
UM PROTOCOLO CONFIÁVEL E FLEXÍVEL DE COMUNICAÇÃO PARA FUTEBOL DE ROBÔS

Murilo Fernandes Martins

murilo_fm@yahoo.com.br

Flavio Tonidandel

flaviot@fei.edu.br

Reinaldo A. C. Bianchi

rbianchi@fei.edu.br

Centro Universitário da FEI
Avenida Humberto de Alencar Castelo Branco, 3972
CEP 09850-901, São Bernardo, São Paulo, Brasil

ABSTRACT

A good communication among the computer and the mobile robots is essential for real-time controlling. There are many communication protocols in the literature and they have several and specific features. However, features like reliability and flexibility are not fully present in these protocols although they are very important for an effective and expansible control. In this paper, several protocols features are classified and analyzed, and a new protocol is proposed in order to allow reliable communication and expansion flexibility.

KEYWORDS: Mobile Robots, Robot Soccer, Communication protocol.

RESUMO

Uma boa comunicação entre computador e robôs em uma partida de futebol é crucial para um controle de tempo real efetivo e eficiente dos jogadores. Existem diversos protocolos de comunicação com diversas características. Entretanto, algumas características como flexibilidade e confiabilidade não são facilmente encontradas na literatura. Neste artigo, as características dos protocolos encontrados na literatura bem como sua utilidade em futebol de robôs são classificadas e analisadas. Um novo protocolo é proposto de modo a permitir que certas características, julgadas importantes, como flexibilidade e confiabilidade de comunicação estejam presentes.

PALAVRAS-CHAVE: Robôs Móveis, Futebol de Robôs, Protocolo de Comunicação.

1 INTRODUÇÃO

O desenvolvimento de robôs móveis autônomos para serem utilizados em uma partida de futebol de robôs, da categoria Mirobot [1], envolve diversas áreas de pesquisa. Nessa categoria, assim como em outras categorias, o sistema é composto por uma câmera posicionada acima do

campo e um computador, que recebe a imagem capturada, caracterizando o sistema de visão global. O computador é responsável por dar autonomia ao sistema, ou seja, por analisar a imagem, determinar as ações e comunicar-se com os robôs a serem controlados. Essa comunicação de dados entre computador e robôs é feita via radiofrequência e utiliza uma política de comunicação que não caracteriza um protocolo, tornando tal implementação específica apenas a determinada aplicação, implicando no desenvolvimento de uma nova política de comunicação a cada alteração de hardware e software de tal sistema.

A sintaxe (o formato que os dados são enviados), a semântica (o controle da informação e tratamento de erros) e a velocidade, isto é, a taxa de transmissão de dados, definem a maneira pela qual o computador e os robôs irão se comunicar. Essa política de comunicação caracteriza um protocolo. Para aplicações de controle de robôs, seja numa partida de futebol ou mesmo um robô móvel em outra aplicação, o protocolo deve ser otimizado para que sua estrutura seja suficiente para conter apenas informações relevantes a serem passadas aos robôs, caracterizando uma boa sintaxe. Esse protocolo deve, ainda, ser confiável, ou seja, capaz de separar dados bons de dados corrompidos, mostrando ter uma boa semântica. A comunicação de dados pode ser unidirecional, onde apenas o computador envia dados para os robôs, ou bidirecional, onde os robôs também enviam dados para o computador. Nenhuma vantagem muito evidente é vista no uso de um sistema de comunicação bidirecional que utiliza visão global.

O pacote de dados de um protocolo pode ser enviado utilizando-se o conceito de broadcast, onde todos os robôs recebem todos os dados e interpretam apenas o bloco de dados que lhe diz respeito. O pacote pode também ser enviado para um grupo restrito de robôs, ou apenas para um robô, por meio de um protocolo endereçável com informação referente ao destinatário do pacote.

Diante de todas as características e aplicações dos protocolos existentes em equipes de futebol de robôs, este

artigo se propõe a analisá-las e caracterizá-las sob o ponto de vista de um controle efetivo e em tempo real dos robôs. Este artigo propõe ainda um novo protocolo de comunicação que visa diminuir, e em alguns casos impedir, as restrições existentes nos mais diversos protocolos utilizados.

Este artigo está dividido da seguinte forma. Seção 2 apresenta as características dos protocolos utilizados em futebol de robôs e analisa suas propriedades. Na seção 3 o protocolo proposto por este artigo é detalhado. Discussões e análises sobre o protocolo proposto são mostradas na seção 4, e a seção 5 finalmente conclui o artigo.

