

AMBIENTE VIRTUAL 3D PARA AUTOMAÇÃO DA MANUFATURA

HENRIQUE B. LISBOA¹, LUIZ A. R. O. SANTOS¹, EDER R. MIYASHIRO¹, KAREN J. SUGAWARA¹, PAULO E. MIYAGI¹,
FABRÍCIO JUNQUEIRA¹

1. *University of São Paulo, Brazil*

E-mails: henriquelisboa@outlook.com, luizarthur.santos@gmail.com, edermiyashiro@gmail.com, karen.sug@gmail.com, pemiyagi@usp.br, fabri@usp.br

Abstract— The increase in the use of automation and robotics in the industries, aiming to increase productivity and lower costs, makes important the development of tools to mitigate the difficulties found in this new reality and optimize the already existing processes. Using virtual reality concepts, virtual manufacturing software allows 3D graphical visualizations of a product or a manufacturing process, to identify problems in the sequence of commands of industrial equipments by means of simulations. The aim of this work is to design and implement a virtual manufacturing environment more immersive, intuitive and interactive, providing an experience as close as possible to the real. For this, it was used a virtual manufacturing environment consisting of an industrial robot (Kuka KR15/2), and the conventional input devices (mouse, keyboard or joystick) were partially replaced by Microsoft Kinect. It was used the arm movement of the operator to control the robot and the camera, and voice commands to activate other auxiliary functions. As a result, this virtual manufacturing environment can be used in engineering courses as well as specialized employee training without the need for supervision of use or disrupt the actual machine on the shop floor or in the teaching unit.

Keywords— Virtual reality, Manufacture, Industrial robots

Resumo— O aumento do uso de automação e robótica nas indústrias, com o objetivo de aumentar a produtividade e reduzir os custos, torna importante o desenvolvimento de ferramentas para mitigar as dificuldades encontradas nesta nova realidade e otimizar os processos já existentes. Usando os conceitos de realidade virtual, *softwares* de manufatura virtual permitem visualizações gráficas em 3D de um produto ou processo de fabricação para, por meio de simulação, identificar problemas na sequência de comandos de equipamentos industriais. O objetivo deste trabalho é projetar e implementar um ambiente de manufatura virtual mais imersivo, intuitivo e interativo, proporcionando uma experiência o mais próxima possível da real. Para isso, utilizou-se um ambiente de fabricação virtual consistindo de um robô industrial (Kuka KR15/2), e os dispositivos de entrada convencionais (*mouse*, teclado ou *joystick*) foram parcialmente substituídos pelo Microsoft Kinect. Foi utilizado o movimento do braço do operador para controlar o robô e a câmera, e comandos de voz para ativar outras funções auxiliares. Como resultado, este ambiente virtual de manufatura pode ser usado em cursos de engenharia, bem como treinamento de funcionários especializados, sem a necessidade de supervisão do uso ou interromper a máquina real no chão de fábrica ou na unidade de ensino.

Palavras-chave— Realidade virtual, Manufatura, Robôs industriais

1 Introdução

A Realidade Virtual (RV) é, antes de tudo, uma “interface avançada do usuário” para acessar aplicações executadas no computador, tendo como características a visualização de ambientes tridimensionais, a movimentação nestes ambientes em tempo real, e a interação com elementos desse ambiente (Tori; Kirner, 2006). A RV proporciona uma nova forma para representar a interação homem-máquina e explorar a relação existente entre estes no mundo real, bem como interações entre usuários (Li e Sun, 2009). Utilizando diversos sentidos do homem, como a visão, audição, tato, etc. para a percepção, bem como gestos, posturas, sinais, falas, para interação com os equipamentos, um ambiente virtual pode produzir efeitos de imersão.

Dentre todas as suas áreas, a realidade virtual não imersiva (com uso de monitores convencionais) é a mais utilizada, devido ao seu baixo custo. Para tornar a experiência mais eficaz e aproveitável, buscam-se novas técnicas de modo a tornar a mesma mais interativa (maior influência nos objetos do meio) e intuitiva (facilidade de interação). Sendo assim, a

tendência da indústria de RV é a de eliminar ou restringir o uso de dispositivos de entradas mais convencionais como *mouse*, teclado e *joystick* (Gutiérrez; Vexo; Thalmann, 2008).

