

INTEGRAÇÃO DAS PLATAFORMAS ANDROID E LEGO MINDSTORMS NXT PARA CONTROLE DE UM ROBÔ MÓVEL

GUILHERME ALVARENGA ANDRADE* , ANOLAN MILANÉS* , TALES ARGOLO JESUS*

**Departamento de Computação - DECOM
Centro Federal de Educação Tecnológica de Minas Gerais
Avenida Amazonas, n.7675, CEP: 30510-000
Belo Horizonte - Minas Gerais, Brasil*

Email: guilhermeaandrade91@gmail.com, anolan@decom.cefetmg.br,
talesargolo@decom.cefetmg.br

Abstract— This work presents the development of an application for mobile devices, inspired by SCADA Systems, responsible for controlling mobile robot. This application integrates *Android* and *Legó Mindstorms NXT* platforms via *Bluetooth* communication establishing a distributed client-server architecture. The result is an application capable of increasing the dynamism and process automation, and therefore suitable for the distributed robotic control. It was found that the test environment configured, even though in a smaller proportion, reproduces the deployment of this application in a real scenario of specific greater complexity.

Keywords— *Android Platform. Legó Mindstorms NXT Platform. Bluetooth. SCADA System. Distributed Systems.*

Resumo— Este trabalho apresenta o desenvolvimento de uma aplicação para dispositivos móveis, inspirada no Sistema SCADA, responsável pelo controle de um robô móvel. Esta aplicação integra as plataformas *Android* e *Legó Mindstorms NXT* por meio da comunicação *Bluetooth*, estabelecendo uma arquitetura distribuída cliente-servidor. O resultado é uma aplicação capaz de aumentar a dinamicidade e automação de processos, sendo, portanto, adequada para o controle robótico distribuído. Verificou-se que o ambiente de teste configurado, mesmo sendo em menor proporção, reproduz a implantação desta aplicação em um cenário real específico de maior complexidade.

Palavras-chave— *Plataforma Android. Plataforma Legó Mindstorms NXT. Bluetooth. Sistema SCADA. Sistemas Distribuídos.*

1 Introdução

Atualmente, o conceito de robótica está sendo cada vez mais difundido e se faz presente em uma sociedade visivelmente evoluída tecnologicamente. Os robôs deixaram de ser utilizados apenas no ambiente industrial e passaram a ser inseridos em diversas atividades cotidianas. Novos estudos científicos como a exploração de ambientes distintos (Dalmarco et al., 2003), o controle de trajetória (de Angelis Cordeiro et al., 2013) e os sistemas inteligentes autônomos (Cazangi and Figueiredo, 2001) refletem sobre a robótica proporcionando o aprimoramento dos seus conceitos e a emergência de novas tecnologias que oferecem maior flexibilidade, autonomia, interação e agilidade aos robôs.

A robótica, além de empregar conceituações referentes à mecânica, controle e automação, inteligência artificial, entre outras, passou a explorar os conceitos de outras disciplinas, como a de sistemas distribuídos. Notoriamente, uma diversidade de aplicações beneficiam-se da ação conjunta de sistemas com esta classificação para proporcionar eficiência na execução de tarefas complexas.

A integração entre a robótica e estas aplicações é responsável por possibilitar que determinadas funções sejam efetuadas dinamicamente. Isso ocorre porque os componentes localizados em diferentes locais estabelecem entre si uma comuni-

cação e, por meio da troca confiável de mensagens, enviam e recebem informações referentes à realização das tarefas. Esta estruturação permite independência física entre os componentes atuantes, oferecendo um monitoramento mais específico e menos complexo realizado à distância.

O presente artigo consiste no desenvolvimento de uma aplicação móvel distribuída centralizada baseada nos Sistemas SCADA (em português, Controle Supervisório e Aquisição de Dados). O aplicativo se comunica com o *kit* robótico por meio do protocolo de comunicação sem fio *Bluetooth*, único suportado pela plataforma *Legó Mindstorms NXT*. Esta comunicação permite controlá-lo e monitorá-lo por meio de uma interface gráfica interativa. Além disso, a aplicação é responsável por obter e registrar medições e sintetizá-las a fim de que possam ser exibidas e analisadas pelo usuário. É importante salientar que este trabalho tanto trata das estratégias de controle robótico quanto das técnicas de Sistemas Distribuídos utilizadas para melhor resolução do problema proposto. O resultado representa o incentivo à combinação de conceitos multidisciplinares e assinala o fato que essa aplicação móvel desenvolvida pode ser utilizada em um ambiente real, definindo-se os componentes atuantes, a forma de comunicação e a estrutura estabelecida entre eles.

