

Interface Humano-Robô e Arquitetura de Controle para Robôs Móveis controlados por *Smartphones*

Caetano Mazzoni Ranieri, Humberto Ferasoli Filho
Departamento de Computação: Faculdade de Ciências
Universidade Estadual Paulista, UNESP
Bauru, Brasil
[cmranieri, ferasoli]@fc.unesp.br

Silas Franco dos Reis Alves
Departamento de Engenharia Elétrica e Computação:
Escola de Engenharia de São Carlos
Universidade de São Paulo, USP
São Carlos, Brasil
silas.alves@gmail.com

Resumo—Este artigo descreve um sistema de robótica móvel voltado para interação humano-robô, em que um *smartphone* controla um robô móvel de pequeno porte em que se encontra embarcado. Pretende-se trabalhar dentro do campo da robótica social e assistiva, bem como no desenvolvimento de interfaces humano-robô. O padrão comportamental do sistema proposto baseia-se em explorar o ambiente, visando estabelecer uma interação por voz ao se detectar uma face na imagem capturada pela câmera do *smartphone*. O *hardware* do robô é apresentado, abrangendo tanto o *design*, como a mecânica e a eletrônica envolvidos. A arquitetura de *software*, em camadas, é descrita no sentido de contextualizar a arquitetura de controle, baseada no modelo de arquitetura de subsunção.

Palavras-chave—robótica móvel; interface humano-robô; arquitetura de controle; *smartphones*.

I. INTRODUÇÃO

O interesse em projetos de robôs sociais encontra diferentes motivações. Para crianças, esses robôs podem servir como brinquedos ou educadores. Para adultos, como assistentes em ambientes de trabalho ou em casa. Para idosos, como companhia ou ferramenta para suprir limitações físicas [1]. O desenvolvimento de interfaces humano-robô tem sido tema bastante explorado em pesquisas envolvendo robótica social ou assistiva. Em alguns casos, faz-se uso de arquiteturas baseadas em emoções para facilitar e tornar crível essa interação, bem como para favorecer estudos relativos à neurobiologia de seres vivos e criar um campo de testes para teorias acerca de emoções em seres vivos [2].

Ainda que não exista, na atualidade, um robô que possa ser entendido como ser vivo, trabalhos sugerem que crianças podem perceber robôs como formas de vida [3]. Entende-se que crianças podem conceber robôs com características humanas e inventar histórias que retratam robôs como seres sociais, fato possivelmente inspirado pela popularização de heróis robóticos em livros infantis, revistas em quadrinhos, videogames e desenhos animados, entre outros [4]. Portanto, a criação de robôs dotados de comportamentos com os quais a criança possa interagir pode propiciar um cenário promissor para o desenvolvimento de atividades lúdicas ou pedagógicas.

O Grupo de Integração de Sistemas e Dispositivos Inteligentes (GISDI), da Universidade Estadual Paulista (UNESP), vem desenvolvendo, em parceria com pesquisadores de outras universidades, projetos visando o uso da robótica móvel em atividades com crianças portadoras de deficiência motora [5]. Em trabalhos mais recentes, tem-se feito o uso de *smartphones* para controle de características funcionais dos robôs e para a interação com o ser humano, fazendo uso da vasta disponibilidade de sensores presente nesses dispositivos [6].

O propósito deste projeto é desenvolver uma arquitetura de controle baseada em comportamentos para robôs móveis de pequeno porte. A ideia é que o usuário possa interagir com o robô de forma semelhante à interação com um animal de estimação. O sistema foi desenvolvido para funcionar em *smartphones*, com sistema operacional Android, embarcados em um robô móvel de pequeno porte. Espera-se que esse sistema possa ser usado no desenvolvimento de atividades com crianças, além de servir como plataforma para desenvolvimento de novos projetos de interface humano-robô.

II. SISTEMA PROPOSTO

O ambiente é compreendido por um *smartphone* embarcado no robô móvel Roburguer, apresentado mais detalhadamente na próxima seção. O robô e o *smartphone* comunicam-se por meio da interface Bluetooth. O robô não apresenta nenhuma funcionalidade autônoma embarcada no microcontrolador, limitando-se a enviar informações de estado ao *smartphone* e receber comandos deste. Um usuário deve dar início ao sistema, configurando alguns parâmetros na tela inicial no aplicativo, desenvolvido para ambiente Android e instalado no *smartphone*, responsável tanto pela interação humano-robô quanto pelas funções básicas de controle do robô móvel. Entre os parâmetros configuráveis, estão o endereço MAC do módulo Bluetooth embarcado no robô e o valor de corte para que os sensores de proximidade infravermelho determinem que um obstáculo foi encontrado. Um ou mais usuários podem se dispor próximos ao robô, que pode funcionar sobre uma mesa de trabalho.

