

SimRobô: simulador para robótica com propósito educacional

Marco Túlio Chella

Departamento de Computação – Universidade Federal de Sergipe (UFS)
Av. Marechal Rondon, s/n - CEP 49100-000 - São Cristóvão - SE

chella@.ufs.br

***Abstract.** Robotics activities include design, construction, integration of mechanical and electronics devices, programming a control system, debugging and evaluating the results. This set of steps has a significant financial and human cost that may limit the use of robotics as an educational tool. The simulation with software can be an option in education with robots. This paper presents the SimRobô, a software for simulate mobile robots with realtime visualization of control programs created in various programming tools.*

***Resumo.** Atividades de robótica compreendem o projeto, construção, integração dos elementos mecânicos e eletrônicos, programação de um sistema de controle, depuração e avaliação dos resultados obtidos. Esse conjunto de etapas tem um custo financeiro e humano significativo que de certa forma limita a utilização da robótica como ferramenta educacional. A simulação da operação de robôs por meio de aplicações específicas pode ser uma alternativa ou complemento para atividades relacionadas à robótica educacional. Esse trabalho apresenta o SimRobô, software para simular robôs móveis que permite visualizar graficamente em tempo real a execução de programas controladores implementados nas mais diversas ferramentas de programação.*

***Palavras chave.** simulação, robótica móvel, programação robôs, competição*

1. Introdução

Nesse trabalho designou-se por robótica educacional o ambiente constituído pelo computador, componentes eletrônicos, eletromecânicos e programa, onde o aluno, por meio da integração destes elementos, constrói e programa dispositivos automatizados com o objetivo de explorar conceitos das diversas áreas do conhecimento. Estes dispositivos automatizados passam a comportar-se como “músculos” do computador atuando no ambiente, ou seus “órgãos do sentido”, coletando dados e enviando-os ao computador onde será processado [D’Abreu 1993].

Os trabalhos de Papert (1980) e Piaget (1952) sugerem que alunos aprendem ao fazer, por pensar e refletir sobre o que foi feito. Os princípios derivados da teoria de Piaget sobre o desenvolvimento cognitivo revisados por Papert (1985) conhecido por construcionismo, sugerem que o centro do processo relacionado ao aprendizado é a participação ativa do aprendiz, que amplia seus conhecimentos por meio da construção e

manipulação de objetos significativos para o próprio aprendiz e a comunidade na qual está inserido.

A atividade de robótica em um ambiente educacional cria oportunidade para que o aluno no propósito de construir um objeto físico com determinada funcionalidade e comportamento se envolva nas etapas de projetar, construir, programar, depurar e avaliar o resultado obtido. Esse conjunto de ações coloca o aluno no centro do processo de aprendizado em que para conseguir atingir seu objetivo deve aplicar seus conhecimentos existentes ou buscar novos por meio de orientação do professor, livros, ou conteúdo digital.

De forma geral as atividades em escolas com robótica educacional são realizadas com o suporte de conjuntos para robótica. Esses conjuntos ou kits são constituídos por uma interface eletrônica responsável pela comunicação com o computador, e para ativação de atuadores como motores, lâmpadas e relês, aquisição e conversão de grandezas físicas obtidas de sensores diversos que captam grandezas como temperatura e toque, e um sistema para programação que possibilita descrever na forma de uma linguagem de programação o comportamento planejado para o robô.

O aspecto didático acompanhado de uma metodologia adequada é um fator determinante para a implementação com sucesso da robótica em um contexto educacional. O aspecto tecnológico que pode ser representado por interfaces ou sensores dotados de recursos e capacidades elaboradas não fará sentido do ponto de vista educacional se não puderem ser usados pelo aluno na construção de projetos que resultem em objetos que façam sentido para ele próprio e a comunidade. O foco apenas no aspecto tecnológico pode levar a frustração em razão de muitas vezes ser necessário aplicar um esforço excessivo no domínio de como funcionam sensores, motores, comandos, reforçando para professores e alunos o senso comum de que robótica é algo difícil e acessível apenas a profissionais especializados.

