetooth é de cerca de 38ms com 5,5 ms de jitter. Não houve diferença significativa em termos de tempo de atraso e jitter ao variarmos a distância, no entanto o percentual de mensagem perdida cresceu de 0,9 para 4,6%.

4.2.2 Teste de desempenho com o ZigBee

Os testes de desempenho com o ZigBee foram feitos de forma bem parecida para atermos uma boa base de comparação. Foi feito um programa para o microcontrolador em que ele retornava o dado recebido. Não foi possível a utilização do “loop-back” pois

no firmware API os frames de envio e recebimento são diferentes. No entanto o tempo que o PIC demora para receber e responder está na ordem de microssegundos, o que não influencia na comparação com o Bluetooth.

Foi feito, também um programa no LabView que envia bytes aleatórios e aguarda o recebimento do mesmo registrando o tempo entre o envio e recebimento. Caso não houvesse retorno do byte, este era contado como dado perdido. Foram feitas coletas de mil amostras e a distribuição pode ser vista no gráfico da figura 15.

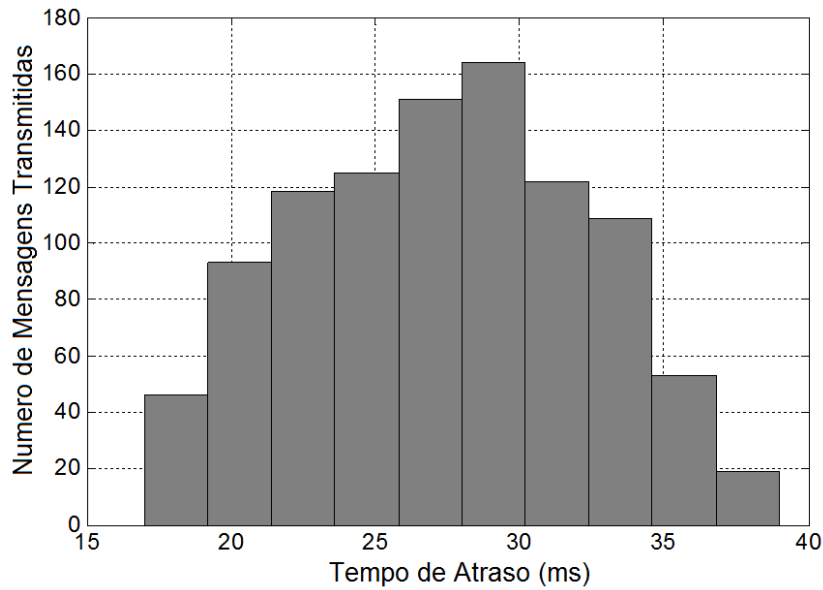


Figura 15 – Histograma com dos tempos de atraso a 10 metros com o ZigBee

A partir deste histograma obtemos um valor de médio de 27,38ms. Note que nessa figura os valores são do atraso *round-trip* do dado, já na tabela 3 são os valores de “ida”. A Tabela 3 apresenta um resumo do desempenho para diferentes distâncias do módulo ZigBee utilizado.

Tabela 3: Desempenho do ZigBee a diferentes distâncias

Distância (m)	Atraso (ms)	Jitter (ms)	Pior (ms)	Mensagens Perdidas (%)
5	14,22	2,5	22,5	0
10	14	2,46	19,51	0
20	14,12	2,48	21,6	0

Assim como no Bluetooth, não houve diferenças significativas no atraso e jitter quando a distância variou. No entanto o ZigBee não apresentou nenhuma mensagem perdida.

4.2.3 Conclusão dos testes de desempenho

De acordo com os resultados foi possível observar a superioridade do ZigBee, já que este obteve melhores resultados em todos os parâmetros medidos. Os resultados inferiores do Bluetooth podem ser explicados principalmente pelo hardware embarcado no módulo utilizado. Em outros trabalhos como [20], que fez testes com um módulo bluetooth OEM (Original Equipment Manufacturer), obteve um atraso *round-trip* de 36,4ms, que é aproximadamente a metade do atraso obtido neste trabalho.

Mas de qualquer forma é um resultado inferior ao do ZigBee que foi a nossa escolha para a implementação nos Sistemas de Controle via Rede.

4.3 Desenvolvimento

Em [10] foi desenvolvido uma plataforma de pesquisa em controle via rede baseada em CAN. Tal plataforma trata-se de uma bancada que representa um pequeno sistema de manufatura. Este sistema é composto por diferentes malhas de controle fechadas geralmente utilizadas na área industrial.

Neste projeto utilizarei esta plataforma, no entanto a realimentação (sensores) das malhas de controle dos sistemas será feita sem fio via ZigBee.

Como foi dito na revisão bibliográfica, dispositivos ZigBee não foram desenvolvidos para amostragem rápida e latência baixa, como deveria ser uma rede utilizada em controle. No entanto os resultados observados no teste de desempenho mostraram que esta aplicação é possível. Serão testadas também estratégias para diminuir os efeitos das limitações do ZigBee.

4.3.1 NCS Velocidade

Para os testes foi utilizado um NCS desenvolvido em [10] de controle de velocidade de um motor DC realimentado por um encoder de 600 pulsos.

4.3.1.1 Estrutura da NCS

Para a realização dos testes foi utilizado um NCS composto basicamente por um kit didático da T&S Equipamentos [21] adaptado para a aplicação.