2 CARACTERÍSTICA DOS PROTOCOLOS

A maioria dos protocolos existentes para aplicações em robôs móveis é específica para a arquitetura de hardware na qual serão utilizados. Tais protocolos podem se comunicar com os robôs enviando pacotes contendo o endereço para qual robô está sendo enviado, caracterizando um endereçamento direto, ou ainda, enviando um pacote para todos os robôs, que, de acordo com sua configuração de endereço, interpretam apenas a parte dos dados recebidos destinados a ele, comunicação classificada como tipo broadcast. Protocolos que utilizam endereçamento direto, ou seja, que possuem um valor indicativo de destinatário (seja o destinatário entendido como todos os robôs, um grupo restrito ou apenas um único robô) faz com que, enquanto determinado robô esteja recebendo o pacote de dados, os outros robôs fiquem ociosos, sem receber informação [5].

Há casos em que as estratégias de jogo aliadas a protocolos de endereçamento direto podem impedir que haja garantia que os robôs irão receber os dados em um determinado intervalo de tempo fixo, pois segundo essa mesma estratégia, robôs como o goleiro, por exemplo, teriam maior prioridade de controle e seriam “super-controlados”, enquanto os demais seriam “sub-controlados”, visto que evitar o gol é mais importante. A incerteza do intervalo de tempo em que os robôs irão receber um pacote de dados implica em uma deficiência no controle de seus movimentos, algo não desejável para um controle em tempo real, como no caso do futebol de robôs.

Nos protocolos do tipo broadcast os robôs possuem também um endereço, mas esse endereço determina apenas qual o bloco de dados do pacote recebido que será processado, sendo assim não é necessário enviar dados referentes ao destinatário [2] [3] [4].

Esse tipo de protocolo é mais interessante para aplicações que necessitem controlar mais de um robô, pois, mesmo que resulte em um pacote de dados de tamanho maior, nenhum robô ficará ocioso e o intervalo de tempo em que um novo pacote é recebido será constante. Nesse ponto deve-se considerar a velocidade de transmissão dos dados, fator que pode limitar o tamanho do pacote ou a frequência pela qual os pacotes são transmitidos, o que pode ser crítico para um controle em tempo real.

Outro ganho é que, apesar de o pacote ser maior, a quantidade de dados que o computador deverá transmitir será menor, visto que não há necessidade de transmissão de um valor referente ao destinatário do pacote. No entanto, a maioria dos protocolos do tipo broadcast tem um tamanho de pacote de dados fixo, eliminando a possibilidade de se configurar a quantidade de robôs a serem controlados e a quantidade de dados destinados a cada robô.

Os protocolos ainda podem ser classificados em unidirecionais (quando a comunicação é apenas do computador para o robô) e bidirecionais (quando os robôs também se comunicam com o computador e outros robôs). Na categoria Mirosoft e demais categorias de futebol de robôs que utilizam visão global para controle e detecção de movimento e posição dos robôs, o protocolo bidirecional não é muito utilizado. Ele se torna útil quando não há visão global, mas apenas visões locais em cada robô. Neste caso, um protocolo de comunicação bidirecional é de extrema importância para que tanto o computador (que neste caso pode ser considerado o técnico do time) quanto os demais robôs saibam a posição, direção, velocidade e outras informações de cada robô.

Além disso, protocolos podem ainda ser confiáveis ou não. A confiabilidade é importante para impedir que interferências de radiofrequência possam atrapalhar a comunicação de dados. Protocolos com bits ou mesmo bytes de conferência são mais confiáveis que protocolos que apenas utilizam a comunicação direta sem verificação do conteúdo. Protocolos de broadcast são, por vezes, mais sensíveis a este tipo de interferências por não possuírem endereçamento direto ao robô e, portanto, devem possuir algum tipo de verificação de dados, por mais simples que seja, para impedir a interferência na comunicação de dados.

A maioria dos protocolos de comunicação usados em futebol de robôs é específica para o hardware e a quantidade de robôs utilizados. Isso significa que caso o robô mude de configuração eletrônica e mecânica, ou mesmo deseje-se aumentar o número de jogadores, o protocolo deverá ser modificado. Essa característica é conhecida como inflexibilidade do protocolo que, por vezes, é caracterizado por um tamanho fixo de pacotes de informações. Um protocolo para ser flexível deve permitir ser expansível ou facilmente adaptado para suportar mudanças em hardware e software em diferentes quantidades de robôs.