Este artigo está organizado da seguinte forma. A seção 1 introduz o problema abordado. Na se-

ção 2 é exposta a revisão de literatura, descrevendo e apresentando os principais resultados nos trabalhos correlatos. Na seção 3, tem-se a descrição do desenvolvimento do presente trabalho. Na seção 4 apresentam-se testes e análises realizadas. Na seção 5 são descritos os resultados. Por fim, na seção 6 apresentam-se as conclusões.

2 Revisão de Literatura

O interesse por sistemas que se integram e facilitam a realização de determinadas tarefas é crescente. Portanto, a aplicabilidade deles é vasta, seja para o processamento distribuído quanto para o compartilhamento de recursos. Especificamente neste trabalho, é estabelecida uma relação direta com a robótica.

O trabalho realizado por da Silva Pisco (2015), propõe o desenvolvimento de uma aplicação *Android* que simula o comportamento de um Sistema *SCADA*. Esta aplicação visa o monitoramento e controle da planta de uma linha de produção flexível composta por cinco módulos interligados com funcionalidades distintas. O aplicativo desenvolvido funciona como MTU¹ e oferece interfaces interativas semelhantes as dos sistemas *SCADA*. A simulação das RTU's² foi realizada utilizando um simulador da linha de produção e um programa do *Unity Pro XL*, responsável por controlar a produção e movimentação das peças entre os módulos citados. Para a comunicação entre o RTU e MTU utilizou-se o protocolo *Modbus TCP/IP*.

O trabalho realizado por da Silva Júnior et al. (2013), propõe o desenvolvimento de uma aplicação *Android* que se comunica com um único robô *Lego Mindstorms NXT* utilizando o protocolo de comunicação *Bluetooth* baseado na arquitetura *cliente - servidor*, na qual o robô desempenha a função de servidor e a aplicação *Android* a de cliente. Este trabalho consiste em um aplicativo que coordena o movimento de um robô nas quatro direções possíveis (para direita, para esquerda, para frente e para trás) e define a velocidade na qual essa ação será executada.

Outro trabalho relevante realizado por Goebel et al. (2011) objetiva apresentar uma discussão sobre os possíveis métodos de conexão (direta via USB, indireta via Microcontrolador e *Bluetooth*) entre o dispositivo móvel e robô e expor uma solução técnica para modelar uma interface capaz

de realizar a comunicação entre o dispositivo *Android* e o *Lego NXT*. Neste caso, o aparelho móvel é considerado um serviço remoto responsável por encapsular e controlar a conexão com o robô. Para isso, implementou-se uma biblioteca orientada a objetos que provê uma *API* completa para integração entre as plataformas.

Outro projeto significativo, proposto por Silva et al. (2012), refere-se à criação de um aplicativo móvel em *Android* destinado ao controle de Sistemas Embarcados utilizando o protocolo de comunicação *Bluetooth*. Esse trabalho consiste no controle de um Veículo Teleoperado cujos recursos foram desenvolvidos na Plataforma de Desenvolvimento *Open Source Arduino*.

Já o trabalho proposto por Cruz et al. (2011), apresenta uma aplicação responsável pelo controle de um robô utilizando conexão *Wi-Fi* e sensores de movimento. Esta aplicação capta informações dos movimentos executados pelo celular, mapeia-os em comandos aceitos pelo robô, que são enviados por meio de uma rede sem fio *Wi-Fi*. A captação dos movimentos ocorridos é realizada por acelerômetros presentes nos dispositivos móveis, que são sensores responsáveis por coletar dados que possibilitam a estimação da sua orientação no espaço definido pelos três eixos cartesianos x , y e z .

O diferencial do presente trabalho em relação aos trabalhos apresentados consiste na combinação dos conceitos de Robótica, Sistemas *SCADA*, Sistemas Distribuídos e controle automático. Para que tal objetivo fosse alcançado, desenvolveu-se uma aplicação que simula um Sistema *SCADA* estruturado em uma arquitetura cliente-servidor e configurado a partir da integração das plataformas *Android* e *Lego Mindstorms*, utilizando a comunicação *Bluetooth* para a troca de mensagens entre os componentes atuantes.