O sistema desenvolvido apresenta o seguinte padrão de comportamento observável: o robô explora o ambiente, enquanto evita obstáculos e alterna periodicamente o ângulo de

inclinação do *smartphone* em relação à superfície sobre a qual o robô está apoiado, a fim de aumentar a probabilidade de se detectar uma face humana por meio da câmera. Para que a face seja detectada, é necessário que esteja completamente enquadrada dentro da imagem capturada pela câmera do *smartphone*. Isso significa que, se a face estiver demasiadamente próxima à câmera, a detecção de face não ocorrerá.

Encontrada uma face, o robô para de explorar e inicia-se a interação através de comandos de voz simples. Por meio dessa interação, o ser humano pode controlar o robô, que obedece aos comandos, exceto quando detectar obstáculo. Por se tratar de um aplicativo para ambiente Android, o sistema é interrompido se o usuário aciona, no *smartphone*, comandos que pausam a aplicação, como o botão *voltar*, a qualquer momento. A interação por voz pode terminar de quatro maneiras distintas: o usuário emite um comando de voz específico para essa finalidade, o robô fica sem receber comandos por um período excessivamente longo, ocorre alguma falha no sistema de reconhecimento de fala ou o aplicativo é pausado por comando externo. Terminada a interação, exceto no último caso apontado, o robô volta a explorar o ambiente. A máquina de estados da Fig. 1 ilustra esse padrão de comportamento.

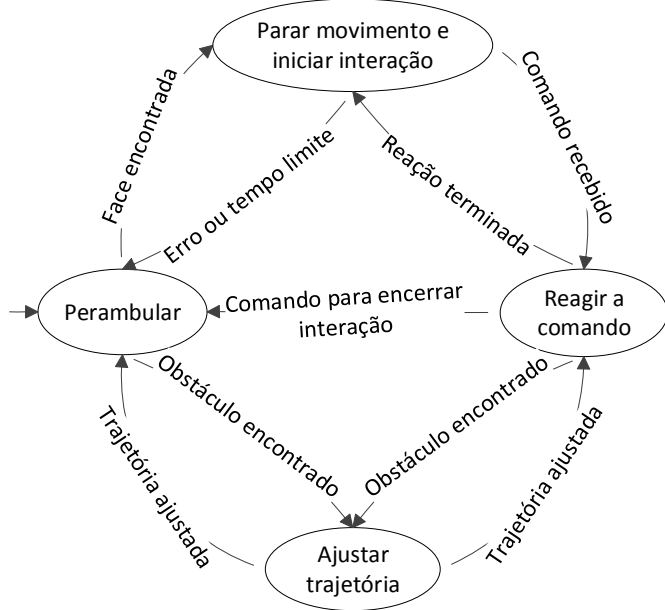


Fig. 1. Máquina de estados, iniciada no estado *Explorar*, representando o padrão comportamental observável do sistema.

A imagem capturada pela câmera é exibida na *display* do *smartphone*, podendo conter retângulos, de cor verde, delimitando as regiões em que faces são detectadas. Além de receber comandos de voz, o *smartphone* também se comunica com o usuário, por meio de um sintetizador de voz. As falas do *smartphone*, que, em uma abstração, podem referir-se ao robô em que ele está embarcado, revelam o estado interno do sistema, além de requisitar comandos ao usuário. Ao encontrar uma face, por exemplo, o sistema diz: “Olá! Quer que eu faça alguma coisa?”. Os comandos reconhecidos são repetidos ao

usuário, a fim de que este possa certificar-se de que o comando foi interpretado corretamente.

III. O ROBÔ MÓVEL

Uma fotografia do robô Roburguer, construído pelo GISDI, é mostrada na Fig. 2. O *design* e a arquitetura de *hardware* do robô foram desenvolvidos como parte de projetos anteriores, voltados a assistir crianças portadoras de deficiências motoras severas [7]. As diretrizes para a concepção do Roburguer vêm da necessidade de o robô apresentar-se de forma amigável, ao mesmo tempo que se desperta curiosidade acerca do seu funcionamento. Essa finalidade justifica a construção do robô baseada em placas de acrílico, que conservam a visão das partes internas do robô. Essa construção apresenta custo de fabricação aproximado de USD 80,00.