A partir dessas discussões alguns protocolos usados em equipes de futebol de robôs foram analisados. O time Guaraná [2], por exemplo, utiliza um protocolo unidirecional, não confiável e de transmissão broadcast. Possui taxa de transmissão de dados de 1200 bps, com um total de 5 bytes por pacote, sendo um byte para identificação de início seguido de 3 bytes de dados, onde cada byte é referente a cada robô controlado, e mais um byte de preenchimento. Esse protocolo é do tipo broadcast, com endereçamento indireto, sendo que cada robô é pré-configurado para ler apenas um byte de dados. O protocolo é restrito ao hardware utilizado, possui tamanho fixo do pacote de dados e quantidade fixa de robôs a serem

controlados. A baixa taxa de transferência faz com que um pacote demore aproximadamente 33 ms para ser transmitido para os robôs.

O time coreano CENDORI [3][4] utiliza um protocolo unidirecional, com taxa de transmissão de 2000 bps, com um total de 17 bytes por pacote de dados, sendo 1 byte para identificação do início do pacote, 5 blocos de 3 bytes de dados cada para os 5 robôs a serem controlados e 1 byte sem utilidade, para um possível uso futuro. Esse protocolo é do tipo broadcast, com endereçamento indireto. Isso significa que os robôs irão receber todos os dados e, de acordo com o endereço que lhes foi atribuído, apenas um dos 5 blocos de 3 bytes de dados será processado pelos robôs. Não há nenhuma checagem de erro dos dados recebidos, portanto, não é confiável. A baixa taxa de transferência de dados faz com que um pacote demore 78 ms para ser recebido, segundo as informações contidas em [3][4]. Para o futebol de robôs, onde geralmente o tempo de captura de um quadro da imagem demora 33 ms, a comunicação de dados seria um gargalo e comprometeria o controle preciso dos robôs.

Um outro protocolo, descrito em [5], também é unidirecional e transmite dados a uma taxa de 38400 bps. Encapsula os 3 bytes de dados com um byte de início e um de finalização, para determinar quando a recepção de um pacote de dados é iniciada e quando deve ser finalizada, para então esse pacote poder ser processado. Assim, qualquer byte recebido antes do byte de início é ignorado. Caso não seja recebido o byte de finalização para completar o recebimento de um pacote, após um intervalo de tempo pré-determinado esse pacote incompleto é ignorado. Esse protocolo não é tipo broadcast, pois foi desenvolvido para o controle de apenas um robô móvel e possui um delay de controle de 100 ms. Assim como no protocolo do CENDORI, esse protocolo não possui checagem de erros e não é expansível. No entanto, por encapsular os dados em um pacote com bytes de início e de finalização, caso haja falha no link de comunicação de dados, quando o link for restabelecido tudo volta a funcionar normalmente, pois o byte de início re-sincroniza a comunicação de dados.

Para que esse protocolo possa ser utilizado para o controle de mais de um robô, caso do futebol de robôs, uma modificação nos bytes de dados poderia ser feita para inserir o endereço do destinatário. Como já citado, um protocolo com endereçamento direto não é o mais indicado para controle de um grupo de robôs móveis, como um time de futebol de robôs.

Diante tais análises e conclusões, um novo protocolo de comunicação é proposto neste artigo. Este protocolo, atualmente utilizado no time de futebol de robôs do Centro Universitário da FEI, RoboFEI, possui características que permitem que este seja classificado como confiável e flexível, além de manter o padrão de unidirecional e com transmissão broadcast comum entre os protocolos usados em futebol de robôs com visão global.

3 O PROTOCOLO PROPOSTO

Visando uma aplicação direta para o futebol de robôs, mas objetivando compatibilidade com diversas arquiteturas de robôs móveis, extrapolando o domínio do futebol de robôs, desenvolveu-se um protocolo que une as vantagens dos protocolos citados anteriormente e resolve alguns problemas apontados.

