

IMPLEMENTAÇÃO DE UMA REDE CAN APLICADA A ROBÔS MÓVEIS

José Victor Cavalcante AZEVEDO (1); Ítalo Jader Loiola BATISTA (2); Antônio Themoteo VARELA (3)

(1) Instituto Federal de Educação, Ciência e Tecnologia do Ceará, Av. Treze de Maio, 2081 – Benfica, CEP: 60040-531 Fortaleza - CE, e-mail: victor2_0@yahoo.com.br

(2) Universidade Federal do Ceará, Av. Mister Hull, S/N – Campus do Pici, CEP: 60455-760 – Fortaleza – CE, e-mail: italoloiola@gmail.com

(3) Instituto Federal de Educação, Ciência e Tecnologia do Ceará, Av. Treze de Maio, 2081 – Benfica, CEP: 60040-531 Fortaleza - CE, e-mail: themoteo@gmail.com

RESUMO

O protocolo CAN (*Controller Area Network*) tem sido vastamente utilizado na implementação de diversas aplicações de controle distribuído em tempo real. Neste trabalho apresenta-se uma implementação de uma rede CAN aplicado a um sistema distribuído de forma descentralizada para navegação de um robô móvel experimental. Os nós CAN implementados possuem arquiteturas diferentes em termos de unidade de controle e será mostrado os detalhes necessários para o desenvolvimento do hardware e do software requeridos para comunicação de dados em tempo real.

Palavras-chave: rede CAN, robô móvel e arquitetura distribuída.

1 INTRODUÇÃO

Os robôs móveis autônomos geralmente são controlados a partir de um sistema de controle que é composto por unidades de sensores, atuadores e central de processamento. Os sensores e atuadores podem estar ligados a um único sistema de processamento formando uma arquitetura centralizada ou ligados a diferentes unidades de processamento interconectadas entre si formando uma arquitetura distribuída.

A arquitetura distribuída é baseada em sistemas sensoriais e controladores distribuídos simples, interligados em uma rede de sensores e atuadores, na qual nenhum sistema individual está em nível hierárquico superior a qualquer outro. Desta forma, o sistema se torna robusto, confiável, flexível, extensível e mais tolerante a falhas do que na abordagem centralizada. A falha de um subsistema não implica necessariamente na falha do sistema global (COSTE; SIMMONS, 2000).

Os sistemas de processamentos de sistemas robóticos moderno geralmente são do tipo distribuído, o que exige uma memória compartilhada ou uma rede de comunicação de dados entre as unidades de processamento que garanta as restrições temporais para transmissão das mensagens. Uma rede de comunicação de dados que possui essa capacidade de comunicação em tempo real é a rede CAN.

Recentemente, têm sido introduzidos barramentos de controle baseado em rede CAN, em muitos sistemas com diferentes aplicações (SANTOS; STEMMER; VASQUES, 2002), (LOPES, 2008) e (ZEPLIN, 2004). Este trabalho propõe a implementação de uma rede CAN em um robô móvel autônomo.

2 A REDE CAN

A rede CAN foi desenvolvida pela empresa Robert Bosch em meados da década de 80 com objetivo de promover a interconexão entre dispositivos de controle em automóveis, tornando-se padrão ISO 11898 em 1993. Devido sua robustez a rede CAN também está sendo aplicada para ambientes industriais de controle de processos distribuídos, como por exemplo, a interface de comunicação proprietária DeviceNet da Rockwell (ROCKWELL, 1997).