Podemos observar na figura 16 seus principais componentes com as respectivas legendas:

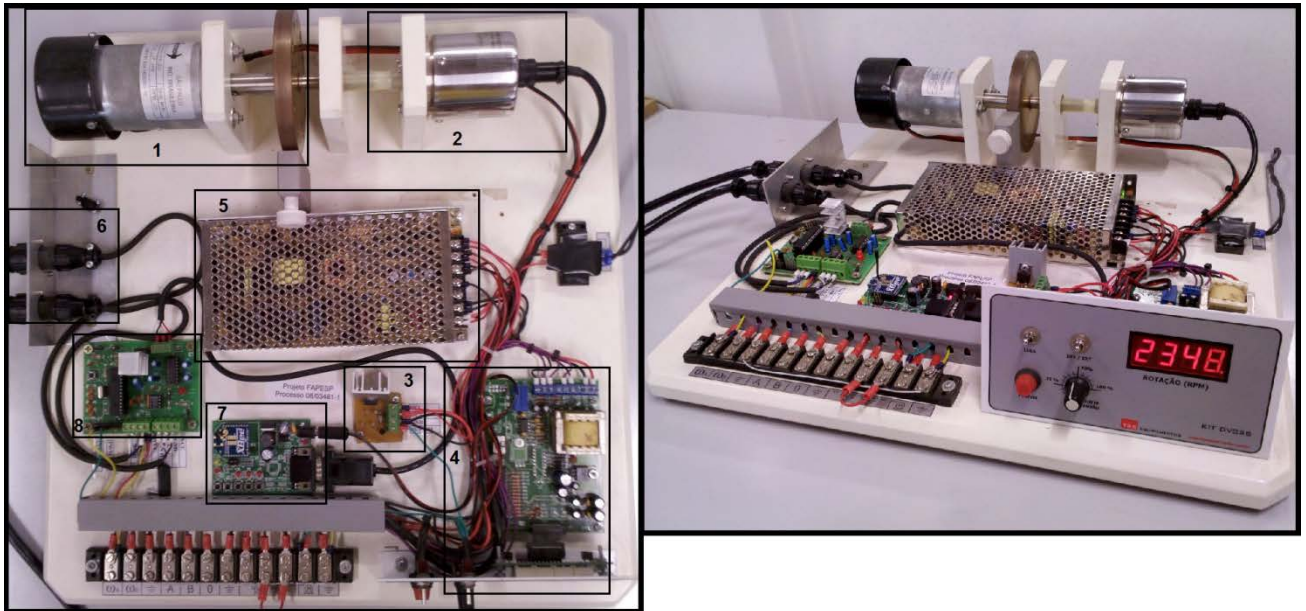


Figura 16 – Foto da estrutura da NCS de controle de velocidade

- 1 – Motor DC com um disco inercial no eixo acoplado;
- 2 – Encoder incremental Hohner [22] de 600 pulsos por revolução;
- 3 – Driver de acionamento (baseado no MOSFET IRF9640);
- 4 – Tacógrafo com um display que informa a velocidade em rpm
- 5 – Fonte AC/DC chaveada de 24V;
- 6 – Conectores específicos para uma rede CAN;
- 7 – Módulo ZigBee
- 8 – ECU baseada no PIC18F2580.

O sinal de input (sinal de controle PWM para acionamento do motor DC) é transmitido do controlador para o NCS via rede CAN e o sinal output (medição da velocidade de rotação do motor através do encoder) é transmitida pela NCS para o controlador via ZigBee. Para a integração da malha de controle à rede CAN e ZigBee foi feita utilizando uma ECU. Poderíamos utilizar duas ECUs (uma para ZigBee e outra para o CAN), que traria modularidade, muito representativo numa aplicação industrial. Mas para fins acadêmicos não julgamos necessário já que a velocidade de processamento da ECU não acarreta atraso significativo.

4.3.1.2 Modelagem do NCS

O motor utilizado nesta malha foi um motor DC da marca Motron [23], mod. M-910 de 24V. As equações (1) a (4), disponíveis na literatura [24], apresentam as relações básicas utilizadas na modelagem matemática para controle de velocidade de motores de corrente contínua.

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

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

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

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

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), ω_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.m²), 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).

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

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

4.3.2 Controlador

O controlador utilizando neste trabalho foi projetado por E.P. Godoy em [10]. Apenas foram adicionadas algumas novas funcionalidades que foram possíveis devido à modularidade do programa desenvolvido. As funcionalidades adicionadas estão relacionadas a integração do ZigBee e a intercalação entre dados reais e simulados.

4.3.2.1 Hardware do Controlador

Pesquisas recentes sobre modelagem de NCS têm sido realizadas utilizando de 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, controlador) são amostrados e então transmitidos numa rede de comunicação [13]. 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.

A estrutura de NCS utilizada neste trabalho foi simplificada na figura 17 onde fica visível que o controlador fica alocado fisicamente separado dos sensores e atuadores, onde o que os mantém conectados são apenas as redes CAN e ZigBee.