A utilização da RV tem se expandido gradativamente nas mais variadas áreas do conhecimento, indo de aplicações de entretenimento até aplicações de tempo real. Isto se deve a muitos fatores, como os avanços das pesquisas, o domínio e acesso das pessoas a recursos computacionais decorrente do barateamento dos equipamentos de informática, seus componentes e interfaces (Curzel; Hounsell; LeaL, 2007).

Na engenharia, de acordo com Mujber; Szecsi; Hashmi (2004) os avanços da RV nos últimos anos proporcionaram um impulso para a sua aplicação em diferentes atividades, tais como, projeto de produto, modelagem, controle de chão de fábrica, simulação de processos, planejamento de produção, treinamentos, testes e verificação. Ela ajuda a reduzir o tempo de desenvolvimento do produto e a aumentar a qualidade da produção (Kim *et al.*, 2006). Ela tem um grande potencial para resolver problemas em aplicações industriais, ainda na fase de projeto, evitando assim, custos indesejáveis durante a implantação (Mujber; Szecsi; Hashmi, 2004). Os projetistas podem controlar as máquinas virtuais dentro da fábrica

virtual e avaliar o produto digital com antecedência. Procura-se, assim, eliminar erros inesperados, bem como reduzir os custos e tempos de processo. Ela também permite a criação de protótipos realísticos, muito mais baratos e rápidos de serem elaborados que os protótipos físicos, os quais podem ser aplicados durante a fase de desenvolvimento de um processo de produção. Nas áreas automotiva e naval, o uso conjunto da RV com sistemas virtuais de manufatura tem promovido o aumento da competitividade. Segundo Kadir *et al.* (2011), nesse ambiente pode-se estudar melhor as condições de segurança. O gerenciamento de dados da fabricação e do ciclo de vida do produto podem ser facilmente documentados, de forma mais sistemática.

Sua aplicação na área da manufatura é recente. Pode-se controlar e verificar os mais diversos processos de fabricação, de modo virtual, reduzindo o custo e tempo necessários quando comparado às tecnologias tradicionalmente utilizadas (Banerjee; Zetu, 2001).

Em decorrência dos benefícios da RV para a área de manufatura, inúmeros sistemas virtuais têm sido desenvolvidos, dentre eles (Kadir *et al.*, 2011): manufatura virtual (*Virtual Manufacturing* - VMfg), máquina ferramenta virtual (*Virtual Machine Tool* - VMT), usinagem virtual (*Virtual Machining* - VMach), montagem virtual (*Virtual Assembly* - VA), ferramentaria virtual (*Virtual Tooling* - VTo), e prototipagem virtual (*Virtual Prototyping* - VP).

No Brasil, ainda poucas empresas utilizam a RV no estudo e desenvolvimento de produtos. Entre elas, pode-se citar: Petrobrás¹², Embraer³, e Vale⁴. Em comum, são grandes empresas com amplos recursos para investir maciçamente neste tipo de tecnologia.

Além disso, sistemas virtuais de manufatura também podem ser utilizados em ambientes de aprendizado (Cheng; Wang, 2011). Potkonjak *et al.* (2010) apontam o crescimento do ensino à distância e do *e-learning* nas últimas duas décadas em decorrência do crescimento da internet e das melhorias nas tecnologias de comunicação. Eles destacam em especial os cursos de engenharias elétrica e mecânica, que necessitam de atividades de laboratório realizadas com o auxílio de equipamentos físicos, reais. Neste sentido, para permitir uma experiência plena de *e-learning* nesta área de conhecimento, Potkonjak *et al.* destacam a RV como uma ferramenta útil na criação de laboratórios virtuais que permitam o ensino à distância, onde modelos matemáticos podem ser utilizadas para simular os principais comportamentos dinâmicos das entidades reais. No caso específico de um laboratório de robótica, uma entidade que utiliza tanto conceitos de mecânica como de eletrônica, além

do robô propriamente dito, é necessário definir o ambiente onde este está inserido. No ambiente virtual, o robô é substituído por um modelo gráfico e um modelo matemático que simula a sua dinâmica. Quanto ao ambiente, este pode ser mais ou menos complexo, podendo possuir apenas elementos estáticos, ou possuir elementos que têm movimentos associados ao comportamento do robô. O ambiente também pode envolver outros robôs, máquinas de comando numérico (CNC), transportadores (VATs – veículos autônomos de transporte), etc. Potkonjak *et al.* apontam ainda as vantagens da utilização de ambientes virtuais no aprendizado.