Empresas privadas como a Lego tem oferecido conjuntos para robótica que podem ter aplicação educacional. O kit Lego NXT (2012) desenvolvido inicialmente com o propósito de entretenimento tem sido utilizado em atividades de robótica educacional com alunos do ensino fundamental, médio e superior. Constituído por blocos de plástico com encaixe e um computador programável permite desenvolver os mais diversos tipos de robô.

A comunidade acadêmica tem pesquisado e desenvolvido conjuntos e sistemas para robótica educacional [Sipitakiat et al. 2002] [Chella 2002], em sua maioria tendo como referencial pedagógico e metodológico o construcionismo.

Sistemas com desenvolvimento não específico para robótica educacional como o Arduíno (2012), uma plataforma para computação física, têm sido adaptados para aplicações com robótica educacional.

Além da montagem do dispositivo robótico o aluno deve criar um programa computacional que seja capaz de ler dados provenientes de sensores, acionar atuadores, dentro de uma sequência lógica que corresponde ao objetivo proposto. Algumas ferramentas computacionais foram desenvolvidas especificamente para aplicação educacional quer ela seja em ambientes formais ou não. A linguagem Logo [Valente 1996] foi uma das primeiras a ser desenvolvida com um propósito educacional. Ao

adotar a metáfora de uma tartaruga que se move em um plano a partir de comandos simples que indicam a direção e a quantidade de “passos”, foi inovadora ao oferecer uma sintaxe clara e objetiva possibilitando manter o foco no projeto e criatividade. A linguagem Scratch(2012) desenvolvida a partir dos conceitos do Logo é uma evolução significativa ao oferecer um sistema de programação baseado em comandos representados por blocos gráficos. A implementação de um programa se dá pela conexão dos diversos blocos. Essas ferramentas computacionais não foram desenvolvidas especificamente para programação de robôs, contudo trabalhos de pesquisa [Chella 2010] [Rusk et al. 2008] ampliaram suas funcionalidades por meio de bibliotecas ou *plugins* para que pudessem se comunicar com interfaces de hardware.

A construção de um sistema materializado na forma de um objeto físico constituído por motores, uma estrutura mecânica e sensores coordenados, controlados por um programa computacional exige do aluno uma forma de pensar semelhante a de um matemático ou cientista da computação, de um engenheiro ou de um cientista da computação. O cientista da computação utiliza uma linguagem formal para expressar idéias (especificamente algoritmos). Engenheiros projetam, associam componentes estruturados na forma de sistemas, avaliam os resultados propondo alternativas. Cientistas observam o comportamento de sistemas complexos, formam hipóteses, e projetam experimentos para comprovar as hipóteses.

Expor alunos a essas formas de pensar pode auxiliar os mesmos na condução das atividades do dia a dia e despertar vocações para áreas como ciência ou engenharia.

As seções seguintes descrevem alguns simuladores específicos para robótica, o desenvolvimento do simulador para robótica SimRobô, avaliação do sistema em ambiente com problema tradicional em robótica e as considerações finais sobre o trabalho.

2. Simuladores aplicados à robótica

O desenvolvimento de sistemas robóticos tem se tornado complexo a medida que o mercado e a academia disponibiliza um número crescente de componentes de hardware e software. Ocorre também um incremento na variedade e complexidade das tarefas planejadas para os robôs. Tomando como exemplo a OBR (Olimpíada Brasileira de Robótica) propõe uma prova que reproduz o resgate de vítimas em um ambiente a priori desconhecido. Para atender essa tarefa o robô deve ser capaz de navegar no ambiente, identificar e recolher objetos definidos como vítimas e evitar obstáculos. Essa prova é aplicada para alunos do ensino fundamental e médio. O desenvolvimento e validação de um robô que atenda de forma satisfatória esse tipo de tarefa, de forma geral consome tempo e recursos. O uso de um simulador pode facilitar o processo de desenvolvimento e validação do robô, permitindo verificar a integração dos componentes, e avaliar o comportamento nas mais diversas condições como por exemplo a mudança no ambiente.