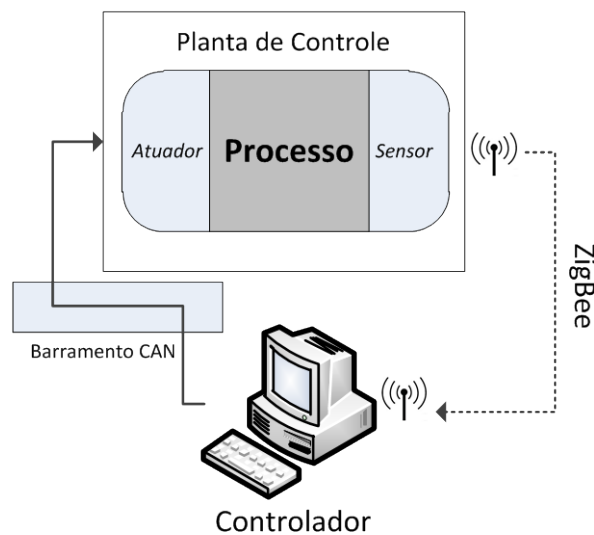


Figura 17 – Esquema de uma NCS

O controlador nesse trabalho é um computador comum equipado com uma interface PCI da National Instruments para comunicação com a rede CAN e uma interface ZigBee–USB XStick da Digi.

A leitura dos sensores por parte do controlador é feita pelas redes de comunicação de modo discreto a uma amostragem. Esta discretização dos sinais acaba motivando o desenvolvimento e a aplicação de controladores baseados no 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) pode induzir a presença de atrasos de comunicação nas redes do NCS. Esses atrasos de comunicação na rede 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. O período de amostragem também representa um importante parâmetro no desenvolvimento do NCS. De acordo com a teoria de controle digital, um período de amostragem deve ser definido o mais rápido (ou de valor pequeno) possível para obter um desempenho de controle melhor e uma dinâmica do sistema mais próxima a de um sistema contínuo no tempo. 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 [2].

4.3.2.2 Software do Controlador

O desenvolvimento de software para a plataforma de NCS com redes CAN requereu-se 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 e envia a informação através de mensagens na rede ZigBee para o controlador. Caso o tempo entre duas amostragens seja maior que um *timeout* estabelecido 115ms esta leitura do sensor é considerada perdida e o controlador calcula o sinal de controle baseado em dados simulados, caso contrário o controlador utiliza a leitura real para calcular o sinal de controle. Este sinal de controle é enviado através da 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 das redes CAN e ZigBee, e existe a possibilidade também de compartilhamento recursos de processamento (CPU) dos desktops por mais de um NCS. 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 18. 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.

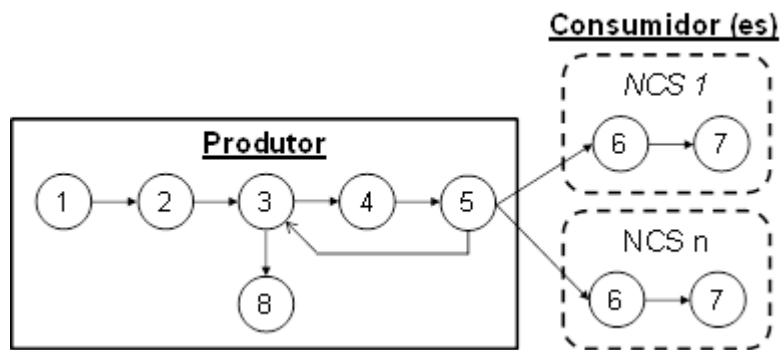


Figura 18 - Estrutura de Software da Plataforma de NCS com Rede CAN

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 se 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 (juntamente com o produtor/consumidor) 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 17, cada um com uma finalidade específica para o funcionamento do controlador [25].

1 – Inicialização: estado responsável pela configuração e inicialização da rede CAN, ZigBee 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 ZigBee nos campos e informações requeridos. Caso o tempo entre duas mensagens seja maior que o *timeout* estabelecido este estado é responsável por enviar dados simulados ao próximo estado.

5 – DeterminaçãoNCS: estado responsável por identificar as mensagens ZigBee 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;

As equações abaixo (7 a 13) descrevem o algoritmo do controlador PID discreto utilizado.

$$e(k) = r(k) - y(k) \quad (7)$$

$$e_p(k) = B.r(k) - y(k) \quad (8)$$

$$e_d(k) = C.r(k) - y(k) \quad (9)$$

$$u_p(k) = K.e_p(k) \quad (10)$$

$$u_i(k) = u_i(k-1) + \frac{K.h}{T_i}e(k) + \frac{K.h}{T_t}e_s(k-1) \quad (11)$$

$$u_d(k) = \frac{T_d}{T_d + Nh}u_d(k-1) + \frac{KT_dN}{T_d + Nh}[e_d(k) - e_d(k-1)] \quad (12)$$

$$u_{PID}(k) = u_p(k) + u_i(k) + u_d(k) \quad (13)$$

Neste trabalho, o controlador PID discreto definido utiliza um tempo de amostragem (h), um valor de N=10 para a constante de filtragem da ação derivativa, B = 1 e C = 0 para as constantes de ponderação de referência e o parâmetro Tt igual a $T_t = \sqrt{T_i.T_d}$ para controladores PID e igual a Ti para controladores PI.

4.4 Estratégia do Controle Baseado em Modelo

Como já foi discutido, sistemas de controle via rede ficam dependentes das condições da rede utilizada. Se tratando de protocolos sem fio essas condições são mais críticas, pois o a camada física é mais vulnerável. Essa vulnerabilidade implica em atraso e perda de dados. Para diminuir ou eliminar este problema podemos utilizar uma estratégia de controle baseado em modelo.