O objetivo deste trabalho é projetar e implementar um ambiente virtual de manufatura, mais imersivo, intuitivo, interativo, e de baixo custo, proporcionando uma experiência o mais próxima possível da real.

Este trabalho está organizado da seguinte forma: a seção 2 apresenta materiais e métodos adotados para este trabalho, a seção 3 apresenta um exemplo de aplicação, e na seção 4 são apresentadas as conclusões.

2 Materiais e Métodos

2.1 Hardware

Para o desenvolvimento e execução do ambiente virtual foi utilizado um computador contendo: 1 Placa mãe Intel DX58OG; 1 Processador Intel Core i7 950 3.06GHz; 8Gb de memória DDR3 - PC1333; 1 HD SATA II 1Tb, 7200RPM; 1 Placa de vídeo GeForce GTX 580 – 1,5Gb – DDR3 – PCI-Xpress; 1 Kit Óculos Nvidia 3D Vision c/ transmissor de dados; 1 Monitor LED BenQ XL2410T Full HD 23,6” 3D Ready; Teclado e Mouse óptico; e 1 Microsoft Kinect. Esta configuração tem custo aproximado de R\$7.000,00, o que o torna acessível a pequenas e médias empresas.

2.2 Softwares

Para este projeto foram utilizados os *softwares*: Windows 7 Professional 64bits; Visual Studio 2010; XNA Game Studio 4; Kinect SDK 1.5; e Blender. Com exceção do Windows, os demais podem ser obtidos gratuitamente.

O Visual Studio 2010 foi escolhido por permitir a integração das demais tecnologias utilizadas, ou seja, o XNA Game Studio 4 e o Kinect SDK 1.5. Mais ainda, ele permite uma integração natural com equipamentos de automação industrial por meio do OPC (*OLE for Process Control*). A especificação OPC é baseada em OLE (*Object Linking and Embedding*), COM (*Component Object Model*) e DCOM (*Distributed COM*), tecnologias desenvolvidas pela Microsoft para a família do sistema operacional Windows. Esta especificação define um conjunto padrão de objetos, interfaces e métodos para uso em aplicações de automação de manufatura para

¹ revistagalileu.globo.com/Galileu/0,6993,ECT803835-1938-1,00.html

² www.revistapetrobras.com.br/revistapetrobras/files/pdfs/Materia%20de%20Capa.pdf

³ www.embraer.com/pt-BR/ConhecaEmbraer/QualidadeTecnologia/Paginas/Home.aspx

⁴ www.usinagem-brasil.com.br/3531-poli-usp-desenvolve-simulador-de-trem-para-a-vale/

facilitar a interoperabilidade de controle e de processo. O objetivo é reduzir a quantidade de esforço duplicado exigido dos fabricantes de hardware e seus parceiros de software, e de desenvolvedores de sistemas SCADA (*Supervisory Control And Data Acquisition*) e HMI (*Human-Machine Interface*), a fim de promover a interação entre eles.⁵⁶

Para o gerenciamento do ambiente 3D, foi adotado o XNA. Sua escolha se deve por ser, segundo Pathak e Joshi (2009), um excelente *framework* para o desenvolvimento de aplicações 3D. Ele é baseado no Microsoft .NET *Framework*. Ele suporta codificação em C# (C-Sharp) e os ambientes de desenvolvimento Visual Studio 2010.

A escolha do Blender⁷ se deve ao fato deste ser distribuído gratuitamente (Amici *et al.*, 2010). Mesmo assim, deve-se destacar que existem outras diversas ferramentas para a criação de modelos 3D no contexto da RV, como Blender, 3D Studio Max⁸, Maya⁹ e Rhino¹⁰ (Paneels *et al.*; 2010).

