
Um Ambiente para Simulação e Animação de Robôs

Humberto Cardoso Marchezi¹, Hans-Jorg Andreas Schneebeli¹

¹Departamento de Engenharia Elétrica – Universidade Federal do Espírito Santo (UFES)

Av. Fernando Ferrari, S/N – 29060-970 – Vitória, ES

hcmarchezi@yahoo.com, hans@ele.ufes.br

Abstract. *This article presents an environment to simulate robots for educational purposes for engineering students. Different from other robot simulators, this one lets the students build their own robotic mechanical configuration by chaining reusable mechanical parts called link types which can also be created by the students. Besides that, it is also possible to program the robot to perform a specific task. By commanding the joints, a student can store manipulator positions with distinct velocities in specific time instants in order to describe a path where a smooth trajectory is calculated and can be visualized in a tri-dimensional animation..*

Resumo. *Esse artigo apresenta uma proposta de ambiente de simulação de robôs com finalidades didáticas para alunos de engenharia. Ao contrário da maioria dos simuladores, esse ambiente permite que o aluno crie a sua própria configuração mecânica de robô através do encadeamento de peças mecânicas reutilizadas chamadas de tipos de elo (link) que por sua vez também podem ser criados pelo próprio aluno. Além disso, é possível programar o robô para desempenhar uma tarefa específica. Comandando as juntas, o aluno pode armazenar diferentes posições do manipulador com velocidades distintas em instantes de tempo específicos, descrevendo um caminho onde será calculada uma trajetória suave visualizada através de uma animação tridimensional.*

1. Motivação

Os robôs são cada vez mais usados nas mais diversas áreas como agricultura, geração de eletricidade, equipamentos eletrônicos, indústria automotiva, aeroespacial, entre outros. Com isso, cresce também a importância da disciplina de robótica ensinada nas universidades nos cursos de engenharia elétrica e mecatrônica.

Embora existam iniciativas para aperfeiçoar o ensino dessa disciplina, a robótica ainda é apresentada ao aluno de maneira positiva através dos livros nas universidades brasileiras.

Entretanto, construir um robô exige mão-de-obra e peças muito caras para o orçamento da maioria das instituições de ensino surgindo a necessidade de simular o funcionamento de um robô virtual no computador.

Para isso, existem softwares livres disponíveis como o simulador JRobot e o RoboSim. Mas estes softwares, simulam apenas um determinado tipo de manipulador. Por outro lado, os simuladores comerciais como EASY-Rob custam vários milhares de

dólares. Essas razões justificam a construção de um tipo de simulador que permita a simulação de robôs manipuladores descritos como uma cadeia de elos *links* de várias formas diferentes e onde um determinado tipo de elo possa ser reutilizado em outros projetos de robôs.

Dessa forma, o aluno não estará preso as configurações pré-determinadas como robô *puma* ou *scara* em outros simuladores e pode simular qualquer robô manipulador que seja representado por uma cadeia de elos. Somado ao isso, é importante que os tipos de elo possam ser criados pelo próprio aluno que poderá criar a sua própria biblioteca de tipos de elo possibilitando mais liberdade de criação.

A partir da concepção final do robô, o simulador deve permitir ao aluno programar e verificar o funcionamento do robô desempenhando uma tarefa específica por meio de uma animação.

Basicamente, esse ambiente de simulação e animação de robôs deve ser capaz de imitar o comportamento de um robô permitindo, dessa forma, que este possa ser montado e testado tanto na sua arquitetura como na sua mecânica e controle.

2. O Problema da Simulação de Robôs Manipuladores

2.1 Descrição de Robôs

Um robô manipulador possui um braço que assegura mobilidade e um punho que assegura destreza. O manipulador é composto de uma cadeia de elos rígidos conectados por juntas que são responsáveis pela movimentação do próximo elo na cadeia. [Craig, 1989]

Existem vários tipos de junta para robôs que variam de acordo com o tipo de deslizamento entre as superfícies de dois elos. Entretanto devido a considerações mecânicas, manipuladores são geralmente construídos com juntas de um grau de liberdade. Desses, a maioria dos manipuladores possuem juntas de revolução ou possuem juntas deslizantes chamadas de juntas prismáticas.

Dada a descrição acima e como se trata de uma simulação no espaço de coordenadas tridimensionais, resta saber como posicionar e orientar unicamente todos estes objetos no espaço tridimensional.

Para representar as transformações de rotação e translação em uma mesma matriz reduzindo o cálculo para multiplicação de matrizes, usa-se o sistema de coordenadas homogêneas onde um ponto $P(x,y,z)$ passa a ter um quarto componente vetorial que é um escalar através da forma (xh, yh, zh, h) . [Mortenson, 1999]

Dessa forma, uma rotação R e uma translação T ficariam representados dentro de um única matriz denominada *frame* transformando o P em coordenadas homogêneas no ponto P' .

$$[P'] = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} [P]$$

Por isso, um *frame* ${}^{i-1}T_i$ define uma transformação única de um sistema de coordenadas cartesianas $i-1$ para um sistema de coordenadas i . [Craig, 1989]

2.2 Modelagem Cinemática

A cinemática direta trata do movimento do Robô sem levar em consideração as forças que o causaram mas levando em consideração a posição e orientação dos elos da base até o atuador final.