Outro aspecto da simulação é que a mesma é relativamente menos custosa em termos de recursos físicos e financeiros, oferecendo maior flexibilidade ao permitir explorar condições que de outra forma seriam inviáveis: um ambiente simulado pode ser mais complexo e ocupar um espaço físico maior que o disponível no laboratório. Outra possibilidade é a de simular múltiplos robôs mesmo sem a disponibilidade de hardware.

Simuladores para robótica podem ser classificados em dois grupos: os que são restritos para validar determinado componente como o processamento de dados de um tipo específico de sensor ou um sistema cinemático em particular [Simeon et. al 2001] [Kanehiro et. al 2002] e os simuladores que possibilitam a avaliação de robôs a nível de sistema, com flexibilidade para alterar e integrar componentes de software e hardware.

Nessa segunda categoria se enquadra a proposta do simulador SimRobô descrito nesse trabalho. Alguns simuladores que seguem esse modelo são descritos a seguir:

O aplicativo *Webots*(2012) é um ambiente de simulação com aplicação na modelagem, programação e simulação de robôs móveis. Os modelos simulados podem ser configurados com atributos como textura, massa, forma e coeficiente de fricção. O sistema disponibiliza várias plataformas já configuradas como o Lego NXT e Pioneer (2012) e módulos com os sensores e atuadores mais comuns. Distribuído comercialmente oferece versões com preços que podem chegar a 2.300,00 dólares.

O simulador *Easy-Rob*(2012) é um sistema distribuído comercialmente aplicado a planejamento e simulação de plataformas robóticas em plantas industriais que operam com células de trabalho. Permite programar e visualizar em 3D processos como manipulação de objeto, montagem e embalagem. Um conjunto de API's (application programming interface) habilita a integração com aplicações externas.

Projeto de código fonte aberto *Stage*(2012) é um sistema para simulação com visualização em 2D com recursos para programação de robôs em ambientes multiagentes, oferece alguns modelos de plataformas robóticas e um conjunto de sensores genéricos. Provê recursos para detecção de colisão e geração de mapas.

O *toolkit* CARMEN (2012) é constituído por uma coleção de software com aplicação no controle de robôs móveis. As simulações são visualizadas em um ambiente 2D e programadas nas linguagens C e Java no sistema operacional Linux. Oferece suporte a atuadores e sensores mais comuns como os de proximidade, GPS e colisão.

Os sistemas de software avaliados são representativos permitindo sintetizar algumas conclusões: aplicativos comerciais oferecem suporte e documentação adequados porém o custo pode ser proibitivo para inserção nos ambientes educacionais, as soluções livres e de código aberto nem sempre oferecem um suporte e documentação adequados, e para instalação e configuração demandam domínio de compilação do código fonte em compiladores, geralmente C/C++.

Tendo como referencia esse cenário considera-se justificado o desenvolvimento de um simulador com aplicação em ambiente educacional, flexível e extensível com facilidade para programação de robôs, configuração e instalação.

3. Arquitetura do SimRobô

Ao planejar o desenvolvimento do simulador SimRobô alguns critérios forma definidos, entre eles pode-se destacar facilidade de instalação e configuração, independência de plataforma de hardware e software, e flexibilidade para se modificar robôs, sensores, ambientes, operação em tempo real e utilização de software livre.

Para atender a independência de hardware foi adotada como ferramenta de desenvolvimento o Python, associado ao motor para desenvolvimento de jogos 2D Pygame (2012), ambas portáteis possibilitando execução em plataformas que incluem

computadores PC baseados em processadores x86 e dispositivos móveis como *smartphones* e *tablets* baseados processadores de tecnologia ARM.

Seguindo a tendência das arquiteturas de simuladores para robótica e outras aplicações nas quais plataforma de hardware e software heterogêneas devem ser capazes de trocar informações, o SimRobô implementa um sistema sob o conceito de *middleware*. Por meio do *middleware* ocorre a comunicação da aplicação do usuário com mecanismo do simulador, objetos gráficos, modelos de sensores, atuadores, objetos do cenário, detector de colisão, entre outros sistemas.