A rede CAN é um barramento serial multi-mestre que usa transmissores para enviar para todos os “nós” da rede, possuindo um sistema sofisticado e robusto de detecção de erro. Os elementos para formar a rede CAN são, conforme a Figura 1:

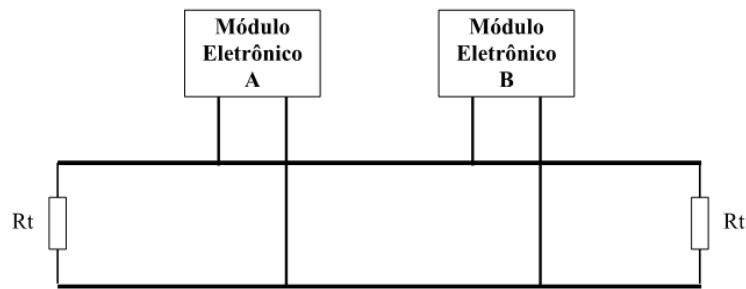


Figura 1 – Estrutura de um barramento CAN

Os elementos para formar a rede CAN são:

- **Módulo Eletrônico:** subsistema responsável pela aquisição dos dados de sensores e/ou comandar atuadores do sistema. Nesse módulo encontra-se o processador responsável pelos periféricos ligados a ele, o controlador CAN que é o responsável por implementar a norma para condicionamento do dado e um transceptor que adequa o sinal digital aos níveis elétricos da rede. Esse conjunto é denominado de “nó” da rede CAN;
- **Cabo:** é o meio físico para a transmissão dos dados, geralmente é utilizado cabo par trançado já que a norma não especifica o meio físico;
- **Resistores Terminais (Rt):** resistores colocados no final da rede para casamento de impedâncias, não permitindo reflexão de dados na rede e deterioração da comunicação.

2.1 O protocolo CAN e o padrão ISO/OSI

A arquitetura OSI (*Open Systems Interconnection*) é um modelo, criado pela ISO (*International Organization for Standardization*), que divide as redes de computadores em sete camadas: camada de aplicação, camada de apresentação, camada de sessão, camada de transporte, camada de rede, camada de enlace e camada física. Cada protocolo de comunicação implementa uma funcionalidade a determinada camada. Para o protocolo de comunicação CAN é definido apenas as suas camadas inferiores: enlace e física.

2.1.1 Camada de enlace no protocolo CAN

Na camada enlace é onde ocorre o encapsulamento/descapsulamento da mensagem, detecção e sinalização de erros de mensagens e onde as mensagens são filtradas (ZEPLIN, 2004).

O máximo de dados transmitido por uma mensagem é 64 bits, sem qualquer endereço explícito, sendo a estação receptora responsável por aceitar ou não a mensagem através do campo da identificação da mensagem que pode ser 11 bits (CAN 2.0A, padrão) ou 29 bits (CAN 2.0B, estendido).

Um conceito importante para rede em tempo real presente no protocolo CAN é CSMA/CD com NDA (Acesso Múltiplo com Detecção de Portadora com Arbitragem Não Destrutiva por Operação Lógica Bit-a-Bit). Por ser uma rede do tipo multi-mestre dois ou mais “nós” podem iniciar uma transmissão ao mesmo tempo, por isso após enviar um bit o “nó” verifica se nenhum outro “nó” da rede o sobrescreveu, caso isso ocorra o “nó” que estiver enviando uma mensagem com o campo de identificação de menor prioridade interrompe sua transmissão e passa a receber do barramento, enquanto que o “nó” que estiver enviando uma mensagem com maior prioridade (o campo identificador é que determina essa prioridade) continua a enviar a mensagem sem ter que recomençar a transmissão.

Os tipos de mensagens definidas para o protocolo CAN são: mensagem de dados, remota, erro e sobrecarga.

As mensagens de dados e as remotas podem ser do tipo padrão ou estendida (ver Figura 2). A diferença entre a mensagem de dados e a remota é que na mensagem remota não consta o campo de dados, já que o objetivo dessa mensagem é apenas requisitar de um “nó” da rede determinada informação.