A estratégia desenvolvida nesse trabalho trata-se de um *timer* que dispara sempre que o uma resposta (real ou simulada) é enviada ao controlador. Caso esse *timer* ultrapasse um *timeout* estabelecido o sinal de controle enviado ao sistema é calculado a partir de uma resposta simulada. Obtida através de uma simulação baseada no modelo do sistema que roda em paralelo ao sistema real.

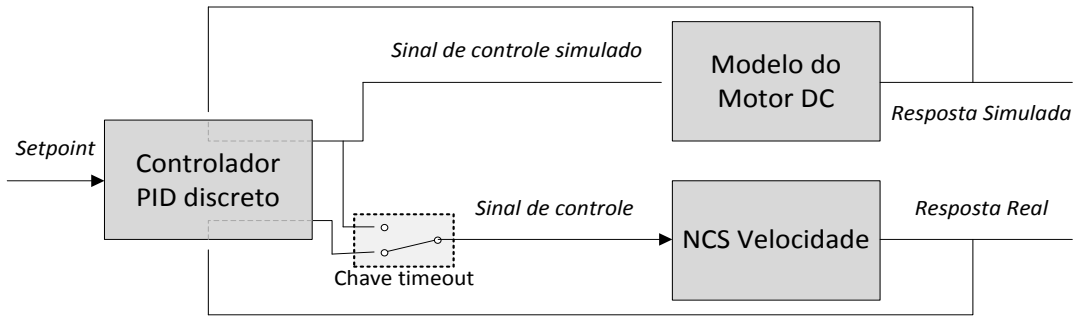


Figura 20 – Esquema simplificado da estratégia do controle baseado em modelo

De forma simplificada a figura 20 mostra a estratégia de controle utilizada. A “chave timeout” representa a escolha do sinal de controle simulado ou real dependendo se o tempo entre duas mensagens foi maior ou menor que o *timeout*. Desta forma o *timeout* é estabelecido levando em conta o *delay* e o *jitter* da rede em que o sensor esta conectado.

Esta estratégia é capaz de corrigir problemas de perda de pacotes ou de grandes atrasos mantendo a estabilidade e desempenho do sistema.

5. Resultados

Para validar a proposta foram feitas coletas de dados e gráficos que mostram as melhorias e a estabilidade que este projeto possibilitou. A estratégia utilizada foi a resposta do sistema a um degrau de 100 rad/s em diversas circunstâncias. Dessa forma será possível observar os efeitos da perda de dados e as correções de acordo com a estratégia proposta.

O microcontrolador foi programado para enviar a leitura do sensor para a interface ZigBee a cada 0,1 segundos (amostragem). O mesmo valor foi utilizado no controlador PID discreto. Os parâmetros do controlador foram definidos utilizando o método de Ziegler-Nichols (ZN) em conjunto com uma posterior sintonia manual [10]. Os melhores parâmetros encontrados foram $K_p=0,06$, $T_i=0,65$ $T_d=0$.

5.1 Validação do modelo

A fim de testar o sistema e validar o modelo criado para o NCS de velocidade do motor DC, foi feito um ensaio comparando a resposta real e simulada. Para esta simulação foi utilizado um *toolkit* de simulação e controle para o LabView que irá rodar em paralelo com o sistema real. Esta simulação é basicamente composta pela função transferência do motor DC (equação 6).

A leitura do sensor do NCS e a resposta simulada são inseridas no controlador PID discreto (figura 19) e os sinais de controle simulado e real são enviados à simulação e ao NCS respectivamente. Este processo pode ser exemplificado pela figura 20 desde que a “chave timeout” nunca seja acionada.

As respostas do sistema real (NCS velocidade) e da simulação a um degrau de 100 rad/s foram colocadas no gráfico da figura 21. A partir deste gráfico observamos que o modelo não está completamente exato, no entanto, como estratégias de modelagem não é a intenção deste trabalho, consideraremos que modelo esta validado.