3 Desenvolvimento

3.1 Modelagem do Problema

O objetivo do trabalho consiste em fazer o robô com posição inicial (x_i, y_i) descrever uma trajetória circular de raio R em torno de uma posição alvo (x_a, y_a) , como mostra a Figura 1. Além disso, o usuário tem autonomia para guiá-lo manualmente de modo a fazê-lo alcançar a posição desejada.

O comportamento do robô com tração diferencial pode ser matematicamente descrito pelo modelo do uniciclo cinemático (Siciliano et al., 2010), no qual considera-se que ele se movimenta com uma velocidade linear v e uma velocidade angular ω , dadas em função das velocidades angulares das rodas, e apresenta uma direção de orientação no plano cartesiano xy especificada pelo ângulo θ obtido em relação ao eixo x . Este modelo pode ser

¹Master Terminal Unit são computadores que coletam dados provenientes da RTU, armazenam-os, enviamos para interfaces de forma que possam ser exibidos nitidamente e transferem mensagens contendo comandos de controle para RTU.

²Remote Terminal Unit são componentes responsáveis por atuar diretamente sobre o sistema por meio de atuadores e recolher informações advindas dos sensores, transformá-las em sinais digitais por meio de conversores analógico-digitais e, por fim, enviá-las para as MTUs.

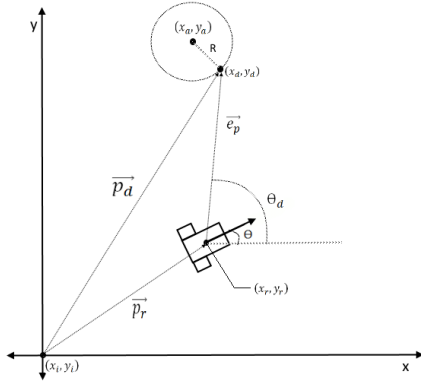


Figura 1: Modelagem do problema.

representado pelas seguintes equações:

$$\dot{x} = v \cos(\theta), \quad (1)$$

$$\dot{y} = v \sin(\theta), \quad (2)$$

$$\dot{\theta} = \omega, \quad (3)$$

em que $\dot{x}$ é a taxa de variação da coordenada x no tempo, $\dot{y}$ é a taxa de variação da coordenada y no tempo e $\dot{\theta}$ é a variação da orientação do robô no plano cartesiano.

Como o robô móvel utilizado neste trabalho é do tipo diferencial de duas rodas, de (1), (2) e (3), tem-se

$$\dot{x} = \frac{r}{2}(\omega_r + \omega_l) \cos(\theta), \quad (4)$$

$$\dot{y} = \frac{r}{2}(\omega_r + \omega_l) \sin(\theta), \quad (5)$$

$$\dot{\theta} = \frac{r}{L}(\omega_r - \omega_l), \quad (6)$$

em que ω_r é a velocidade angular da roda direita, ω_l é a velocidade angular da roda esquerda, r é o raio da roda e L é o tamanho do eixo das rodas do robô.

Define-se, portanto

$$v = \frac{r}{2}(\omega_r + \omega_l), \quad (7)$$

$$\omega = \frac{r}{L}(\omega_r - \omega_l). \quad (8)$$

Manipulando-se algebricamente (7) e (8), tem-se

$$\omega_r = \frac{2v + \omega L}{2R}, \quad (9)$$

$$\omega_l = \frac{2v - \omega L}{2R}. \quad (10)$$

A posição real (x_r, y_r) e a orientação θ do robô se alteram, podendo-se, portanto, para cada nova posição do robô, definir o vetor posição $\vec{p}_r$, o vetor posição desejada $\vec{p}_d$ e o vetor erro $\vec{e}_p$, dado por

$$\vec{e}_p = \vec{p}_d - \vec{p}_r = \begin{bmatrix} x_d - x_r \\ y_d - y_r \end{bmatrix} = \begin{bmatrix} e_x \\ e_y \end{bmatrix} \quad (11)$$

sendo e_x o erro de posicionamento no eixo x e e_y o erro de posicionamento no eixo y . A orientação desejada do robô θ_d e o erro de orientação e_θ são dados por