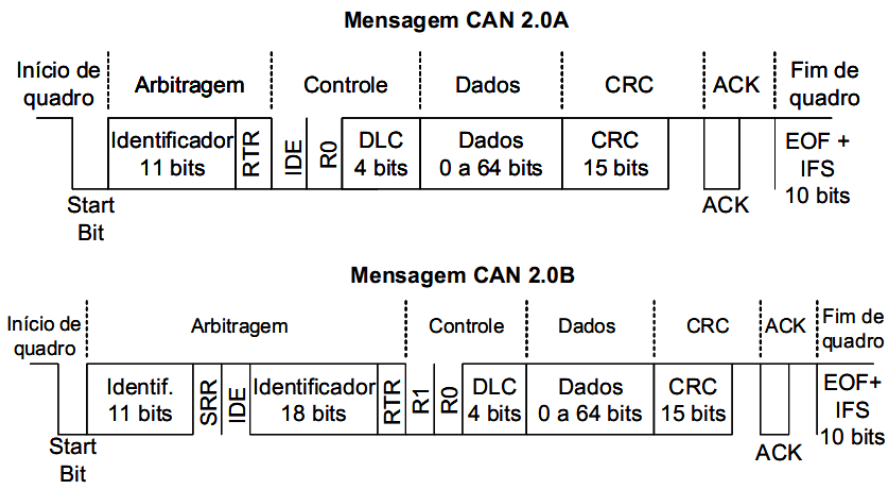


Figura 2 – Estrutura das mensagens CAN 2.0A e CAN 2.0B

2.1.2 Camada física

A camada física é responsável pela transferência dos bits entre os “nós” da rede, definir níveis elétricos, o esquema de ligação e impedância do cabo.

Devido à arbitragem, já descrito anteriormente, a velocidade máxima possível para o barramento CAN é limitada pelo comprimento do barramento, conforme pode ser visto na Figura 3, pois o esquema de arbitragem necessita que a onda se propague até o nó mais remoto e retorne antes de ser amostrada.

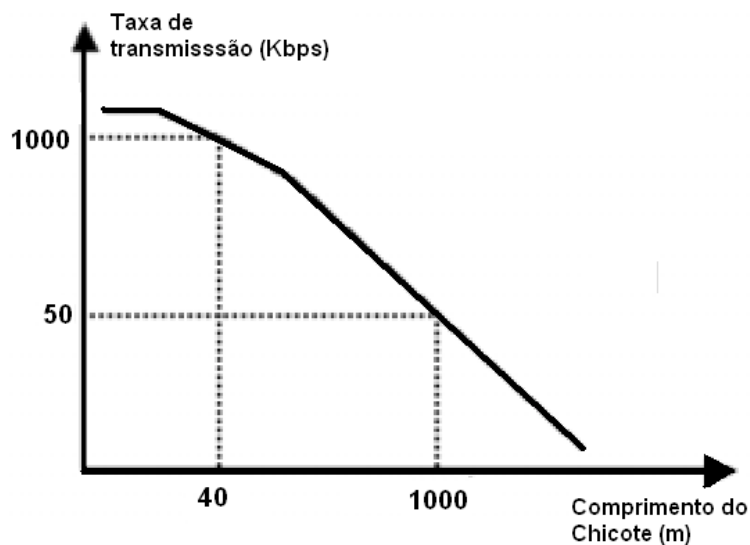


Figura 3 – Relação entre a velocidade de transmissão e comprimento do barramento

A transmissão dos dados na rede CAN se dá através de um par diferencial CAN_H (CAN *High*) e CAN_L (CAN *Low*). Quando o meio de transmissão for através de fios, a atenuação dos efeitos de interferência eletro-magnética no par diferencial pode ser amenizada usando par trançado ou fios blindados. No protocolo CAN os bits são representados por bits dominantes (nível lógico “0”) e bits recessivos (nível lógico “1”). A Figura 4 mostra os níveis de tensão para o par diferencial para os bits recessivos e dominantes.

Quando não há transmissão de mensagem na rede o barramento recebe bits recessivos, assim os dois fios ficam sob o mesmo nível de tensão, portanto só existe dissipação de potência na rede apenas quando há a transmissão de bits dominantes.