A comunicação entre *middleware* e aplicação do usuário se dá pela troca de mensagens transportadas por *socket* em um modelo cliente-servidor. No lado cliente foram desenvolvidas API's nas linguagens Python e Java, possibilitando ao usuário criar aplicações de controle utilizando chamadas a API. A comunicação por meio de *socket* permite que qualquer ferramenta de desenvolvimento com suporte a comunicação em rede possa ser utilizada para criar aplicações de controle para o SimRobô. O protocolo de mensagem baseado em texto facilita o desenvolvimento e extensão de API's.

O diagrama em blocos da figura 1 apresenta os vários sistemas que compõe o Simrobô e a interconexão entre eles.

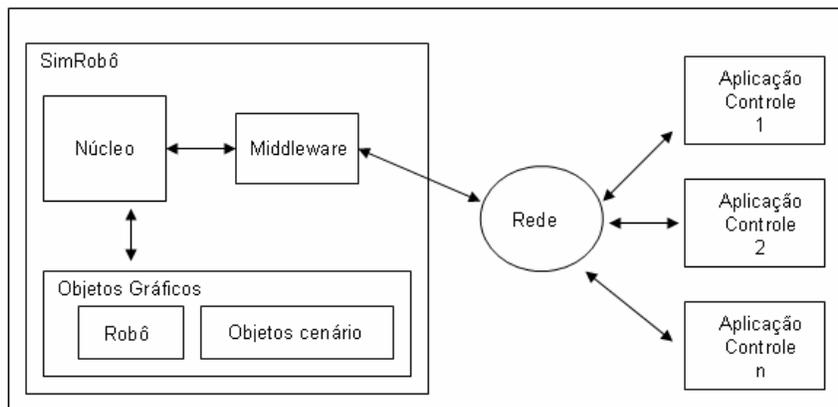


Figura 1. Diagrama sistema SimRobô

O *Middleware* é responsável pela comunicação em rede, gerenciamento das conexões, detecção de erros, recebimento e despacho de mensagens. O Núcleo atua no gerenciamento, decodificação e codificação de mensagens, controle e animação dos objetos gráficos, processamento dos módulos de sensor e atuadores, e tratamento de exceções de execução.

A aplicação do usuário envia comandos contidos nas API's para controle dos atuadores e leitura dos dados de sensores.

As API's oferecem as primitivas básicas para leitura de sensores e acionamento dos motores, com esses recursos o desenvolvedor de robô pode desenvolver algoritmos complexos que utilizam por exemplo inteligência artificial, mapeamento, entre outros.

O SimRobô tem modelado um robô móvel constituído por um sistema de tração diferencial com dois motores opostos, um sensor de distância posicionado a frente do

robô e dois sensores de contato para detectar a colisão física com obstáculos, um posicionado à frente e outro na traseira (figura 2).

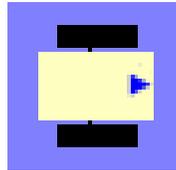


Figura 2. Representação robô móvel

O sistema foi implementado em classes contendo as funcionalidades de sensores e atuadores de forma que o usuário pode expandir o simulador programando ou modificando classes existentes adaptadas a determinado tipo de componente de hardware.

4. Execução de simulações

Para demonstrar a funcionalidade do SimRobô foi criado um ambiente de execução semelhante a um prédio com, corredores, passagens e obstáculos (figura 3).

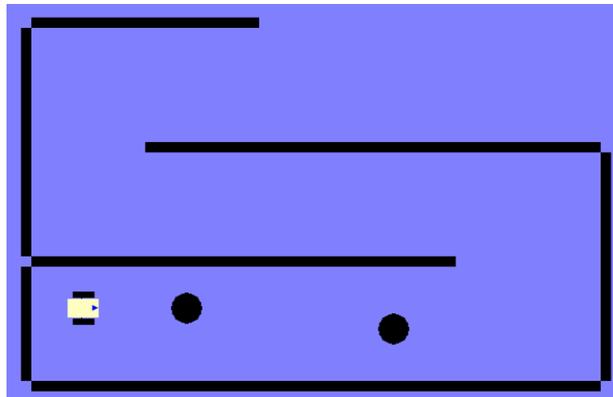


Figura 3. Ambiente de execução da simulação

Um programa de controle foi desenvolvido com o propósito de que o robô navegue no ambiente, desvie de obstáculos utilizando os sensores de distância e contato