Fig. 2. Robô Roburguer, com *smartphone* embarcado.

Sobre o robô, é colocado um *smartphone*, cujo ângulo de inclinação em relação ao plano sobre o qual o robô está apoiado pode ser ajustado por um motor servo padrão. Uma variação nesse ângulo permite a mudança de visão de objetos, pessoas ou caminhos, o que pode ser usado para elaboração de interfaces humano-robô ou navegação baseada em visão computacional. A conexão com o *smartphone* é realizada por meio de comunicação sem fio Bluetooth. Portanto, para que o sistema esteja funcional, basta que o *smartphone* seja posicionado no suporte do robô e que o aplicativo responsável pelo sistema seja iniciado.

O robô é dotado de um sistema de locomoção com tração diferencial, composto por dois motores de rotação contínua. Como aproximação, pode-se dizer que o centro de massa do robô se situa no ponto médio entre as duas rodas sobre as quais esses motores atuam. Com finalidade de prover estabilidade ao deslocamento do robô, há um apoio na região traseira, em um ponto equidistante dos dois pontos de contato entre cada uma das rodas e a superfície de apoio do robô, porém afastado do eixo que os une. Três sensores refletivos de luz infravermelha são colocados na dianteira do robô, alinhados entre si e paralelos ao plano sobre o qual o robô está apoiado, podendo ser utilizados para detecção de obstáculos.

A Fig. 3 ilustra a arquitetura de *hardware* do robô. Tanto os sensores de luz infravermelha quanto os motores que servem ao sistema de locomoção e ao posicionamento do *smartphone* são conectados às interfaces de um microcontrolador. Conforme já

mencionado, inexistem funcionalidades autônomas embarcadas na programação do microcontrolador, em que atua apenas um sistema monitor. Esse sistema executa instruções e envia dados de acordo com as deliberações de um dispositivo remoto, com quem troca mensagens por meio da interface serial, ou RS-232. Um módulo Bluetooth é conectado a essa interface do microcontrolador, responsabilizando-se pela comunicação sem fio entre o robô e, por exemplo, um *smartphone*.

O tamanho e a mobilidade proporcionados pelo sistema mecânico permitem que o trabalho de desenvolvimento possa ocorrer sobre uma mesa de trabalho. Uma vez recarregado, o robô pode funcionar entre uma e quatro horas, dependendo da atividade realizada.

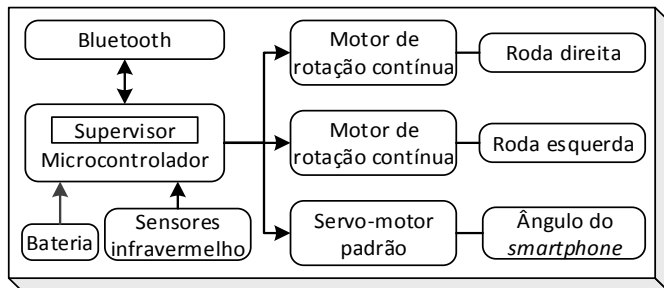


Fig. 3. Arquitetura de *hardware* do robô Roburguer.

IV. ARQUITETURA DE SOFTWARE.

A Fig. 4 ilustra a arquitetura de *software* do ambiente. Todo o sistema já foi implementado e encontra-se funcional. O *software* desenvolvido e instalado no *smartphone* faz uso de alguns recursos e bibliotecas disponíveis para projetos de aplicativos para Android. Essas bibliotecas oferecem funcionalidades que permitem o uso de abstrações, a serem colocadas como entradas para a arquitetura de controle projetada e implementada. Essa arquitetura de controle será descrita com detalhe na próxima seção. A programação foi realizada em linguagem de programação Java, com uso do pacote *Android Development Tools* (ADT), incorporado ao ambiente de desenvolvimento Eclipse.

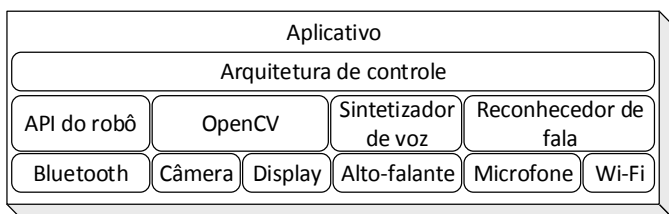


Fig. 4. Arquitetura de *software* do sistema desenvolvido.