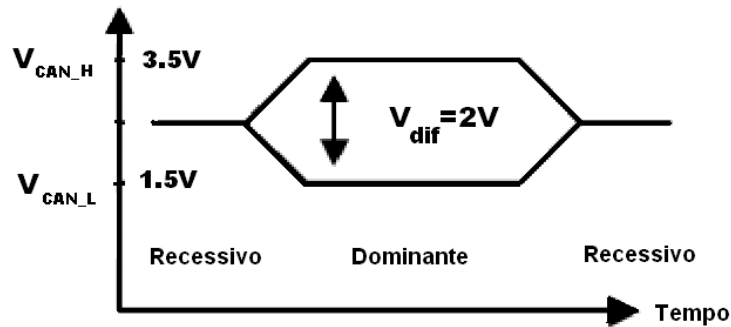


Figura 4 – Níveis de tensão no barramento CAN

3 ARQUITETURA DO SISTEMA

3.1 Descrição da plataforma

Assume-se um robô móvel tal como descrito no diagrama de blocos da Figura 5. Sensores de obstáculos, colisão e captura provêm as informações utilizadas pelo sistema de navegação. Os sensores de obstáculos estão dispostos à frente do robô. O sensor de colisão detecta as colisões contra obstáculos. Os sensores captam estímulos provenientes do ambiente e os transmitem para o sistema de navegação. A cada instante o sistema de navegação recebe os sinais captados pelos sensores e determina a ângulo de ajuste para correção da direção de movimento do robô móvel. Após o ajuste de direção de movimento, o robô avança um passo na trajetória pré-estabelecida.

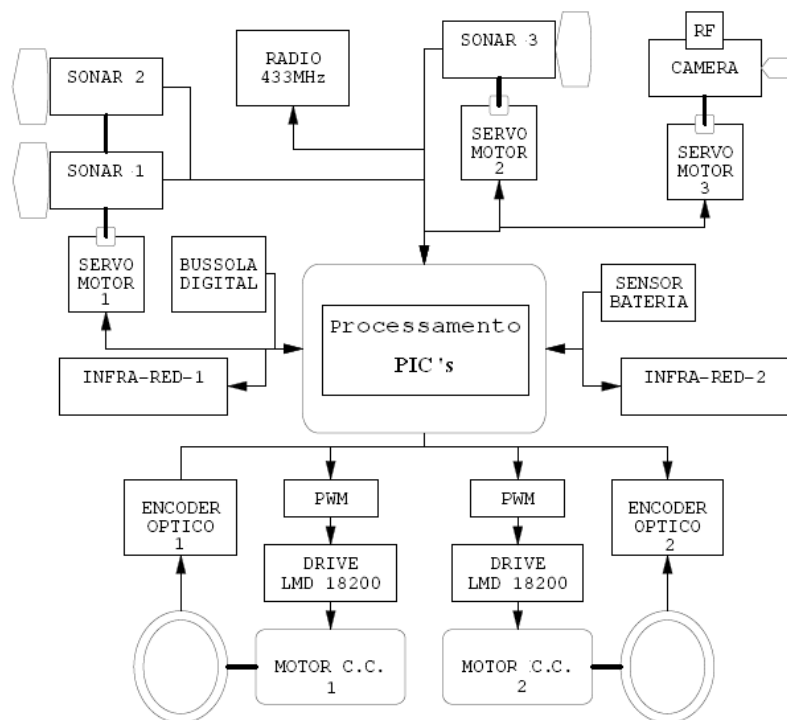


Figura 5 – Diagrama de blocos do robô móvel.

Na Figura 6 é mostrado o protótipo de um robô móvel experimental que está sendo utilizado como plataforma de testes para a rede CAN implementada. O robô é composto por periféricos, dentre eles sonares baseados em ultra-som, sensores de infra-vermelho, ambos utilizados neste trabalho para detecção de obstáculos e medição de distâncias para alimentar o sistema de navegação do robô. É incorporado ao robô um sistema de comunicação sem fio, conferindo assim uma maior flexibilidade no tratamento das informações obtidas pelos sensores, com um propósito de monitoramento. Os movimentos de rotação e translação são efetuados por dois motores de corrente contínua acoplados diretamente às rodas de tração do robô, e outras duas rodas apenas para apoio. Esta configuração é bastante utilizada por robôs móveis em

diversas aplicações na literatura especializada (BORENSTEIN, 1996). O circuito de processamento e controle é baseado em microcontroladores da família PIC de fabricação da Microchip interligados por meio de uma rede CAN, conforme será descrito. Quanto à sua geometria, ele possui forma cilíndrica com 0,80m de altura e 0,30m de raio.