$$\theta_d = \text{atan2}(e_y, e_x), \quad (12)$$

$$e_\theta = \theta_d - \theta. \quad (13)$$

Para a resolução do problema foram utilizadas as duas malhas de controle em cascata e uma estratégia de navegação. A malha mais interna mostrada na Figura 2, é responsável por controlar a velocidade das rodas. A entrada de referência são as velocidades angulares desejadas da roda direita e esquerda, $(\omega_{rd}, \omega_{ld})$. Os erros obtidos da diferença entre as velocidades angulares desejadas e estimadas são as entradas para um controlador PID. Este controlador produz sinais PWM que são aplicados aos motores. Por sua vez, velocidades angulares reais, $(\omega_{rr}, \omega_{lr})$, são estimadas a partir das medidas obtidas pelos encoders³ do robô. Estes produzem medições para a realimentação negativa da malha de controle II e para a estratégia de navegação.

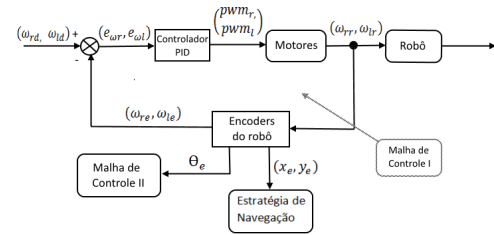


Figura 2: Malha de controle de velocidade do motor.

A malha de controle II e a estratégia de navegação, mostradas na Figura 3, controlam, respectivamente, o direcionamento e o posicionamento do robô no plano. A estratégia de navegação recebe como entrada a posição estimada, (x_e, y_e) , e a posição desejada, (x_d, y_d) , e produz a orientação desejada do robô, θ_d . Este valor será comparado com o valor recebido da malha de controle I, θ_e , e produz um erro, denominado e_θ , aplicado a um controlador proporcional-integral, de parâmetros K_p e K_i , que, por sua vez, produz a velocidade angular (ω) . Este valor, bem como a velocidade linear, são entradas para o cálculo realizado no bloco assinalado pelo número 1, em que R é o raio da roda e L o tamanho do eixo das rodas, resultando nas velocidades angulares desejadas da roda direita e esquerda, que são sinais de referência para a malha de controle I.

³Dispositivos responsáveis pela medição do movimento rotacional de um eixo a partir da contagem de pulsos elétricos.

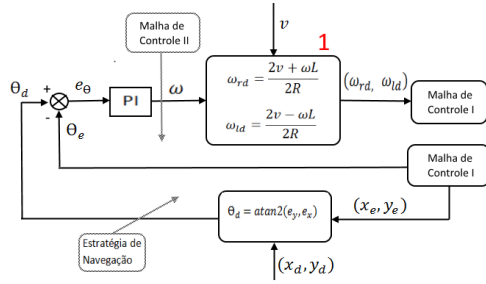


Figura 3: Malhas de controle de direcionamento do rob  e estrat gia de navega  o.

3.2 Implementa  o

A primeira defini  o realizada foi estabelecer o papel desempenhado por cada plataforma na estrutura cliente-servidor configurada entre elas. Definiu-se que a aplica  o teria o papel de cliente e o rob  a de servidor, visto que ele   respons vel pelo processamento mais complexo e pelo envio dos resultados.

A integra  o entre as plataformas   realizada com a transfer ncia de informa  es pelo canal de comunica  o criado ao estabelecer entre elas uma conex  o *Bluetooth*. A aplica  o envia as informa  es sobre os movimentos do controle manual, os valores dos ganhos dos controladores proporcional e integral inseridos pelo usu rio e os valores dos posicionamentos inicial e do alvo, e recebe do rob  as informa  es atualizadas do processo: x , y , o  ngulo θ , a velocidade linear v , a velocidade angular ω e o erro de posi  o $\|e_p\|$, que s o exibidas dinamicamente pelo aplicativo.