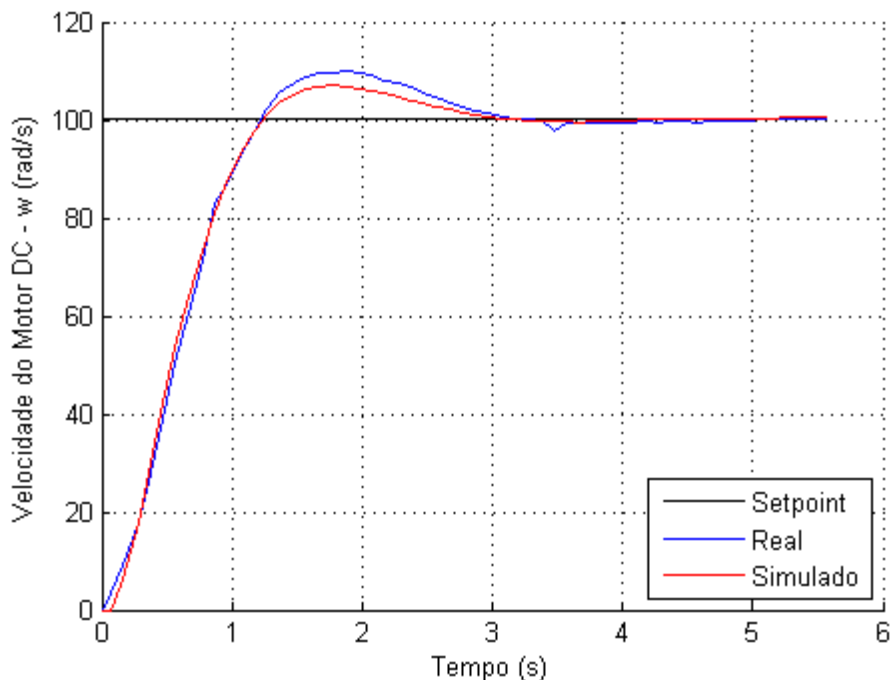


Figura 21 – Resposta ao degrau NCS Velocidade

5.2 Validação da estratégia de controle baseado em modelo

Apesar da confiabilidade do protocolo ZigBee [6], todo protocolo de comunicação, principalmente sem fio, esta sujeito a perda de dados e atrasos não previsíveis [27]. Esta perda de dados pode acarretar perdas de estabilidade e desempenho do sistema. A partir disso, o desenvolvimento de estratégias que corrijam estes problemas tem sua importância. Controle baseado em modelo é uma solução que vem sendo pesquisada [11] com bons resultados.

Para mostrar os efeitos negativos da perda de dados em um sistema, programamos o microcontrolador do NCS velocidade para deixar de enviar um número determinado de leituras do sensor num instante também determinado. Os pacotes deixam de ser enviados de forma sequencial, isto é, caso seja solicitado o não envio de 20 pacotes, a uma amostragem de 100ms, a ECU ficará por aproximadamente dois segundos seguidos (a partir de um instante determinado) sem enviar a leitura do sensor. Assim pudemos simular a perda de pacotes de uma forma controlada. Para obtermos o pior caso, a perda de dados será programada para ocorrer no transitório.

As respostas sem pacotes perdidos, e com 5, 10 e 20 pacotes perdidos (a partir de aproximadamente 0,5 segundos) foram colocadas no gráfico da figura 22. Não houve perda de estabilidade, no entanto uma significativa perda de desempenho.

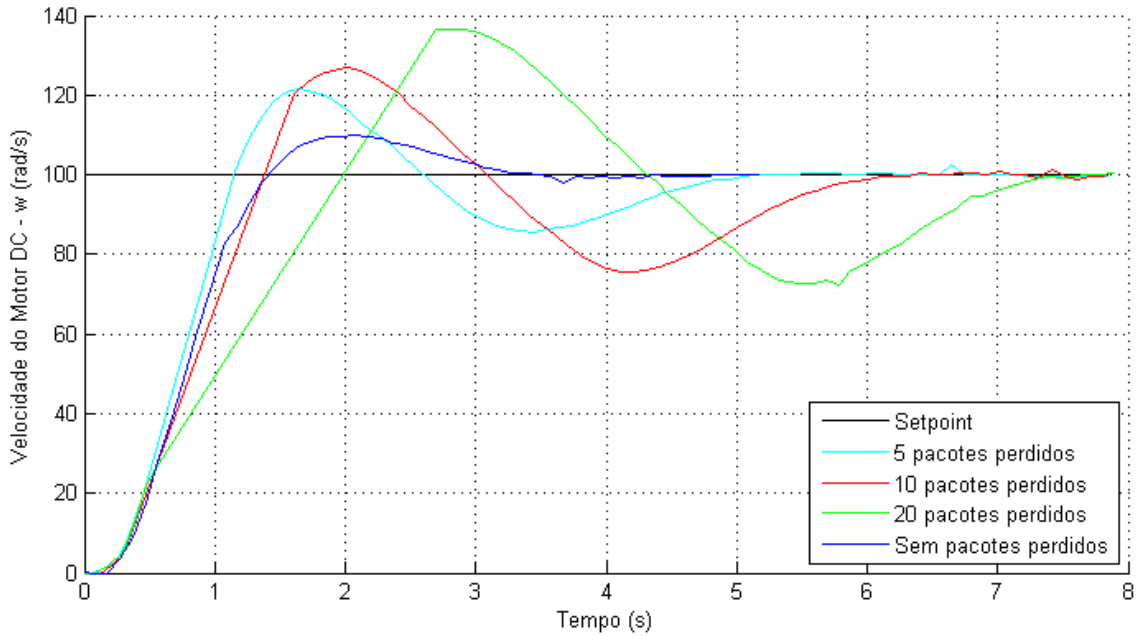


Figura 22 – Efeito da perda de pacotes no NCS Velocidade

Os mesmos ensaios foram repetidos utilizando a estratégia de controle baseado em modelo discutida no capítulo anterior. Nesta estratégia caso um pacote demore mais que o *timeout* estabelecido 0,115s, o controlador ira utilizar dados simulados para estimar o sinal de controle. Os resultados são mostrados nas figuras 23, 24, e 25 que comparam as respostas reais utilizando ou não a estratégia de controle baseado em modelo além dos dados simulados para que possamos observar a sua utilização durante os ensaios.