2.3 Método

Para o projeto e implementação do ambiente de manufatura virtual seguiu-se as etapas:

- Definir os elementos que devem compor o ambiente virtual de manufatura.
- Modelar graficamente, em 3D, cada um dos elementos, em separado, utilizando o Blender.
- Exportar os elementos no formato FBX¹¹.
- Importar os elementos gráficos para o projeto em XNA. Por adotarem (Blender e XNA) sistemas de coordenadas diferentes, cada elemento deve ser rotacionado 90° em torno de X e 180° em torno de Z.
- Modelar cinemática e dinamicamente cada elemento, incluindo o *bounding box* para a detecção de colisão, bem como outras funções e métodos específicos de cada elemento.
- Associar os elementos às entradas, como, por exemplo, teclado, *mouse*, *joystick*, e o próprio Kinect (gestos e/ou comandos de voz).

3 Exemplo de Aplicação

Como exemplo de aplicação foi utilizado um robô Kuka KR15/2 utilizado nas aulas de robótica do curso de engenharia mecatrônica da Escola Politécnica da USP. Foi projetado e implementado um modelo virtual do ambiente didático, incluindo-se o robô citado.

Para interação com o ambiente, além do *mouse* e do teclado, são utilizados os movimentos do punho

direito do usuário para movimentar o efetuator do robô e a câmera. Os comandos de voz são utilizados em caráter secundário, de modo a auxiliar a utilização das funções implementadas, como a reprodução e gravação de movimento.

As principais funcionalidades do *software* são: detecção do movimento, movimentação do robô, cálculo de cinemática inversa, detecção de colisão e gravação e interpretação de código de programação do robô.

A seguir são apresentados alguns dos principais aspectos da modelagem gráfica e implementação do ambiente de manufatura virtual escolhido.

3.1 Modelagem gráfica utilizando Blender

A modelagem em Blender pode e deve ser feita de duas formas: no “modo objeto” e no “modo edição”. No “modo objeto” (Fig. 1.a), para garantir a compatibilidade com o XNA, os elementos devem ser posicionados na origem do sistema de coordenadas, ou seja $(x, y, z) = (0, 0, 0)$, não devem ser feitas rotações, e deve-se utilizar a escala 1.0. Já no “modo edição” (Fig. 1.b), os vértices, arestas, faces, ou mesmo o elemento como um todo podem ser deformados (transladados, rotacionados ou ter suas dimensões alteradas) livremente.

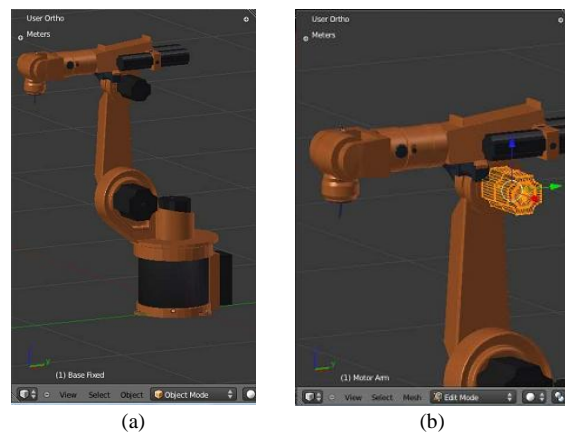


Figura 1. (a) Modelagem em (a) modo objeto e (b) modo edição no Blender.

Para permitir a movimentação do elemento no XNA, é necessário que seja criada uma armadura que envolve o modelo e, a cada parte móvel seja associada a um osso (Fig. 2.a). A armadura é similar a um esqueleto, formada por ossos ligados entre si que, ao se movimentarem, movimentam e deformam as partes associadas a eles (Gumster, 2009).

Em seguida, cada modelo deve ser exportado no formato FBX, conforme demonstrado na Fig. 2.b.