O desenvolvimento do código de controle exemplo foi baseado na abordagem de robótica reativa e comportamental [Brooks 1986] em que regras simples são estabelecidas para que o robô navegue no ambiente, evitando obstáculos e em caso de colisão o mesmo executa movimento no sentido oposto ao do obstáculo. A tabela 1 exibe trecho do código desenvolvido utilizando a API Python.

```

while True:
    distancia = float(robo.leranaloga(1))
    colisao_parede = robo.lerdigital()
    if (distancia < 70):
        robo.diferencial(4)
        time.sleep(0.1)
        robo.diferencial(1)
        time.sleep(0.1)
    if (colisao_parede == 2):
        print "menor 130"
        robo.diferencial(3)
        time.sleep(0.1)
        robo.diferencial(1)
        time.sleep(0.1)

```

Tabela 1. Ambiente de execução da simulação

A função *leranalógica(x)* é genérica para leitura de dados de sensores com saída linear e nesse exemplo aplicada para ler a distância dentro de um limite, ao implementar um robô real esse sensor poderia ser baseado em ultrassom ou infravermelho. Para executar os movimentos a função *diferencial(x)* de acordo com o parâmetro *x* executa giros no eixo do robô ou deslocamento para frente ou sentido contrário.

Caso ocorra uma colisão o valor do sensor de contato é atualizado e ao chamar a função *lerdigital()* se obtém valores que indicam o estado desse sensor. Um robô real para sensor de colisão de forma geral utiliza chaves que se fecham ao contato, ou sensores óticos. A execução da simulação ocorre em tempo real e recursos como a exibição do rastro do robô a atuação dos sensores facilita a depuração e avaliação do algoritmo em execução. A figura 4 exibe o resultado da execução do código da tabela 1.

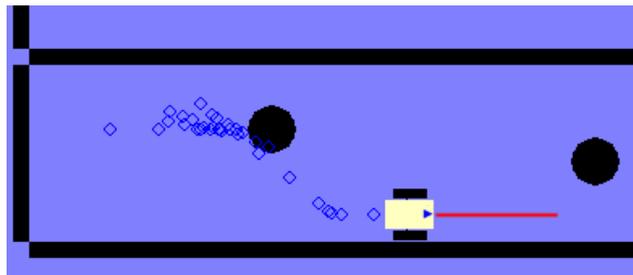


Figura 4. Simulação em execução

5. Considerações Finais

O desenvolvimento de um robô envolve as etapas de definição de projeto, construção física, montagem e integração dos elementos eletrônicos como sensores e motores, desenvolvimento do sistema de controle utilizando uma ferramenta de programação, depuração e avaliação dos resultados. Esse conjunto de etapas pode ter um custo financeiro e humano significativo o que de certa forma pode limitar a implantação da robótica como ferramenta educacional tanto nos níveis de ensino básicos como no nível superior. Ferramentas para simulação têm sido consideradas como alternativa e complemento para viabilizar o contato de alunos com a robótica. No caso de instituições que não possuem robôs físicos a simulação pode oferecer a oportunidade de colocar

alunos em contato com vários aspectos relacionados à programação e controle de um robô. Já as instituições de ensino e pesquisa que possuem robôs físicos podem utilizar a simulação em etapa inicial para testar algoritmos e técnica com maior rapidez e menor custo, após a validação o robô pode ser construído e controlado com comportamento parecido ao obtido na simulação.

O SimRobô tem como propósito oferecer a comunidade educacional um simulador para robôs móveis de configuração e instalação simplificada, flexível e com estrutura modular possibilitando a expansão de seus recursos com facilidade. Um protocolo de comunicação baseado em mensagens de texto permite a implementação de APIs na mais diversas linguagens podendo se adaptar aos mais diversos contextos educacionais.