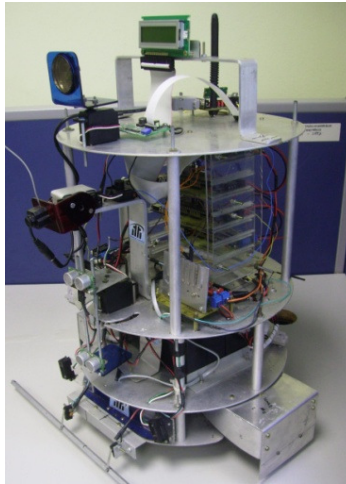


Figura 6 – Robô móvel experimental.

3.2 Modelo distribuído

O robô móvel apresenta uma arquitetura distribuída formada por seis unidades de processamento, no qual cada uma forma um “nó” da rede CAN. Essa distribuição foi feita analisando as funcionalidades de processamento requeridas (timer, interrupção, pinos de entrada e saída e etc.) para o controle dos periféricos. A distribuição dos periféricos em cada “nó” é mostrada na Figura 7.

O Nó 1 é responsável pelo sistema de navegação do robô, pelo sistema de comunicação sem fio e por gerar referência de velocidade para os motores dos “nós” 2 e 3, a partir dos dados obtidos dos outros “nós”. Os “nós” 2 e 3 têm implementado um controlador proporcional-integral-derivativo (PID) para controle de velocidade dos motores das rodas de tração.

Nos “nós” 4, 5, 6 e 7 são distribuídos os servos motores de posicionamento angular os sensores para navegação, no qual cada um desses “nós” faz o tratamentos dos sinais dos seus respectivos sensores e enviam esses dados já processados para o Nó 1.