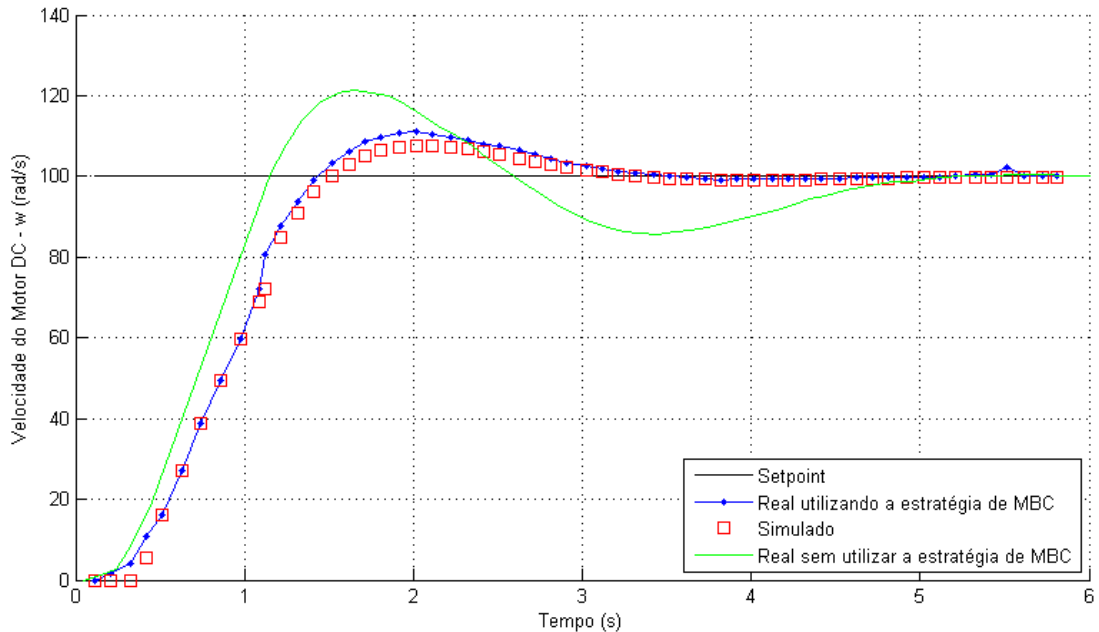


Figura 23 – Resposta ao degrau com cinco leituras perdidas

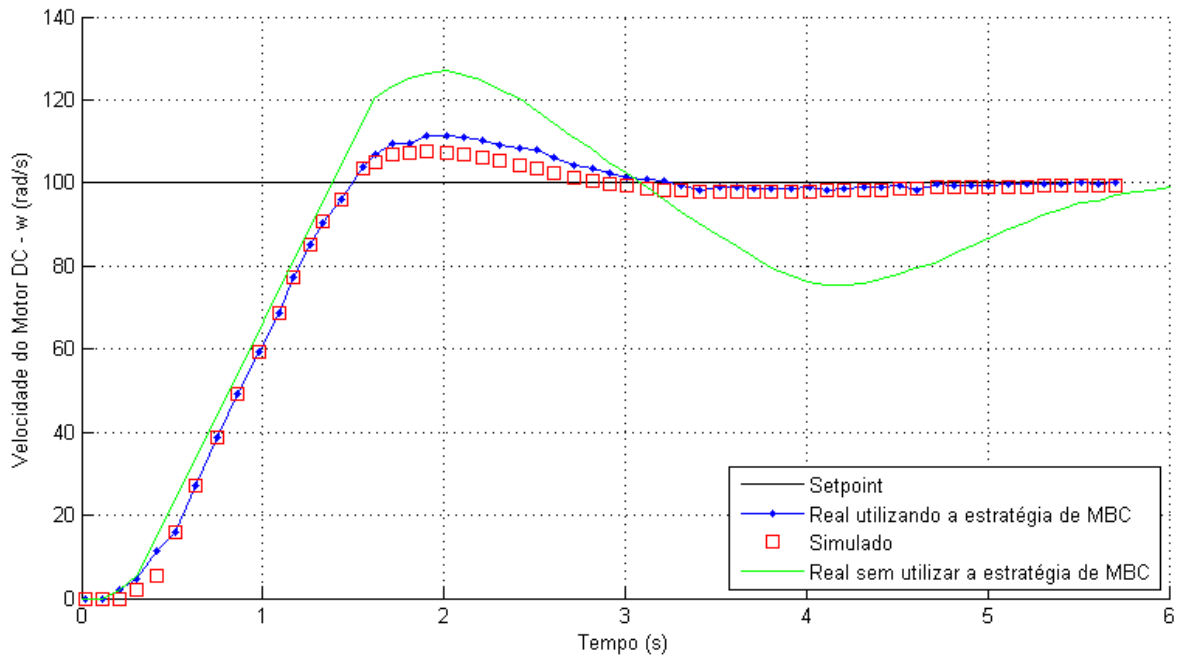


Figura 24 – Resposta ao degrau com dez leituras perdidas

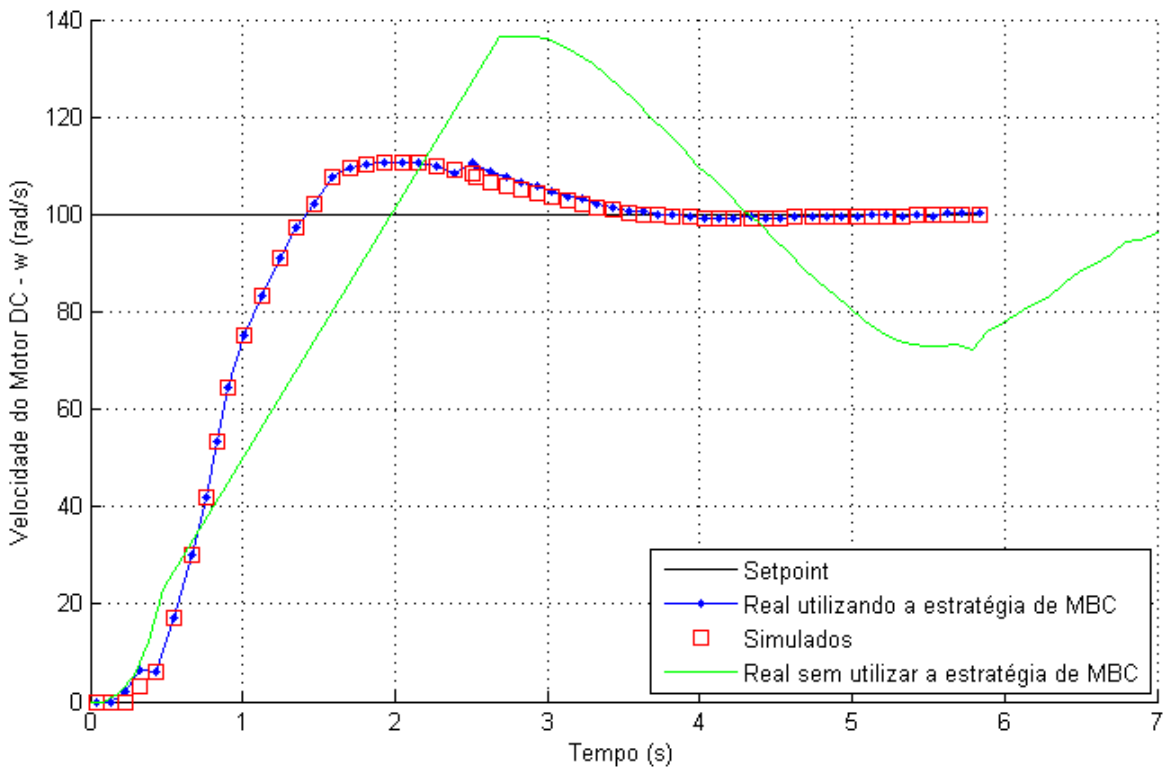


Figura 25 – Resposta ao degrau com vinte leituras perdidas

Comparando os gráficos dos sistemas que utilizaram a estratégia do MBC com os que apenas perderam os dados podemos observar um considerável ganho de

desempenho. A estratégia funcionou conforme esperado e foi validada como um eficiente método de controle de perda de pacotes.

6. Conclusão

O desenvolvimento de NCS é um problema multidisciplinar que envolve desde a modelagem do sistema físico, passando por sua validação, pela simulação do controle via rede e finalmente pela sua implantação real.

Inicialmente foram criadas as bibliotecas para microcontroladores PIC e LabView que facilitaram modificações nos programas e facilitarão aplicações futuras. Neste trabalho foi feita ainda uma análise comparativa entre os módulos Bluetooth e ZigBee citados para determinar qual teria melhores resultados em termos de métricas de desempenho para aplicações em NCS como atraso de transmissão e jitter. O ZigBee apresentou menor jitter e menor atraso, além de não perder nenhum pacote nos ensaios realizados, e portanto foi o protocolo escolhido.

Foi feita a validação do modelo (apesar de não exato) comparando a resposta real e simulada. Com este ensaio também podemos concluir a viabilidade da construção de sistemas de controle via rede híbrido com CAN e ZigBee dado que o controle foi realizado com êxito.