3.2 Detecção de movimento do punho do usuário

Para a reconstrução do esqueleto humano, há estudos (Straka *et al.*, 2011) que fazem a reconstrução usando algoritmos customizados e que conseguem delimitar

⁵ www.opcfoundation.org/

⁶ en.wikipedia.org/wiki/OLE_for_process_control

⁷ www.blender.org

⁸ www.autodesk.com/3dsmax

⁹ www.autodesk.com/maya

¹⁰ www.rhino3d.com

¹¹ en.wikipedia.org/wiki/FBX

os erros do resultado em diversas situações dinâmicas. Em geral, para aplicações de rastreamento, os dados devem ser adquiridos entre 1 e 3 metros do sensor.

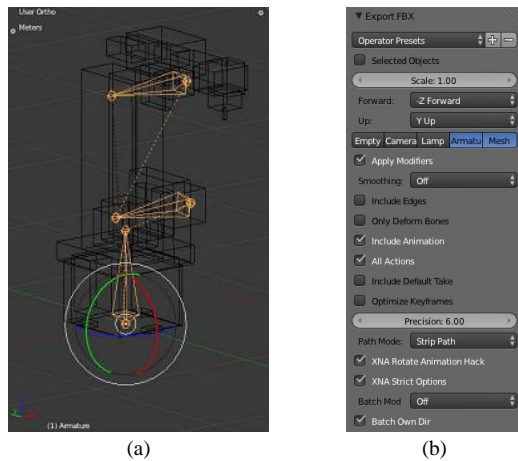


Figura 2. (a) Armadura e ossos no Blender; e (b) Exportando o modelo no formato FBX para ser usado no XNA.

Para determinar a precisão na detecção do movimento optou-se por desenhar em tempo real um círculo na posição onde se localiza a junta da mão direita e escrever na tela a profundidade detectada. Dos dados enviados pelo sensor, a profundidade fornecida da junta observada é dada em metros. Os valores máximos e mínimos de x e y dependem da distância do usuário à câmera. Eles podem variar aproximadamente entre -0,8 e 0,8, próximo da câmera (1m), até -1,2 e 1,2 longe da mesma (4m) (Fig. 3). A origem do sistema de coordenadas é aproximadamente o centro da imagem obtida. O valor de x aumenta para a direita do usuário (esquerda do Kinect) e y aumenta para cima. O Kinect compara o plano da imagem obtida com um plano de referência, determinando, desta forma, a profundidade das juntas.

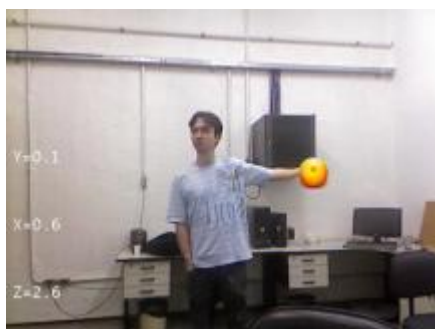


Figura 3. Resultado obtido na detecção da junta da mão direita (a imagem está espelhada).

Em decorrência das oscilações na posição das juntas (fornecidas pelo Kinect) observadas ao longo do projeto, e observadas também em (Gonzalez-Jorge *et al.*, 2013), foi necessário implementar um filtro. Neste caso, adotou-se um filtro por histerese.

3.3 Movimentação das juntas do robô e sua cinemática inversa

O equacionamento do robô foi baseado em Siciliano *et al.* (2010) e será omitido por não ser o foco do trabalho. No entanto, destaca-se que o cálculo da cinemática inversa, com base na posição da junta da mão direita do usuário necessita de atenção especial quando esta é posicionada fora do volume de trabalho do robô. Esta consideração é importante para evitar singularidade de fronteira. Para evitar este problema, a cada incremento no deslocamento, verificase se a nova posição está contida no volume de trabalho, para depois calcular a cinemática inversa. Ao posicionar a junta da mão fora do volume de trabalho, o robô se movimenta o mais próximo possível da mesma (Fig. 4.a).