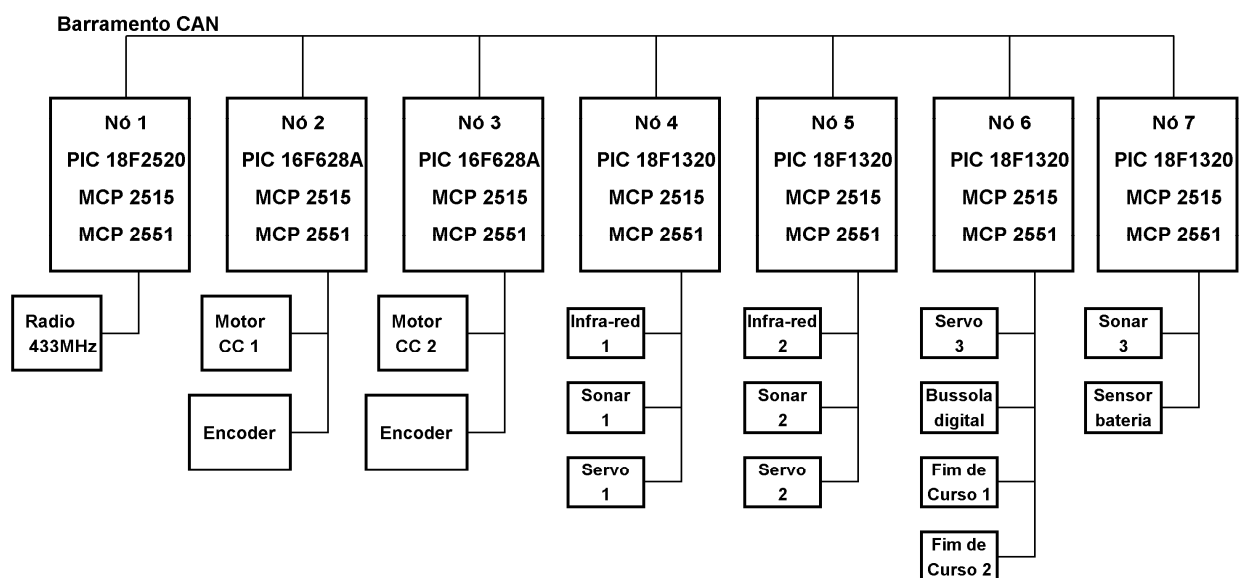


Figura 7 – Esquema da arquitetura do robô móvel experimental

4 IMPLEMENTAÇÃO DA REDE CAN

4.1 Desenvolvimento do hardware

Cada “nó” é baseado em um microcontrolador da família PIC, conforme Figura 7, sendo sua escolha de acordo com a necessidade de cada “nó”. Esses microcontroladores não apresentam controlador CAN interno, portanto torna-se necessário o uso de um controlador CAN externo e um transreceptor CAN para adequar os níveis de tensão do barramento CAN. O controlador CAN utilizado é o MCP2515 e o transreceptor é o MCP2551, ambos de fabricação da Microchip, assim como os microcontroladores. A estrutura simplificada de cada “nó” é apresentada na Figura 8.

O Nó 1 utiliza o microcontrolador PIC18F2520 como unidade de processamento enquanto que os “nós” 2 e 3 utilizam PIC16F628A e os “nós” 4, 5, 6 e 7 utilizam o PIC18F1320.

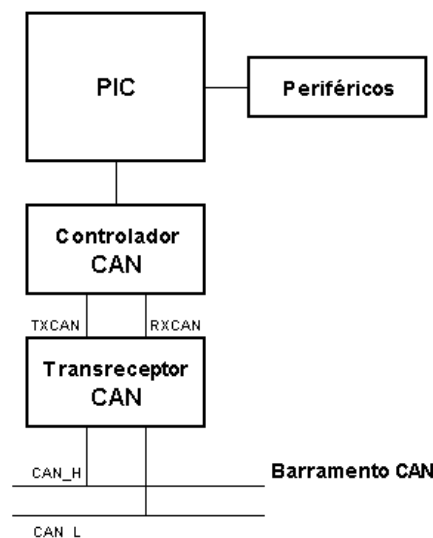


Figura 8 – Estrutura de um Nó

4.2 Desenvolvimento do firmware

Entendendo o funcionamento e principalmente a arquitetura do robô móvel é necessário para a implementação da rede CAN determinar as mensagens que deverão ser trocadas entre os “nós” da rede.

Foram seguidos os seguintes passos para determinar as mensagens: determinar os dados que cada “nó” deve enviar e receber; determinar os dados que serão enviados na mesma mensagem, sabendo que é possível enviar até 8 bytes em uma mensagem e a prioridade de cada mensagem para poder determinar seu identificador.

Devido o número de mensagens não ser tão extenso foi escolhido o padrão CAN 2.0A, cujo campo identificador é de 11 bits.

Os bits do campo identificador da mensagem foram determinados pela seguinte regra:

- Bits 10 a 8: tipo de mensagem, 011 para mensagem de dados e 001 para mensagens remotas;
- Bits 7 a 4: número do “nó” que envia a mensagem;
- Bits 3 a 0: número da mensagem do respectivo “nó”.

Exemplo: para um identificador 01101000010, temos que é uma mensagem de dados enviada pelo Nó 4 e corresponde a mensagem 2 desse “nó”.

A Tabela 1 mostra as mensagens de dados, e seus respectivos identificadores, “nó” de origem e “nó” de destino.

Como o Nó 1 é responsável pelo sistema de navegação torna-se necessário que ele possa requisitar a qualquer instante dos outros “nós” da rede os dados dos sensores de navegação, deslocamento e velocidade atual de cada roda. Portanto foram definidas as seguintes mensagens remotas, conforme Tabela 2.