A fim de possibilitar a programação em nível mais alto de abstração, uma API (*Application Programming Interface*) para controle do robô Roburguer foi adaptada para o sistema operacional Android. Trata-se de uma biblioteca para ambiente Windows desenvolvida como parte de projetos anteriores [8], em linguagem C++, com uso do paradigma de orientação a objetos. Essa biblioteca é compatível com diversos robôs desenvolvidos pelo GISDI, os quais também trocam mensagens

com uma base computacional remota fazendo uso do padrão de comunicação Bluetooth.

Para visão computacional, foi incorporada a biblioteca *OpenCv4Android*, utilizada para realizar detecção de face. Essa biblioteca captura quadros usando as câmeras traseira ou frontal (se disponível) do *smartphone*, exibe esses quadros no *display* e processa as imagens com uso de diversos recursos, como a aplicação de classificadores para reconhecimento de padrões.

A síntese de voz demanda que um serviço para essa finalidade esteja instalado no *smartphone*. Para os experimentos realizados neste projeto, fez-se uso do *eSpeakTTS*, serviço para síntese de fala em português. O áudio sintetizado é reproduzido com uso do alto-falante do *smartphone*.

O reconhecimento de fala é realizado pelo sistema do Google, incorporado nativamente à API de desenvolvimento para Android. O áudio ambiente, capturado com uso do microfone, é enviado para o serviço de reconhecimento. Para *smartphones* com versão do Android inferior à versão 4.1, conhecida como *Jelly Bean*, o bom funcionamento desse serviço demanda que esteja estabelecida uma conexão estável com a Internet. Isso ocorre porque o serviço para reconhecimento de fala disponibilizado pelo Google, para essas versões, consiste em processamento remoto, na nuvem. Em *smartphones* com Android 4.1 ou superior, é possível que o reconhecimento de fala seja realizado *off-line*. Nos experimentos realizados, com uso de um Samsung Galaxy Y com Android 2.3, a conexão com a Internet foi estabelecida por meio da interface de rede Wi-Fi.

V. ARQUITETURA DE CONTROLE

Embora a implementação intuitiva de conjuntos de regras de controle possa adequar-se a alguns sistemas robóticos relativamente simples, dificilmente atenderá aos requisitos de sistemas mais complexos. Por esse motivo, os mecanismos de controle de um sistema robótico podem ser organizados segundo algumas diretrizes e restrições. A essa organização, dá-se o nome de arquitetura de controle [9].

A arquitetura de controle implementada seguiu os princípios da arquitetura de subsunção (*subsumption architecture*), proposta por Rodney Brooks [10]. Subsumir é considerar algo como parte de um todo maior. Por exemplo, na zoologia, situar uma classe dentro de um filo, e um filo dentro de um reino. No caso da arquitetura de subsunção, o sistema é organizado em módulos comportamentais dispostos hierarquicamente em níveis de competência, cada qual compreendendo um subsistema de controle. Quanto mais elevado o nível de competência, mais específico o sistema definido por ele. Em outras palavras, um nível caracteriza-se por subsumir todos os níveis que estiverem acima dele. Como particularidade da arquitetura proposta por Brooks, um nível de competência pode suprimir as entradas ou inibir as saídas dos níveis inferiores. A Fig. 5 ilustra o modelo da arquitetura de subsunção utilizado para elaboração da arquitetura de controle do sistema desenvolvido. Note que somente a inibição de saídas é realizada, não havendo suporte para a supressão de entradas.

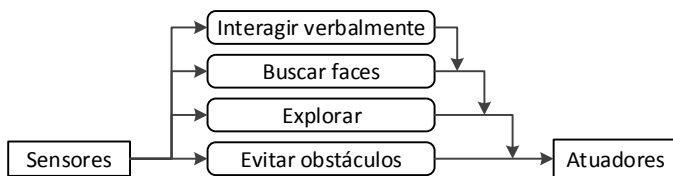


Fig. 5. Arquitetura de controle projetada e implementada.

O sistema desenvolvido baseia-se em quatro níveis de competência. Do mais baixo para o mais alto, são eles: *evitar obstáculos*, *explorar*, *buscar faces* e *interagir verbalmente*.