Foram feitos ensaios com perda de dados que mostram a redução do desempenho do NCS Velocidade. Posteriormente foi validada a estratégia do MBC com a repetição dos ensaios com perda de dados e a reposição destes dados perdidos por dados simulados.

Em se tratando de controle baseado em modelo, quanto melhor for a modelagem melhores serão os resultados. No entanto, como existem sistemas difíceis de serem modelados ou simulações que exigem muito processamento, a utilização de modelos muito exatos pode ficar inviável. A validação da estratégia do MBC para o modelo obtido neste trabalho mostra que a estratégia é válida também para modelos simples e não exatos. Dessa forma, como a perda de dados de sistemas de controle via rede não é predominante, um modelo simples é suficiente para manter o funcionamento adequado do sistema.

7. Trabalhos Futuros

Entre as perspectivas desse trabalho, espera-se que os resultados obtidos possam ser utilizados como em futuros trabalhos que utilizem sistemas de controle via rede híbrido bem como a estratégia de controle de pacotes perdidos.

Em NCS a taxa de amostragem é um fator que influencia muito na qualidade do controle [7]. Uma outra melhora que pode ser implementada em um sistema de controle via rede utilizando o controle baseado em modelo é a técnica chamada *multi-rate*. Nesta técnica são utilizados dados simulados para aumentar a amostragem. Dados reais são intercalados com dados simulados de forma que o período de amostragem seja tão pequeno quanto o necessário.