O protocolo é independente da arquitetura de hardware, pois robôs com diferentes configurações de rodas e motores podem ser utilizados, considerando algumas características necessárias ao hardware, como chaves ou jumpers de configuração de endereços dos robôs e a forma a qual tais endereços são interpretados, para a implementação desse protocolo. É flexível quanto ao tamanho do pacote de dados, por ser encapsulado com byte de início e byte de finalização. A figura 1 ilustra um pacote genérico de dados:



Figura 1 – Pacote genérico de dados

Isso faz com que o protocolo seja independente do número de robôs a serem controlados e da quantidade de bytes de dados destinados a cada robô. No entanto não é possível que robôs com endereços diferentes recebam quantidade de bytes de dados diferentes. É um protocolo do tipo broadcast e o endereçamento é feito indiretamente.

É necessário que o robô tenha chaves de configuração. Essas chaves têm a função de determinar o endereço do robô e o número de bytes de dados a serem recebidos. Por exemplo, 6 chaves são disponibilizadas para a configuração. Admitindo que cada chave representa um bit e que desses 6 bits, 3 seriam para endereço e 3 para a quantidade de bytes de dados, tem-se um número máximo de 8 endereços, ou seja, até 8 robôs podem ser controlados e cada robô pode receber até 8 bytes de dados, totalizando em 64 bytes de dados. Como o pacote é encapsulado com byte de início e byte de finalização, tem-se um pacote de dados de 66 bytes, ou 528 bits. Admitindo que a taxa de transmissão de dados seja de 19200 bps, o tempo de transmissão de um pacote seria de 27,5 ms. A taxa de transmissão de dados varia de acordo com a solução de comunicação sem fios adotada.

3.1 O FORMATO DO PACOTE DE DADOS

Esse protocolo é utilizado nos times de futebol de robôs da FEI, tanto da categoria Mirosoft, como a categoria F180 da RoboCup [6].

Na categoria Mirosoft o time é composto por 3 robôs e 2 bytes de dados são destinados a cada robô, totalizando 6 bytes de dados. Encapsulando esses dados com os bytes de início e finalização, tem-se um pacote de dados de 8 bytes, como mostrado na figura2:



Figura 2 – Pacote de dados utilizado no time MirosoT

Com a taxa de transmissão de 19200 bps, cada pacote é transmitido em aproximadamente 3,3 ms. Assim, o computador irá ter um bom tempo para processar os quadros de imagem capturados e rodar o algoritmo de inteligência artificial da estratégia.

Já a categoria F180 possui 5 robôs, os quais recebem 4 bytes de dados cada um, totalizando 20 bytes de dados. Os dados encapsulados com os bytes de início e finalização resultam em um pacote de dados de tamanho de 22 bytes, como a figura 3 ilustra:

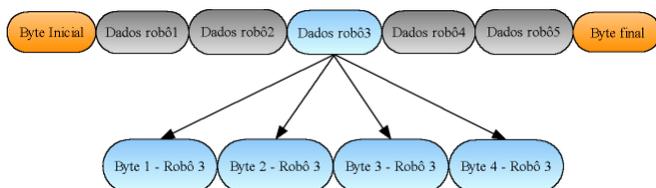


Figura 3 – Pacote de dados utilizado no time F180

Com essa mesma taxa de transmissão de 19200 bps, cada pacote de dados será transmitido em aproximadamente 9,2 ms. Apesar de a transmissão de um pacote de dados do time F180 demorar praticamente 3 vezes mais do que a transmissão de um pacote de dados do time MirosoT, esse tempo ainda é muito baixo, visto que a janela de tempo para um loop do programa (o processamento da imagem, a execução do algoritmo de inteligência artificial e a transmissão do pacote de dados) é de 33 ms, pois a placa de aquisição utilizada captura a imagem a uma amostragem de 30 quadros por segundo.

Em ambos robôs as chaves de configuração são lidas e interpretadas de maneira que o robô armazene seu endereço em uma variável denominada ENDEREÇO, a quantidade de bytes de dados destinados a ele em uma segunda variável denominada NUM_BYTES e o número de robôs que estão sendo controlados em uma terceira variável denominada NUM_ROBOS. O produto das variáveis NUM_BYTES por NUM_ROBOS determina o valor da variável TOTAL_BYTES, que representa o número de bytes do pacote de dados. O protocolo é do tipo broadcast, então todos os robôs recebem o pacote inteiro de dados. Todos os bytes recebidos são ignorados, enquanto não for recebido o byte de início de pacote. Ao ser recebido o byte de início, os próximos bytes recebidos serão bytes de dados. Ao receber um byte de dados, uma variável denominada BYTES_RECEBIDOS é incrementada. Enquanto o valor da variável BYTES_RECEBIDOS não for igual ao valor da variável TOTAL_BYTES, significa que ainda falta receber bytes de dados. Assim que a variável BYTES_RECEBIDOS tiver o mesmo valor da variável TOTAL_BYTES o robô entende que recebeu todos os bytes de dados e supõe que o próximo byte a receber seja o byte de finalização. Ao receber o byte, então, é feita a comparação. Caso esse byte seja igual ao byte de finalização, foi recebido um pacote de dados completo que está pronto para ser processado,