Existem tr s tipos de comandos que podem ser recebidos pelo rob , advindos da aplica  o: (i) o de configura  es, (ii) o de controle autom tico e (iii) o de controle manual. O comando para configura  es altera os valores das vari veis (posi  es inicial e alvo e ganhos dos controladores proporcional e integral) pr -definidos inicialmente para os novos valores recebidos utilizados no controle do rob . Os comandos de controle manual e autom tico s o respons veis, respectivamente, por guiar o rob  manualmente definindo a sua nova dire  o a cada instante de tempo e por iniciar os c lculos que o conduzam ao ponto alvo para que ele possa descrever, em torno desta coordenada, uma trajet ria circular com raio pr -estabelecido.

A velocidade v do rob    dada em fun  o da velocidade do alvo e da margem de corre  o que depende da dist ncia do rob  em rela  o a este alvo, de acordo com a express  o:

$$v = pv_a + q \tanh \left(\sqrt{e_x^2 + e_y^2} \right), \quad (14)$$

em que $p, q \in (0, 1)$ s o fatores de pondera  o.

A implementa  o do controle autom tico foi realizada com base nas equa  es definidas no modelo do uniciclo.

4 Testes

Os testes apresentados a seguir foram propostos com objetivo de variar as posi  es inicial e alvo do rob , assim como os ganhos dos controladores proporcional e integral. Portanto, escolheram-se experimentalmente os valores atribuídos a esses par metros de tal forma que a execu  o n o apresentasse comportamento oscilat rio. Para o primeiro teste, os seguintes par metros foram definidos: $(x_i, y_i) = (0, 0)$, $(x_a, y_a) = (0,90; 0,90)$, $R = 0,20$, $K_p = 1,35$, $K_i = 0,35$ e $\theta_i = \pi$ rad. A Figura 4 ilustra o resultado.

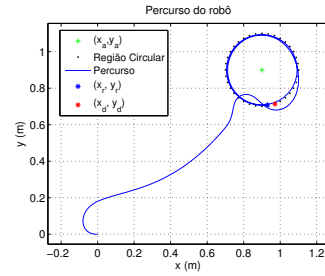


Figura 4: Trajet ria seguida pelo rob  a partir da origem do plano cartesiano.

Para o segundo teste, os par metros utilizados foram $(x_i, y_i) = (-0,1; -0,1)$, $(x_a, y_a) = (0,9; 0,45)$, $R = 0,20$, $K_p = 1,475$, $K_i = 0,385$ e $\theta_i = \pi$ rad. O resultado   mostrado na Figura 5.

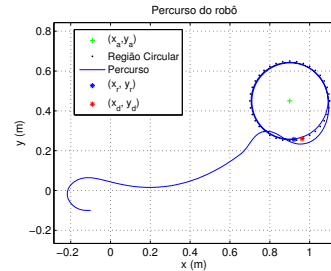


Figura 5: Trajet ria seguida pelo rob  a partir do ponto $(-0,1;-0,1)$.

O terceiro teste consistiu em alterar (x_a, y_a) e os ganhos dos controladores em tempo de execu  o. A princ pio, os par metros definidos foram: $(x_i, y_i) = (0; 0,9)$, $(x_a, y_a) = (0,3; 0,3)$, $K_p = 0,85$, $K_i = 0,25$ e $\theta_i = 0$ rad. Decorridos 22s ap s in cio da execu  o do comando autom tico, alterou-se o valor de (x_a, y_a) para $(0,9; 0)$, K_p para 1,15 e K_i para 0,55. A trajet ria realizada nas duas regi es circulares pode ser verificada pela Figura 6.

A Figura 7 mostra o comportamento do erro de posi  o para o terceiro teste. Este erro   obtido por meio da dist ncia Euclidiana entre os pontos atual e alvo, em metros, a cada instante de tempo. Inicialmente, seu valor, 0,671m,   grande devido a dist ncia entre o ponto inicial e o primeiro ponto alvo. Este erro decresce at  atingindo seu menor valor, 0,024m, aos 4,6s de execu  o. Em

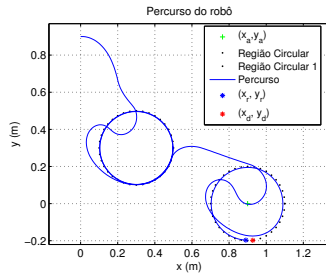


Figura 6: Trajetória seguida pelo robô a partir do ponto (0; 0,9).

seguida, o valor do erro cresce de forma oscilatória por 8s, devido à correção do posicionamento, e novamente decai até atingir o valor de 0,036m. Aos 62s, alterou-se o ponto alvo para (0,9; 0,0) e novamente o erro apresenta uma oscilação devido à mudança do ponto desejado. Após os 78s de execução o erro diminui até estabilizar com valor de 0,035m.