No nível *evitar obstáculos*, os sensores de proximidade são verificados continuamente. Enquanto se estiver detectando obstáculo, o robô retrocede. Embora os três sensores apontem para frente, é possível verificar, dependendo de quais sensores estejam detectando obstáculo, qual a melhor estratégia para retroceder ao mesmo tempo que se ajusta a postura do robô. Caso o sensor à direita esteja detectando um obstáculo e o sensor à esquerda não, por exemplo, pode-se dizer que o obstáculo em questão situa-se à direita do robô. Nesse caso, o robô retrocede alterando a sua postura no sentido anti-horário. O inverso ocorre quando um obstáculo é detectado pelo sensor à esquerda esteja detectando um obstáculo e o sensor à direita não. Nas demais situações, o robô retrocede em uma trajetória linear.

No nível *explorar*, o robô se desloca pelo ambiente com base em movimentos aleatórios. São realizados um deslocamento linear, na direção em que está posicionado, e uma rotação do robô em torno do seu próprio eixo, alterando a sua postura. Tanto a velocidade quanto a duração de ambos os movimentos são determinados de forma aleatória. No movimento de giro, determinou-se que a probabilidade de o robô girar no sentido horário é de 70%. Essa deliberação ocorreu tendo em vista que, se as probabilidades de o robô girar nos sentidos horário e anti-horário fossem iguais, o robô se deslocaria, em média, para frente, afastando-se indefinidamente do ponto de partida.

No nível *buscar faces*, o ângulo de inclinação do smartphone é modificado periodicamente, dentro de um intervalo, de modo a aumentar a probabilidade de se encontrar uma face.

Finalmente, no nível *interagir verbalmente*, o robô inicia, ao encontrar uma face, uma interação por voz, por meio da qual o usuário pode controlar o robô através de comandos. A interface humano-robô, baseada em comandos de voz, está incorporada a esse nível. Ao iniciar a interação, o nível *interagir verbalmente* inibe as saídas dos dois níveis imediatamente inferiores (*explorar* e *buscar faces*), preservando o nível *evitar obstáculos*.

VI. EXPERIÊNCIAS E RESULTADOS

A fim de validar o sistema desenvolvido, alguns ensaios foram realizados. Esses ensaios ocorreram dentro da UNESP, no Laboratório de Integração de Sistemas e Dispositivos Inteligentes (LISDI) e na sala multiuso do Laboratório de Ensino, Pesquisa e Extensão em Computação (LEPEC). Consistiram em alguns testes individuais dos quatro subsistemas que compreendem a arquitetura de controle, a fim de verificar a funcionalidade de cada um. Os procedimentos

para teste de cada nível e os resultados correspondentes são descritos a seguir:

- **Evitar obstáculos:** o robô foi posicionado sobre uma mesa, onde permaneceu imóvel até que um obstáculo fosse colocado próximo a um dos sensores de proximidade. Dessa forma, foi possível conduzir o robô a diferentes posições alcançáveis, bastando posicionar convenientemente objetos nas proximidades dos sensores.
- **Explorar:** sobre a mesa, foi colocada uma plataforma plana de 60 cm x 80 cm, ao redor da qual foi posicionada uma borda plana de metal com altura de 5 cm, perpendicularmente à mesa. Essa borda pôde ser usada para impedir que o robô caísse da plataforma, ao mesmo tempo que serviu como obstáculo detectável pelos sensores de proximidade. Nos testes realizados, foram posicionados, também, alguns blocos, na condição de obstáculos a serem evitados. O robô locomoveu-se em diferentes direções, alterando posição e postura constantemente. Ao detectar um obstáculo ou a borda da mesa, o sistema reagiu rapidamente, retrocedendo antes de colidir. O ajuste na posição do robô é assegurado pelo nível *evitar obstáculos*. O robô locomoveu-se de modo a explorar toda a plataforma, sem ficar preso a alguma região delimitada.
- **Buscar faces:** esse nível adiciona ao sistema alterações periódicas no ângulo de inclinação do *smartphone* em relação à superfície sobre a qual o robô está apoiado. Nos testes realizados, verificou-se que esse comportamento não interfere naquele determinado pelos níveis inferiores. São realizadas inclinações diante das quais o usuário pode ser detectado, bastando olhar diretamente para o *smartphone*.
- **Interagir verbalmente:** enquanto não houver face para ser detectada, o sistema funciona somente com comportamentos das três camadas inferiores. Detectada uma face, esse nível inibe os níveis *explorar* e *buscar faces*. Como resultado, todos os movimentos do robô param de ser realizados, e inicia-se a interação por voz. Os comandos emitidos pelo usuário e reconhecidos pelo reconhecedor de fala são repetidos pelo sintetizador de fala do *smartphone*, de modo que o usuário possa verificar que o comando foi interpretado corretamente. A latência entre o término do comando pelo usuário e o início da emissão da resposta pelo *smartphone* é de cerca de um segundo, contanto que exista conexão estável com a Internet. Em seguida, esses comandos são traduzidos em movimentos de locomoção. Quando o robô encontra um obstáculo durante a realização de um comando emitido pelo usuário, o nível *evitar obstáculos* atua, enquanto o nível *interagir verbalmente* sintetiza uma mensagem de voz pedindo cuidado e requisitando um novo comando assim que a posição do robô terminar de ser ajustada. Quando a interação é encerrada, a inibição dos outros níveis também cessa, e o sistema volta a comportar-se como anteriormente.