Outro problema surge com o robô já no limite do volume de trabalho (Fig. 4.b) e o usuário movimentando o punho para uma outra posição também fora do volume de trabalho. Nessa situação, como o caminho a ser percorrido pelo robô não possui solução, o mesmo ficaria travado. Para solução desse problema, implementou-se uma função que, a partir de um vetor localizado no centro do sistema de coordenadas, caso a posição desejada seja fora do volume de trabalho, obtêm-se o ponto nos limites do volume mais próximo do desejado, na intersecção dessa reta com a circunferência que delimita o volume de trabalho (Fig. 4.c).

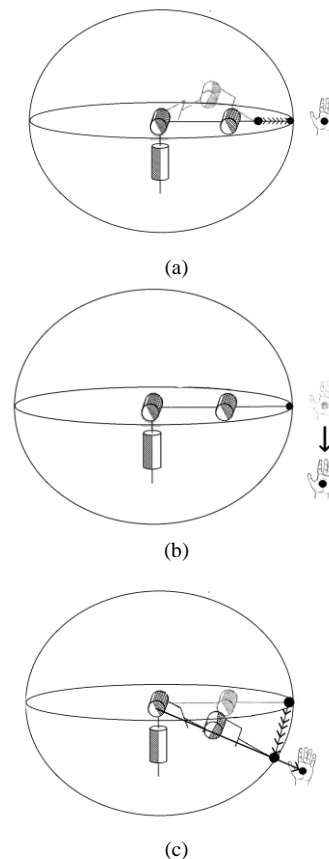


Figura 4. (a) Movimentação do robô até o limite do volume de trabalho; (b) Movimentação partindo de posição no limite do volume de trabalho; e (c) Movimentação linear do robô limitando-se ao volume de trabalho.

3.4 Geração e interpretação do código de programação do robô

Decidiu-se armazenar os dados de movimentação do robô virtual de modo a poder controlar o robô real por meio dos movimentos do corpo, aplicação futura deste projeto. Para tal utilizou-se a própria linguagem de programação KRL (Kuka Robot, 2010), criada pelo fabricante Kuka, para armazenar esses dados que posteriormente podem ser lidos pelo robô. Utilizando-se o ANTLR¹² (*Another Tool for Language Recognition*) e uma gramática¹³ baseada na linguagem KRL para gerar duas classes, a KRLParser e a KRLLexer, para auxiliar na interpretação do código KRL.

Implementaram-se três comandos aceitos para deslocamento do efetuador do robô: LIN, PTP e LIN CONT. LIN é o comando de movimentação Linear. Nele o robô se movimenta de modo contínuo através de pontos com pequenos incrementos entre si que determinam uma trajetória reta entre os pontos iniciais e finais. PTP (*Point to Point*) é o comando no qual as posições intermediárias não são inseridas na programação da trajetória, que de fato não existe. Todos os motores são acionados ao máximo (limitado pela velocidade programada) de modo que a movimentação entre os dois pontos seja a mais rápida possível, em cada eixo. LIN CONT é a movimentação contínua na qual o ponto de destino não é exato, mas sim aproximado. Desse modo o robô não precisa ser desacelerado até parar no ponto de destino, ocorrendo menos desgaste no robô real e um menor tempo de ciclo.

3.5 Resultados

Esta seção apresenta alguns dos resultados obtidos com o ambiente virtual de manufatura implementado:

- Visualização estereoscópica do ambiente.
- Uso de interfaces de entrada tradicionais (teclado e *mouse*) para movimentação do robô (Fig. 5). Nesta figura ainda se apresenta a possibilidade de mover cada uma das juntas separadamente (cinemática direta) ou impondo uma posição ao efetuador (cinemática inversa).
- Movimentação do robô baseado na interpretação de arquivos com código KRL (Fig. 6).
- Interação com o robô virtual por meio do sensor Kinect (Fig. 7).

4 Conclusão

O objetivo deste trabalho foi o projeto e implementação de um ambiente virtual de manufatura mais imersivo e interativo, intuitivo e de baixo custo, com aplicações nas áreas de educação e treinamento. Neste sentido, modelou-se um robô Kuka KR15/2 utilizado

nas aulas de robótica do curso de engenharia mecatrônica da Escola Politécnica da USP serviu de base para a implementação de um *software* de simulação de um ambiente virtual de manufatura. Os dispositivos de entrada convencionais foram parcialmente substituídos. Neste sentido, o projeto foi bem sucedido, incluindo o uso do sensor de movimento (Kinect) e a tecnologia de estereoscopia.