interpretado, caso contrário, todos os bytes recebidos são ignorados, pois não se pode assegurar que os dados são válidos. Existe ainda um tempo máximo de espera entre o recebimento de um byte e outro. Caso esse tempo seja excedido, a recepção do pacote atual é cancelada e um novo byte de início é esperado.

3.2 INTERPRETAÇÃO DO PACOTE DE DADOS RECEBIDO

Ao receber um pacote de dados completo, analisando o time F180 como exemplo, tem-se 20 bytes de dados, divididos em 5 blocos de 4 bytes. Para que cada robô saiba qual o primeiro byte de dados destinado a ele, após a leitura das chaves de configuração e atribuição dos valores às variáveis, uma variável INDICE é calculada da seguinte maneira:

$$\text{INDICE} = (\text{NUM_BYTES} * \text{ENDEREÇO}) - (\text{NUM_BYTES} - 1)$$

Essa variável aponta exatamente para o primeiro byte de dados destinado ao robô cujo endereço é ENDEREÇO. Esse primeiro byte é lido e verificado e uma variável denominada BYTES_LIDOS é incrementada, assim como também é incrementada a variável INDICE. Quando seu valor se igualar ao valor de NUM_BYTES, todos os bytes de dados referentes ao robô já terão sido lidos e verificados. Para exemplificar, suponha que as chaves de configuração foram posicionadas de tal maneira que o endereço do robô seja 4. Como os robôs do time F180 recebem 4 bytes de dados cada um, então:

$$\text{INDICE} = (4 * 4) - (4 - 1) = 16 - 3 = 1$$

Logo, INDICE aponta para o 13º byte recebido, sendo que os seguintes bytes 14, 15 e 16 também são destinados ao robô cujo endereço é 4.

Assim que um byte de dados é lido, ele é verificado. O protocolo ainda estabelece bits de conferência, para minimizar a possibilidade de receber dados corrompidos, inválidos. Essa verificação faz a checagem desses bits de conferência. Analisando os robôs do time MirosoT como referência, cada byte de dados é referente à velocidade de uma das rodas. O bit mais significativo de cada byte determina o sentido de rotação da roda, enquanto os 5 bits seguintes determinam a velocidade, que pode variar de 0 até 31, ou seja, 32 níveis de velocidade. Os 2 bits menos significativos são os bits de conferência gerados. A figura 4 mostra em detalhes o pacote de dados do time MirosoT.

3.2.1 OS BITS DE CONFERÊNCIA

Os bits de conferência são dois bits inseridos no byte de dados gerados a partir de uma lógica simples, para tornar o protocolo mais confiável e seguro em relação a falhas e recepção de dados corrompidos. O bit 0 é o inverso do bit 3, enquanto o bit 1 é o inverso do bit 7. Como exemplo, para que a roda gire em sentido horário o bit 7 será 0, com velocidade 15, o byte será $(001111xx)_2$. Gerando os bits de conferência chega-se ao byte completo $(00111110)_2$, ou $(62)_{10}$.