Tabela 1 – Mensagens de dados

Mensagem	Identificador (binário)	“Nó” de origem	“Nó” de destino
Referência de velocidade do Motor CC 1	01100010001	1	2
Referência de velocidade do Motor CC 2	01100010010	1	3
Referência de posição do Servo 1	01100010011	1	4
Referência de posição do Servo 2	01100010100	1	5
Referência de posição do Servo 3	01100010101	1	6
Velocidade atual do Motor CC 1	01100100001	2	1
Velocidade atual do Motor CC 2	01100110001	3	1
Distância Encoder 1	01100100010	2	1
Distância Encoder 2	01100110010	3	1
Distâncias do infra-red 1 e sonar 1	01101000001	4	1
Distâncias do infra-red 2 e sonar 2	01101010001	5	1
Orientação bússola e fim de curso	01101100001	6	1
Distância sonar 3 e nível tensão da bateria	01101110001	7	1

Tabela 2 – Mensagens de remotas

Mensagem	Identificador (binário)	“Nó” de origem	“Nó” de destino
Velocidade atual do Motor CC 1	00100010001	1	2
Velocidade atual do Motor CC 2	00100010010	1	3
Distância Encoder 1	00100010011	1	2
Distância Encoder 2	00100010100	1	3
Distâncias do infra-red 1 e sonar 1	00100010101	1	4
Distâncias do infra-red 2 e sonar 2	00100010110	1	5
Orientação bússola e fim de curso	00100010111	1	6
Distância sonar 3 e nível tensão da bateria	00100011000	1	7

5 CONCLUSÃO

O sistema apresentado se baseia em controladores distribuídos interligados em uma rede CAN possibilitando que se torne mais robusto, confiável, com menor carga de processamento e maior tolerância a falhas do que se fosse realizada uma abordagem centralizada. A falha de um subsistema não acarreta na falha do sistema global.

A arquitetura desenvolvida que foi aplicada ao robô móvel experimental que é capaz de lidar com um grande número de sensores e atuadores distribuídos permitindo que estes sejam controlados de forma independente através de microcontroladores dedicados comunicando-se através de uma rede CAN.

A rede CAN implementada ofereceu boas condições para que se avaliasse o seu desempenho. Foi possível observar as vantagens que se obtêm utilizando uma arquitetura distribuída tais como: robustez, confiabilidade, velocidade e possibilidade de expansão para novos pontos de controle. Como pôde ser observado, a implementação de um barramento CAN envolve conceitos de hardware, que é fortemente dependente da estratégia de utilização do microcontrolador e de software.

Como trabalho futuro será realizada uma análise detalhada dos requisitos temporais da arquitetura proposta, considerando a latência para a transmissão de mensagens, situações como a presença de variação estatística do retardo na entrega de dados no conjunto de mensagens (*Jitter*), a alta carga de mensagens sob uma rede e erros de transmissão, utilizando-se da teoria de escalonamento em tempo real.

REFERÊNCIAS

BORENSTEIN, H. R. **Navigation Mobile Robots: Systems and Techniques**, AK Peters. et al., 1996.

BOSH, Robert GmbH. **CAN Specification 2.0**, 1991

COSTE-MANIÉRE, E. e SIMMONS, R. **Architecture, the backbone of robotic systems**. IEEE International Conference on Robotics and Automation , 2000.

LOPES, W.C. **Análise de desempenho do protocolo CAN para aplicação na área agrícola utilizando redes de Petri coloridas**. 2008. Dissertação (Mestrado em Engenharia Mecânica) - Escola de Engenharia de São Carlos (EESC), 2008.

MICROCHIP. **MCP2510 Data sheet**, Microchip technology, USA, Inc, 2002.

MICROCHIP. **MCP2551 Data Sheet**, Microchip technology, USA, Inc, 2002.

ROCKWELL. **DeviceNet product overview**. Publication DN-2.5, Rockwell, 1997.

SANTOS, M.M.D.; STEMMER, M.R.; VASQUES, F. **Uma rede CAN aplicada ao controle de uma aeronave não tripulada - O Helicóptero HELIX**. In *Proc. XIV Congresso Brasileiro de Automática*, Natal - Rio Grande do Norte. 2002.

ZEPLIN, Stefano Romeu. **Desenvolvimento de uma rede CAN utilizando DSPs**. 2004. Dissertação (Mestrado em Mestrado em Automação Industrial) - Universidade do Estado de Santa Catarina.