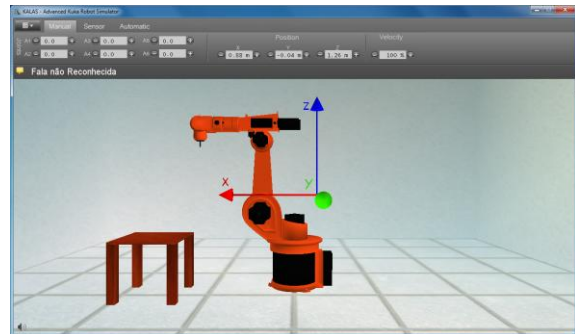


Figura 5. Movimentação do robô pelo acionamento de suas juntas (cinemática direta) ou pela posição do efetuador (cinemática inversa).

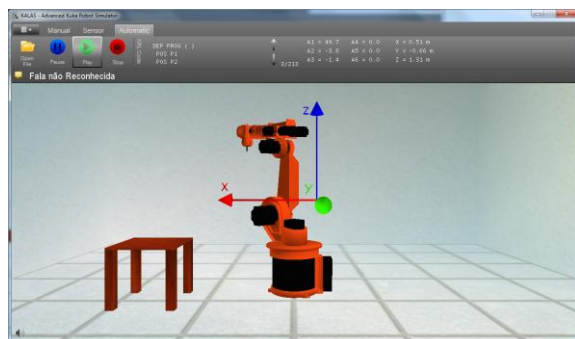


Figura 6. Movimentação do robô baseado em arquivo com código KRL.

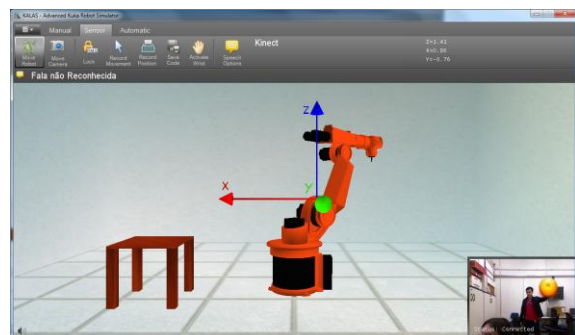


Figure 7. Interação com o robô por meio do sensor Kinect.

Como resultado, o ambiente virtual de manufatura pode ser utilizado tanto em situações de aprendizagem em faculdades e escolas de engenharia como para treinamento de funcionários especializados sem a necessidade de supervisão de uso ou de interromper o funcionamento da máquina real no chão de fábrica ou na unidade de ensino. A função de gravação de movimento pode ser utilizada em programação por aprendizagem passiva. Nela, o programador, que nesse caso é o usuário, movimenta o robô virtual pela trajetória desejada e grava as posições. Implementou-

¹² www.antlr3.org/

¹³ www.antlr3.org/grammar/1317637876413/krl.g

se mais de um modo de gravação, que dependendo da amostragem selecionada, comporta-se como movimentação ponto a ponto ou movimentação contínua, com ou sem aproximação. A partir do código KRL gerado, pode-se promover alguns ajustes, como otimizar as velocidades para cada trajetória e/ou excluir movimentos inúteis ou não intencionais. O código gerado pode então ser carregado e utilizado no robô real.

Mais informações sobre o projeto podem ser obtidos em (Santos, Lisboa, 2012) e (Miyashiro, Sugawara, 2011).

Os próximos passos são utilizar o OPC para promover a interação entre os modelos virtuais e entidades reais, permitindo que os modelos sejam utilizados como sistemas SCADA (*Supervisory Control And Data Acquisition*) para a monitoração e/ou operação remota; promover a interação entre vários usuários por meio do ambiente virtual de manufatura; e a modelagem de outros equipamentos.

Agradecimentos

Os autores agradecem as agências CAPES, CNPq e FAPESP por seus apoios.