8. Referências bibliográficas

- [1] Yang, T. C. ,“Networked control system: a brief survey”, IEEE Proceedings of Control Theory and Applications, Vol. 153, No 4, pp. 403 – 412, 2006.
- [2] Fischione, C.; Park, P.; Di Marco P.; Johansson, K.H., “Design Principles of Wireless Sensor Networks Protocols for Control Applications”, S.K. Mazumder (ed.), Wireless Networking Based Control, Ch. 9, Springer, pp. 203-238, 2011.
- [3] Pereira, R.R.D. “Protocolo ISO 11783: Procedimentos para comunicação serial de dados com o Controlador de Tarefas”, Dissertação (Mestrado em Engenharia Mecânica).Escola de Engenharia de São Carlos – Universidade de São Paulo.São Carlos, 168p, 2009.
- [4] Bosch, “CAN Specification Version 2.0”, Robert Bosch GmbH, Disponível em <<http://www.semiconductors.bosch.de/media/pdf/canliteratur/can2spec.pdf>>. Acesso em Março de 2011.
- [5] ZigBee Alliance, ZigBee-2007 Specification, , Disponível em <<http://www.zigbee.org/>>. Acesso em Março de 2011.
- [6] Aakvaag, N.; Mathiesen, M.; Thonet, G. “Timing and power issues in wireless sensor networks - an industrial test case”, IEEE 34th International Conference Workshops on Parallel Processing, Oslo, Noruega, pp. 419-426.
- [7] YANG, T. C. Networked control system: a brief survey. IEEE Proceedings of Control Theory and Applications, Vol. 153, Nº 4, pp. 403–412, 2006.
- [8] Moyne, J. R.; Tilbury, D. M. The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data, IEEE Proceedings of the Technology of Networked Control Systems, Vol. 95, Nº 1, pp. 29-47, 2007.

- [9] Othman, H. F.; Aji, Y. R.; Fakhreddin, F. T.; Al-Ali, A. R. Controller Area Networks: Evolution and Applications, In: International Conference on Information and Communication Technologies (ICTTA), 2 Proceedings. Vol. 2, pp. 3088- 3093, 24-28, 2006.
- [10] Godoy, E. P. Desenvolvimento de Sistemas de Controle via Rede (NCS) para Aplicações em Redes com Protocolo CAN. 249f. Tese de Doutorado–Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2011
- [11] Estrada, T.; Antsaklis, P. A. Performance of Model-Based Networked Control Systems with discrete-time plants. In: Mediterranean Conference on Control and Automation, Greece. Proceedings. pp. 628-633, 24-26, 2009.
- [12] Sousa, R. V. 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.
- [13] Paavola, M.; Leiviska, K.. “Wireless Sensor Networks in Industrial Automation”, J.S. Blanes (ed.), Factory Automation, Ch 10, InTech, pp. 201-220, 2010.
- [14] Bluetooth, “Bluetooth Specification v2.0”, Disponível em <<http://www.bluetooth.com/>>. Acesso em Março de 2011.
- [15] Baronti, P.; Pillai, P.; Chook, V.W.C.; Chessa, V.; Gotta A. and Hu Y.F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. Computer Communications, Vol.30, Nº 7, pp. 1655-1695.
- [16] GÖTEBORG ENERGI, " Göteborg – The first ZigBee City?" , Disponível em <[http:// www.goteborgenergi.se/](http://www.goteborgenergi.se/)>. Acesso em agosto de 2011.
- [17] Microchip INC. Microchip web site. Disponível em: <http://www.microchip.com> Acesso em 20 julho de 2011
- [18] Digi International, Inc., “XBee®/XBee-PRO® ZB RF Modules”, Disponível em <<http://www.digi.com/>>. Acesso em julho de 2011.

- [19] Lian, F.L.; Moyne, J.R. and Tilbury, D.M. Network Design Consideration for Distributed Control Systems, IEEE Transactions on Control Systems Technology, Vol. 10 Nº 2, pp. 297-307, 2002.
- [20] Anand, D.M.; Tilbury, D.M.. Wireless networksfor factory automation: Performance evaluation via analysis and experimentation. International Journal of Systems, Control and Communication, 2009
- [21] T&S Equipamentos eletrônicos, T&S Equipamentos eletrônicos web site. Disponível em: <http://www.tesequipamentos.com.br/>, Acesso em julho de 2011
- [22] Hohner Eletrônica Ltda, Hohner web site. Disponível em: <http://www.hohner.com.br/site/>, Acesso em julho de 2011
- [23] Motron Indústria de Motores Redutores Ltda, Motron web site. Disponível em: <http://www.motron.com.br/>, Acesso em julho de 2011
- [24] Fitzgerald, A. E.; Junior, C. K.; Umans, S. D.. "Máquinas elétricas com introdução a eletrônica de potência", 6ª edição, Editora Artmed Bookman.
- [25] Endsley, E.W.; Almeida, E.E.; Tilbury, D.M.. "Modular finite state machines: Development and application to reconfigurable manufacturing cell controller generation". Control Engineering Practice, Vol.14, pp.1127–1142, 2006.
- [26] 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 Baseados no Protocolo Can – Controller Area Network", Anais do Congresso Temático de Dinâmica, Controle e Aplicações DICON 2010, Serra Negra, 2010.
- [27] Wettergren A.. "ZigBee in Industry", Bachelor Thesis Performed in Computer Engineering, Linköping Institute of Technology, 2007.