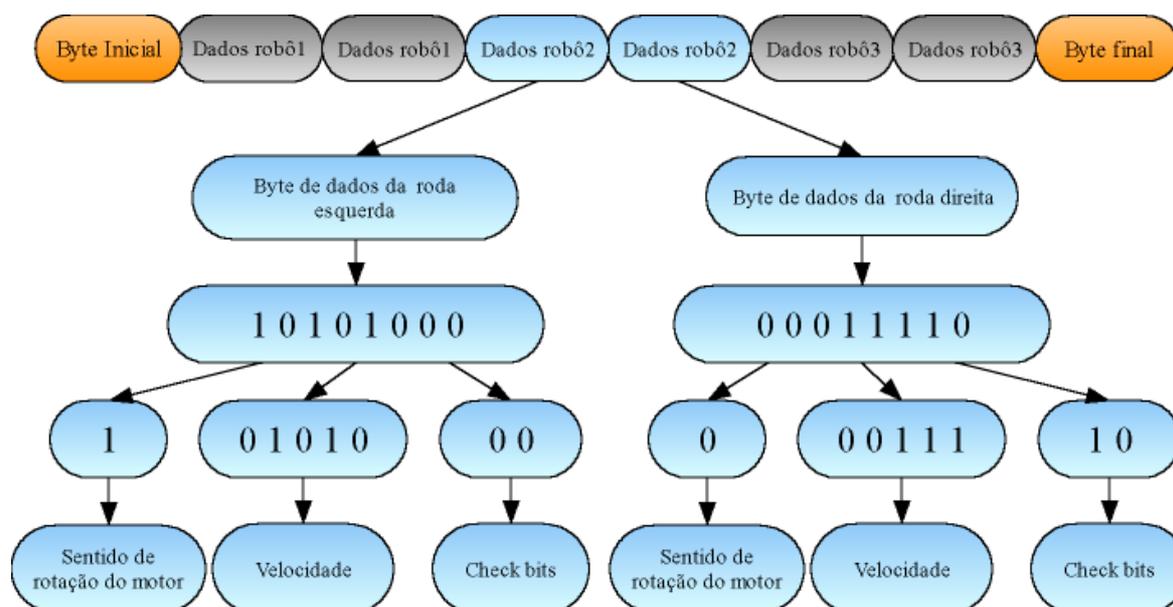


Figura 4 – O significado dos bytes do pacote do time Mirosoft

Para verificar os bytes, os bits de conferência recebidos no byte são armazenados em uma variável. Em outra variável são gerados os novos bits de conferência de acordo com a informação recebida. Caso os valores das variáveis sejam iguais, ou seja, os bits de conferência recebidos sejam iguais aos bits gerados, significa que houve sucesso na checagem dos bits. Caso os bits de conferência recebidos não sejam iguais aos bits gerados, então o byte é invalidado e assim todo o pacote de dados é descartado, pois não se pode assegurar a confiabilidade das informações contidas nesse pacote e atribuir o controle de velocidade de apenas uma roda, já que elas trabalham em conjunto para executar o movimento esperado. Assim, se a checagem não obteve sucesso ela não é confiável, e é aguardado um novo pacote de dados, que irá passar por nova verificação, para só assim essa informação poder ser executada.

O principal motivo pelo qual optou-se por utilizar esse método, em comparação a outros métodos de detecção de erros mais elaborados, foi a necessidade de exigir pouco recurso computacional do hardware em que será implementado.

3.2.2 CONTEÚDO DOS PACOTES

A figura 4, que exemplifica bem o pacote de dados do time Mirosoft, mostra que cada byte de dados é referente a uma roda, a qual é acionada por um motor. O byte mais significativo define o sentido de rotação do motor e, conseqüentemente, da roda. Os cinco bits seguidos representam um valor discreto de velocidade que varia de 0 até 31. Essa velocidade é convertida em um valor de ciclo ativo para o período de um sinal de onda quadrada. Esse valor representa a largura do pulso desse sinal que, quando aplicado em um motor, ao variar a duração desse pulso, tem-se uma variação proporcional na velocidade de rotação. Esse tipo de controle onde é aplicada um pulso de largura variável com período constante é chamado de PWM. Os dois bits menos significativos são os bits de

conferência, como já dito, e assim é feito o controle dos movimentos dos robôs do time Mirosoft, controlando velocidade e sentido de rotação dos motores, que, por sua vez, acionam as rodas.

4 DISCUSSÕES E ANÁLISES

A principal vantagem que pode ser apontada no protocolo proposto é sua flexibilidade. Destinado à estabelecer uma comunicação de dados para o controle simultâneo de robôs móveis em tempo real. Como mostrado, esse protocolo pode ser utilizado em diversas plataformas, pois o encapsulamento dos dados permite que qualquer informação entre os bytes de início e finalização seja transmitida. Essa flexibilidade pode ser observada também na possibilidade de transmitir os dados sem os bits de conferência. Dessa maneira, havendo outras checagens de erro durante a recepção dos dados, pode não ser necessário que esses bits de conferência sejam implementados, fazendo com que todos os bytes contêm apenas informações de controle para os robôs. Os bits de conferência foram implementados para dar mais confiabilidade à transmissão dos pacotes de dados.