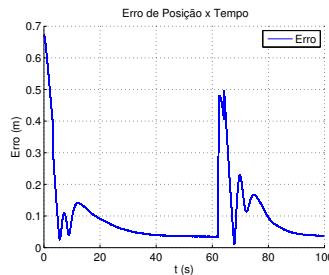


Figura 7: Erro de posição do robô.

Por fim, o último teste consistiu em guiar manualmente o robô utilizando os seguintes parâmetros: $(x_i, y_i) = (0, 0)$, $(x_a, y_a) = (0,9; 0,9)$ e $\theta_i = \pi$ rad. Para comparar a trajetória configurada ao final da execução dos movimentos, plotou-se um gráfico comparativo com a realização de um teste do controle automático com os seguintes definições: mesmos valores para (x_i, y_i) e (x_a, y_a) , $K_p = 1,35$, $K_i = 0,25$, $\theta_i = \pi$ rad. Este gráfico pode ser visto na Figura 8.

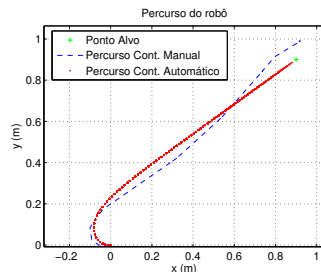
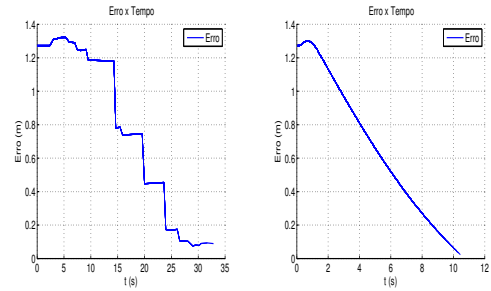


Figura 8: Comparativo entre execução do controle manual e automático.

Nota-se que os percursos realizados pelo robô tanto para o controle manual quanto automático foram similares. Para melhor análise, plotaram-se os gráficos do erro de posição para os mo-

dos de controle apresentando, respectivamente, na Figura 9b e Figura 9a. No primeiro caso, o maior valor do erro, 1,3256m, ocorreu aos 5,18s de execução no ponto (-0,0918; 0,0205) e decresceu até atingir o menor valor, 0,059m, aos 33,73s no ponto (0,9181; 0,8459). No segundo caso, o maior erro, 1,3004m, ocorreu aos 0,719s no ponto (-0,061; 0,0245) e diminuiu até o valor de 0,023m aos 10,44s no ponto (0,8813; 0,8864) do plano cartesiano.



(a) Controle manual. (b) Controle automático.

Figura 9: Erro de posição do robô.

Ressalta-se que os tempos plotados nos gráficos não são idênticos pelo fato dos testes terem sido realizados separadamente. Além disso, destaca-se o formato irregular da curva para o erro do controle manual, na qual existem espécies de degraus em que a parte horizontal indica o momento no qual o usuário clicou e pressionou o botão de uma determinada direção e após alguns segundos cessou esse movimento. Já a parte vertical representa a atualização do erro quando o usuário clicou mais uma vez para movimentar o robô em uma nova direção.

5 Resultados

A aplicação móvel desenvolvida de fato baseia-se em um Sistema SCADA, pois permite controlar, monitorar e gerenciar as variáveis do processo definidas oferecendo interfaces interativas. Além disso apresentou um comportamento esperado, proporcionando visualização do estado atual da execução e adoção de medidas de prevenção e correção ao movimento realizado pelo robô. Duas de suas telas são apresentadas pela Figura 10.

Verificou-se que a implementação do controle automático atingiu um resultado satisfatório, pois ao se escolher valores adequados para os ganhos dos controladores, o robô alcança o objetivo proposto com um erro de posição que decresce ao longo do tempo, tendendo a se anular. Ratificou-se o fato de que o comportamento do robô é influenciado diretamente pela escolha dos valores para os ganhos dos controladores. Obteve-se êxito também na implementação e execução do controle manual, visto que o robô é guiado pelo usuário por



Figura 10: Telas da Aplicação.

meio aplicação até a posição desejada. Vale ressaltar que o erro de posição para este último controle varia de acordo com o caminho percorrido pelo robô. Consequentemente, a escolha do controle deve ser realizada com base no estado atual do processo e na necessidade do usuário.