VII. CONSIDERAÇÕES FINAIS

Este trabalho visa criar novas possibilidades de interação, devido ao uso de arquiteturas de controle baseadas em comportamentos receptivas a comandos humanos e suportadas pelo uso de *smartphones*. Espera-se as que ações comportamentais dos robôs prendam a atenção das crianças, gerando a ideia de um pequeno animal de estimação, e, com aceitação comandos humanos, dando-se a ideia de um animal adestrado. Espera-se, ainda, que o ambiente possa ser expandido, de modo a atender diferentes demandas, seja dentro da educação de crianças, seja como plataforma experimental para o estudo de interfaces humano-robô.

A interação por voz, durante o atual estágio do desenvolvimento, limita-se a interpretar e executar comandos, fornecendo resposta funcional ao usuário. Pretende-se, em trabalhos futuros, introduzir o processamento de linguagem natural, a fim de criar uma interação rica entre o usuário e o robô, aproximando-se da ideia de um ser vivo artificial. Ainda nesse contexto, um uso mais incisivo de visão computacional também pode aumentar as possibilidades de interação do robô com humanos e com o ambiente externo, o que fomenta o interesse em aumentar o uso dessa tecnologia.

Com o rápido avanço na capacidade de processamento de *smartphones*, é possível prever que, em um futuro próximo, a capacidade de processamento desses dispositivos será compatível com a execução de algoritmos sofisticados. No caso de ser inviável que algum algoritmo seja executado pelo *smartphone*, a proposta é que seja usada uma base computacional remota, a qual se comunicaria com o *smartphone* através da rede sem fio Wi-Fi.

Referências

- [1] M. Malfaz e M. A. Salichs, "A new architecture for autonomous robots based on emotions," em *5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisboa, Portugal, 2004.
- [2] M. A. Arbib e J.-M. Fellous, "Emotions: from brain to robot," *TRENDS in Cognitive Sciences*, vol. 8, nº 12, pp. 554-561, Dezembro 2004.
- [3] S. Turkle, "Authenticity in the Age of Digital Companions," *Interaction Studies*, nº 8(3), pp. 501-517, 2007.
- [4] K. E. Bumby e K. Dautenhahn, "Investigating children's attitudes towards robots: a case study," em *Proceedings of the Third Cognitive Technology Conference*, San Francisco, 1999.
- [5] C. Valadão, T. F. Bastos, M. Bôrtole, V. Perim, D. Celino, F. Rodor, A. Gonçalves e H. Ferasoli, "Educational Robotics as a Learning Aid for Disabled Children," em *Biosignals and Biorobotics Conference (BRC)*, Vitória, 2011.
- [6] S. F. R. Alves, C. M. Ranieri, H. Ferasoli-Filho, M. A. C. Caldeira, R. Pegoraro e I. N. Silva, "A Friendly Mobile Entertainment Robot for Disabled Children," em *ISSNIP Biosignals and Biorobotics Conference (BRC)*, Rio de Janeiro, 2013.
- [7] H. Ferasoli-Filho, M. A. C. Caldeira, R. Pegoraro, S. F. d. R. Alves, C. Valadão e T. F. Bastos-Filho, "Use of Myoelectric Signals to Command Mobile Entertainment Robot by Disabled Children: Design and Control Architecture," em *Proceedings of the 3rd IEEE Biosignals and Biorobotics conference (ISSNIP)*, Manaus, 2012.
- [8] S. Alves, J. M. Rosário, H. F. Filho e R. Pegoraro, "Environment for Teaching and Development of Mobile Robot Systems," em *Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, Morelos, 2010.
- [9] M. J. Mataric, *The Robotics Primer*, MIT Press, 2007.
- [10] R. A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, pp. 14-23, Março 1986.