Outra vantagem importante, conseqüência do formato do pacote de dados, é que mesmo que o link de comunicação de dados seja perdido, ao ser restabelecido, a transmissão dos dados voltar a funcionar normalmente, pois pacotes incompletos são ignorados e o byte de início se encarrega de re-sincronizar a transmissão.

Trabalhando no modo broadcast não é necessário endereçar diretamente os pacotes de dados, isto é, não é necessário inserir um byte de endereço de destinatário no pacote de dados. Assim, todos os bytes são informações de controle e cada robô irá interpretar apenas os bytes que se referem ao seu endereço, como já explicado.

A princípio a idéia de se implementar um protocolo do tipo broadcast pode não parecer uma vantagem muito clara, pois todos os robôs terão que receber todos os bytes

de dados de todos os robôs. Aparentemente os robôs iriam dispensar mais tempo na recepção dos dados, já que teriam que receber o pacote de dados inteiro. Protocolos que utilizam endereçamento direto geralmente possuem pacotes de tamanhos diferentes, pois pode passar uma quantidade de parâmetros diferente para cada ação de controle desejada. Entretanto analisando a situação de recepção de dados em um protocolo que utiliza endereçamento direto, o que vai acabar acontecendo é que o canal de recepção irá receber todos os dados. O robô terá que verificar a cada byte recebido se ele é ou não o byte inicial, caso seja, irá verificar se o próximo byte se refere ao seu endereço. Dessa maneira, mesmo não recebendo informações válidas, todos os robôs também irão ter que ficar verificando os dados recebidos.

O principal fator determinante para a utilização do modo broadcast é que dessa maneira pode-se assegurar que os robôs irão receber novas informações de controle sempre no mesmo intervalo de tempo. Geralmente em um protocolo por endereçamento direto o que acontece é que as informações de controle só são enviadas quando há a necessidade de mudança de ação de controle, não garantindo que os robôs receberão sempre um pacote de dados em um determinado intervalo de tempo. Caso o software de controle cuide para que os robôs recebam sempre um pacote em determinado intervalo de tempo, esse problema estaria resolvido. Caso o software de controle priorize algum robô, como o goleiro, por exemplo, o controle do conjunto como um todo não será satisfatório.

Uma breve análise sobre a quantidade de bytes a serem recebidos e uma comparação do tamanho dos pacotes dos protocolos tipo broadcast e por endereçamento direto pode ser feita, considerando o time Mirosoft. Os robôs recebem um pacote de dados com um tamanho de 8 bytes, no caso do modelo tipo broadcast, como já mostrado anteriormente. Em um modelo de endereçamento direto, receberia um pacote com 5 bytes, sendo o byte de início, seguido por outro byte com o endereço do destinatário, mais dois bytes de dados e, por fim, o byte de finalização. Um pacote por endereçamento direto iria ser transmitido em 62,5% do tempo gasto na transmissão de um pacote por broadcast. Entretanto, os robôs que não o destinatário teriam de verificar todos os bytes recebidos, ou seja, verificar todos os bytes dos pacotes dos outros robôs também. O tempo de execução de uma verificação, apenas, e o tempo de execução de uma verificação com armazenamento do byte recebido é da ordem de alguns microssegundos, na maioria dos microcontroladores, podendo ser considerado praticamente o mesmo tempo. Sendo assim, enquanto deveria se verificar 15 bytes (3 pacotes de 5 bytes), apenas é recebido um único pacote com 8 bytes, o que gastaria 53,3% do tempo da transmissão por endereçamento direto.

A desvantagem observada no protocolo proposto é o limite prático do tamanho do pacote de dados, que está diretamente relacionado com a velocidade de transmissão. Essa limitação é entendida como prática porque em teoria o tamanho do pacote não tem limite. Como é desejado um controle em tempo real, não se pode transmitir grandes quantidades de dados para os robôs, pois quanto menor o

tempo em que uma nova ação de controle para corrigir movimentos do robô, mais preciso será seu controle. Esse delay de tempo pode ser aceitável em algumas aplicações, onde o robô tem certa autonomia, mas geralmente é necessário que a comunicação seja feita o mais rápido possível, como é o caso do futebol de robôs, onde se tem uma janela de tempo de aproximadamente 30 ms para que um loop completo do sistema seja efetuado.