6 Conclusões e Trabalhos Futuros

6.1 Conclusões

Nesse trabalho propôs-se o desenvolvimento de uma aplicação móvel, espelhada nos Sistemas SCADA, que integra as plataformas *Android* e *Lego Mindstorms NXT*. A finalidade era simular a execução de processos, proporcionando comodidade ao controlá-los, uma vez que a aplicação foi desenvolvida para uma plataforma móvel.

A mobilidade adquirida torna os processos mais dinâmicos à medida que permite o acompanhamento em tempo real e não necessariamente de forma presencial. Para tal objetivo foram combinadas estratégias de controle e conceitos de Sistemas Distribuídos. Consequentemente, constatou-se que, mesmo em se tratando de uma simulação em menor escala, é possível estabelecer um ambiente correspondente ao que seria a implantação deste *software* em uma aplicação real específica.

6.2 Trabalhos Futuros

A perspectiva que se abre para o futuro é expandir os resultados, inserindo outros componentes robóticos que possibilitam abranger novos termos técnicos, novas tecnologias e diferentes ações de controle. Além disso, é possível combinar redes de computadores de alta velocidade e técnicas de segurança para a transmissão confiável dos dados entre os componentes para a formação de uma rede de robôs. Para tanto, seria desejável uma plataforma em que a comunicação pudesse ser realizada via *Wi-Fi*, como o *Lego Mindstorms EV3*.

Outra perspectiva é evitar que aplicação seja restrita apenas aos usuários que possuem dispositivos móveis com sistema operacional *Android* e disponibilizá-la para as plataformas móveis *iOS*

e *Windows Phone*. Outra medida relevante seria adaptar o aplicativo para aparelhos com dimensões distintas, como *tablet* e outros modelos de *smarthphones*.

Referências

- Cazangi, R. R. and Figueiredo, M. (2001). Sistema autônomo inteligente baseado em computação evolutiva aplicado à navegação de robôs móveis, Canela, Paraná.
- Cruz, B. H. A., Agnese, J. F., Fagundes, B. J., Bastos, M. T., Molz, R. F. and Schreiber, J. N. C. (2011). Desenvolvimento de uma aplicação embarcada em celular visando controle de robô via wi-fi, *Revista Brasileira de Computação Aplicada* 3(1): 43–52.
- da Silva Júnior, C. G., Borges, I. S., Cherubini, L. F. and Martins, R. M. (2013). Aplicação do sistema operacional android em robótica móvel, *XLI Congresso Brasileiro de Educação em Engenharia*.
- da Silva Pisco, R. G. P. (2015). *Scada em android*, Master's thesis, Universidade do Porto, Porto, Portugal. Mestrado Integrado em Engenharia Eletrotécnica e de Computadores.
- Dalmarco, G., de Azevedo, D. F. and Lima, J. C. M. d. (2003). Desenvolvimento de robô para exploração de água subterrânea em marte, *Salão de iniciação Científica (15.: 2003: Porto Alegre, RS). Livro de resumos. Porto Alegre: UFRGS, 2003* 15.
- de Angelis Cordeiro, R. et al. (2013). *Modelagem e controle de trajetória de um veículo robótico terrestre de exterior*, Master's thesis, Faculdade de Engenharia Mecânica. Universidade Estadual de Campinas, Campinas, São Paulo.
- Goebel, S., Jubeh, R., Raesch, S.-L. and Zuen-dorf, A. (2011). Using the android platform to control robots, in R. Stelzer and K. Jafarmadar (eds), *Proceedings of 2nd International Conference on Robotics in Education (RiE 2011)*, INNOC - Austrian Society for Innovative Computer Sciences, pp. 135–142.
- Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G. (2010). *Robotics: modelling, planning and control*, Springer Science & Business Media.
- Silva, J. F., Santos, D. M., Oliveira, K. D., Teixeira, R. G., Rodrigues, T. O., Marques, V. C. and Silva, F. D. (2012). A study of bluetooth application for remote controlling of mobile embedded systems, *Computing System Engineering (SBESC)* p. 116.