Referências Bibliográficas

- Amici, S. de; Sanna, A.; Lamberti, F.; Pralio, B. (2010). "A Wii remote-based infrared-optical tracking system". *Entertainment Computing*, Vol. 1, Issues 3-4, pp. 119-124.
- Banerjee, P.; Zetu, D. (2001). *Virtual manufacturing*. New York: John Wiley & Sons.
- Cheng, Y., Wang, S. (2011). "Applying a 3D virtual learning environment to facilitate student's application ability - The case of marketing". In: *Computers in Human Behavior*, Vol. 27, No. 1, pp. 576-584.
- Curzel, J. L.; Hounsell, S.; Leal, A. B. (2007). "Uso da realidade virtual para ensino de automação da manufatura". *International Conference on Engineering and Computer Education*. pp. 773-777.
- Gonzalez-Jorge, H.; Riveiro, B.; Vazquez-Fernandez, E.; Martínez-Sánchez, J.; Arias, P. (2013). "Metrological evaluation of Microsoft Kinect and Asus Xtion sensors". In: *Measurement*, Vol. 46, pp. 1800-1806.
- Gumster, J. V. (2009). *Blender for Dummies*. Wiley Publishing, Inc.: United States of America, 412p.
- Gutiérrez, M. A.; Vexo, F.; Thalmann, D. (2008). *Stepping into Virtual Reality*. London, Springer, 2008.
- Kadir, A. A.; Xu, X.; Hammerle, E. (2011). "Virtual Machine Tools and Virtual Machining – A Technological Review". In: *Robotics and Computer-Integrated Manufacturing*, Vol. 27, No. 3, pp. 494-508.
- Kim, Y.-S.; Yang, J.; Han, S. (2006). "A multichannel visualization module for virtual manufacturing". *Computers in Industry*, Vol. 57, N. 7, pp. 653-662.
- Kuka Roboter (2010). *Kuka System Software 5.5 - Operating and Programming Instructions for System Integrators*. Disponível em <http://sites.poli.usp.br/d/PMR2560/Manual%20KUKA.pdf>
- Li, S.; Sun, J. (2009). "Application of Virtual Reality Technology in the Field of Sport". *2009 First International Workshop on Education Technology and Computer Science*, pp. 455-458.
- Miyashiro, E. R; Sugawara, K. J. (2011). *Simulação de um robô virtual utilizando manufatura virtual. Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos*.
- Mujber, T. S.; Szecsi, T.; Hashmi, M. S. J. (2004). "Virtual reality applications in manufacturing process simulation". *Journal of Materials Processing Technology*, Vol. 155-156, pp. 1834-1838.
- Paneels, S.A.; Roberts, J.C.; Rodgers, P.J. (2010). "HITPROTO: a tool for the rapid prototyping of haptic interactions for haptic data visualization". *Haptics Symposium, IEEE*, pp. 261-268.
- Pathak, R. and Joshi, S. (2009). "Implementation of Virtual Reality and Agents Model Based Nano Domain Simulation an HPC Setup". *First Asian Himalayas International Conference on Internet AH-ICI2009*, 5p.
- Potkonjak, V.; Vukobratovi, M.; Jovanovi, K.; Medenica, M. (2010). "Virtual Mechatronic/Robotic laboratory - A step further in distance learning". *Computers & Education*, Vol. 55, pp. 465-475.
- Santos, L. A. R. O.; Lisboa, H. B. (2012). *Movimentação de robô virtual por meio de sensores de som e movimento. Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos*.
- Siciliano B.; Sciavicco L.; Villani L.; Oriolo G. (2010). *Robotics: Modelling, Planning and Control*. London: Springer-Verlang, 632p.
- Straka, M.; Hauswiesner, S.; Ruther, M.; Bischof, H. (2011). "Skeletal Graph Based Human Pose Estimation in Real-Time". In *Proceedings of the British Machine Vision Conference, BMVA Press*, pp. 69.1-69.12.
- Tori, R.; Kirner, C. (2006). "Fundamentos de Realidade Virtual". *VIII Symposium on Virtual Reality. Fundamentos e Tecnologia de Realidade Virtual e Aumentada*. pp. 2-21.