A velocidade de transmissão influencia o limite prático do tamanho do pacote, pois é essa velocidade que determina o tempo de transmissão de um pacote para os robôs. Um pacote de dados com tamanho de 8 bytes transmitido a uma taxa de 9600 bps consumiria aproximadamente 5 ms. Nesse mesmo intervalo de tempo, utilizando uma taxa de transmissão de 19200 bps, um pacote de dados com tamanho de 12 bytes poderia ser transmitido. Então, a taxa de transmissão de dados pode influenciar no tamanho do pacote de dados, assim como o tempo que se pode esperar para que essa transmissão aconteça, ou seja, a janela de tempo disponível para tal.

Não por acaso, a taxa de transmissão de 19200 bps foi utilizada como exemplo anteriormente. O módulo de radiofrequência, modelo ER400TRS [7], da fabricante LPRS, que se pretende utilizar, estabelece um link de comunicação sem fios a uma taxa de transmissão máxima de 19200 bps. Até o presente momento esses módulos ainda não foram disponibilizados para que testes pudessem ser efetuados, mas trata-se de uma solução pronta para ser utilizada, que faz a codificação/decodificação e checagem de erros. Optou-se por utilizar uma solução de comunicação sem fios pronta ao invés de desenvolver um sistema novo, dada a complexidade e alto custo do desenvolvimento de circuitos de radiofrequência.

Atualmente o protocolo está funcionando de acordo com o descrito, mas a comunicação é feita por fios. O sistema se comporta exatamente com as características descritas.

Futuramente, quando os módulos de radiofrequência estiverem disponíveis em nosso laboratório, testes serão efetuados para verificar alguns aspectos como a taxa de erro e suas variações em diferentes taxas de transmissão, interferências e imunidade a ruído, assim como o comportamento do sistema e o funcionamento do protocolo em um link de dados sem fio.

5 CONCLUSÃO

Este artigo propõe uma alternativa aos protocolos normalmente utilizados. O protocolo proposto resolve os problemas apontados e acrescenta vantagens em sua utilização.

O objetivo principal é tornar um sistema de controle de robôs móveis, principalmente sistemas que utilizam visão global, como o caso de diversas categorias de futebol de robôs, mais flexível quanto à variação do número de robôs controlados e a quantidade de dados a ser transmitida para os robôs. Um sistema com essa flexibilidade possibilita o controle de, por exemplo, dois times de futebol de robôs com arquiteturas de hardware completamente diferentes, como os times das categorias Mirosoft e F180, sem a

necessidade de alteração ou adaptação de hardware, nem mesmo firmware.

Esse é o primeiro passo para se ter um sistema de controle de robôs móveis que interagem entre si, independente da plataforma. Um protocolo de comunicação que funciona com diversos modelos de robôs e diferentes tamanhos de pacotes de dados descarta a necessidade de qualquer adaptação quando a plataforma é alterada.

O protocolo proposto, então, faz parte do objetivo de se ter um sistema totalmente configurável, tanto quanto ao número de robôs controlados, como a quantidade de dados a serem enviados para os robôs, que não necessita de adaptação alguma ou desenvolvimento de novo software ou hardware.

REFERÊNCIAS

- [1] FIRA Small League Mirosoft Game. Disponível em: <http://www.fira.net/soccer/mirosot/overview.html> (acessado em maio/2005).
- [2] Costa, A. H. R.; Pegoraro, R.; "Construindo Robôs Autônomos para Partidas de Futebol: O Time Guaraná", SBA Controle & Automação Vol. 11 nº3, Set., Out., Nov., Dez. 2000.
- [3] Kim, K. H.; Ko, K. W.; Kim, J. G.; Lee, S. H.; Cho, H. S.; "The Development of a Micro Robot System for Robot Soccer Game", Proc. of 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, pp. 644-647, April 1997.
- [4] Park, S. W. ; Kim, J. H.; Kim, E. H.; Oh, J. H.; "Development of a Multi-Agent System for Robot Soccer Game", Proc. of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, pp. 626-629, April 1997.
- [5] Wolf, J. C.; "Fast Autonomous Mobile Robot Platform", University of Plymouth, 2003-2004.
- [6] RoboCup F180 Small-sized league. Disponível em: <http://www-2.cs.cmu.edu/%7Ebrettb/robocup/> (acessado em agosto/2005).
- [7] ER400TRS datasheet, LPRS; <http://www.lprs.co.uk>.