A evolução do desenvolvimento planeja desenvolver ambientes para programação baseado em ferramentas como Scratch e dialetos do Logo podendo assim atingir alunos do ensino fundamental que ao adquirir contato com a robótica estarão habilitados a participar de atividades como a OBR e atuarem como criadores e não apenas consumidores de tecnologia.

Referências

- Arduino (2012) “Plataforma Arduino”, Disponível em: < <http://www.arduino.cc/>>
Acesso em: 13/07/2012.
- Brooks, R.A., (1986) “A robust layered control system for a mobile robot” In: IEEE Journal of Robotics and Automation. vol.2, issue 1.
- CARMEN (2012) “Carnegie Mellon Robot Navigation Toolkit”, Disponível em: < <http://carmen.sourceforge.net>> Acesso em: 13/07/2012.
- Chella, M. T. (2002) “Ambiente de Robótica Educacional com Logo” In: XXII Congresso da Sociedade Brasileira de Computação - SBC2002, Florianópolis.
- Chella, M. T. (2010) “Robotic Tool with Scratch Language” In: Robocontrol 2010 4th Workshop in Applied Robotics and Automation, UNESP., Bauru.
- D’Abreu, J.V.V. (1993) “Uso do computador para controle de dispositivos: O traçador gráfico (Plotter)”, In: Computadores e Conhecimento. UNICAMP/NIED,. Cap. 17.: p. 350-366., Campinas.
- Easy-Rob (2012) “Easy-Rob 3D Robot Simulation Tool”, Disponível em: < <http://www.easy-rob.com/>> Acesso em: 13/07/2012.
- Kanehiro, F., Fujiwara, K., Kajita, S., Yokoi, K., Kaneko, K., Hirukawa.H. (2002) "Open architecture humanoid robotics platform" In Proceedings of the IEEE International Conference on Robotics and Automation
- Lego (2012) “Lego.com Mindstorms”, Disponível em: <<http://mindstorms.lego.com/>>
Acesso em: 13/07/2012.
- Papert, S. (1994), “A máquina das Crianças Repensando a Escola na Era da Informática”. Artes Médicas, Porto Alegre.
- Papert, S. (1985) “LOGO: Computadores e Educação”. Editora Brasiliense, São Paulo.

- Piaget, J. (1952) "The Origins of Intelligence in Children", International Universities Press, New York.
- Pygame (2012) "Python game development", Disponível em: <<http://www.pygame.org>> Acesso em: 13/07/2012.
- Pioneer (2012) "Pioneer 3-AT", Disponível em: <<http://www.mobilerobots.com/ResearchRobots/P3AT.aspx/>> Acesso em: 13/07/2012.
- Rusk, N., Resnick, M., Berg, R., Granlund, M.P. (2008) "New Pathways into Robotics: Strategies for Broadening Participation" In: Journal Of Science Education And Technology, Volume 17, Number 1 (2008), 59-69
- Scratch (2012) "Scratch", Disponível em: <<http://scratch.mit.edu/>> Acesso em: 13/07/2012.
- Sipitakiat, A., Blikstein, P., Cavallo, D.P. (2002) "The GoGo Board: Moving towards highly available computational tools in learning environments.". In: Interactive Computer Aided Learning International Workshop. Carinthia Technology Institute, Villach, Austria.
- Simeon, T., Laumond, P., Lamiroux., F. (2001) "Move3d: a generic platform for path planning" In in 4th Int. Symp. on Assembly and Task Planning.Fukuoka, Japan.
- Stage (2012) "2D multiple-robot simulator", Disponível em: <<http://playerstage.sourceforge.net/>> Acesso em: 13/07/2012.
- Valente, J.A. (1996) "O Professor no Ambiente Logo: Formação e Atuação", Núcleo de Informática Aplicada à Educação/UNICAMP, Campinas.
- Webots (2012) "Webots: robot simulator", Disponível em: <<http://www.cyberbotics.com/>> Acesso em: 13/07/2012.