Como o manipulador é uma cadeia cinemática aberta de elos e que uma junta só conecta dois elos consecutivos é razoável fazer uso de uma convenção para descrever a posição e orientação desses elos de maneira recursiva. Daí surgiu o propósito da convenção de Denavit-Hartenberg.[Craig, 1989]

Por essa convenção, um elo é definido como um relacionamento entre duas juntas de um robô descrito através do comprimento, torção, deslocamento da junta e ângulo da junta. Os elos que controlam uma junta de revolução, possuem o ângulo das juntas como parâmetro variável enquanto que nos elos que controlam uma junta prismática, o deslocamento das juntas é variável.

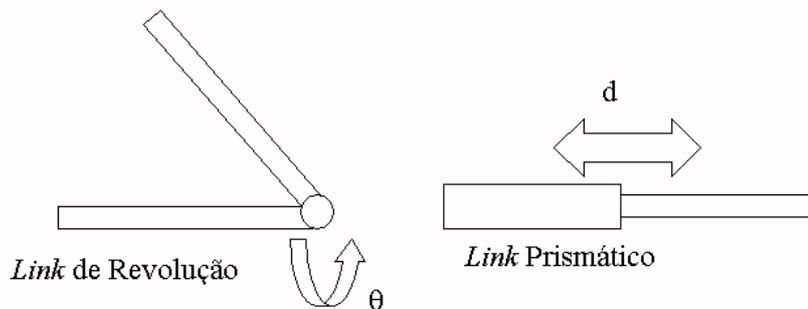


Figura 1. Tipos de junta de robô

Cada elo i é numerado de $(i=0)$ onde está a base até $(i=n)$ onde está o punho. Seguindo a mesma convenção o par de juntas de um elo i é numerada como $i-1$ e i . Além disso, uma junta só possui um grau de liberdade como um ângulo de junta ou deslocamento de elo numa direção. Da mesma forma dada elo i possui um *frame* i que faz referência ao *frame* $i-1$.

A matriz resultante do *frame* é representada por $T(a, \alpha, \theta, d)$. Onde a é comprimento do elo, α é a torção, θ é o ângulo entre as juntas e d é o deslocamento entre as juntas conforme mostrado na figura abaixo:

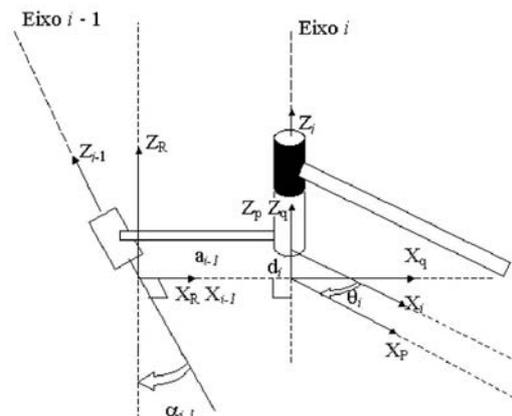


Figura 2. Modelagem cinemática de um robô

2.3 Especificação de Trajetórias

O problema básico a ser tratado aqui é como mover a ferramenta localizada no atuador final do robô manipulador de uma posição inicial para uma posição final informada. Ou seja como mover uma ferramenta de um $T_{inicial}$ a um T_{final} alterando posição e orientação da ferramenta em relação a estação.

A trajetória do robô pode ser descrita através de duas abordagens: espaço de juntas e espaço cartesiano. No espaço cartesiano, a trajetória do manipulador é descrita através de uma seqüência de *frames* que descrevem a posição e orientação da ferramenta localizada na extremidade final do robô em relação a estação enquanto que no espaço de juntas as posições são descritas através dos valores de deslocamento das juntas. Por ser mais simples computacionalmente, o espaço de juntas foi utilizado no simulador proposto e essa abordagem será tratada daqui em diante.

Existem três principais abordagens para a descrição da trajetória.

- Ponto-a-Ponto

Essa é a abordagem mais simples onde o caminho do manipulador é descrito informando-se somente as suas posições inicial e final com velocidade zero em ambas posições especificadas. A trajetória é calculada a partir de uma polinomial cúbica com quatro restrições e quatro incógnitas.

Sendo a polinomial da trajetória:

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3$$

As restrições são definidas através das posições e velocidades inicial e final:

$$\text{Posição Inicial e Final} \begin{cases} \theta(0) = \theta_0 \\ \theta(t_f) = \theta_f \end{cases}$$

$$\text{Velocidade Inicial e Final} \begin{cases} \theta(0)' = 0 \\ \theta(t_f)' = 0 \end{cases}$$

- Ponto-a-Ponto com Vias

Às vezes é necessário especificar o movimento do robô em mais detalhes, já que durante a execução do movimento entre um ponto e outro a trajetória pode não ser retilínea, pois caso contrário, o manipulador poderia colidir com algum obstáculo ou com ele próprio. Também é muito importante notar que em muitas aplicações como em soldagem de materiais, por exemplo ou em pintura de automóveis, o formato da trajetória em si é muito mais importante do que as posições inicial e final. A partir daí (Para resolver esse problema) surgiu a necessidade de se usar pontos intermediários, chamados de pontos via que devem ser escolhidos para garantir a melhor execução da trajetória.

Dessa maneira, a descrição da trajetória de um robô é feita através da definição de um caminho. Onde um caminho é definido por um conjunto de pontos através dos quais o manipulador deve passar na execução de um determinado movimento, ou seja, é a descrição puramente geométrica do movimento. [Sciavicco e Siciliano, 1996]

Nesse método as posições em vários instantes de tempo são informadas com velocidade zero e uma trajetória é calculada para cada par de posições consecutivas que descrevem caminho usando ponto-a-ponto. Embora simples, a desvantagem desse método é que a animação da trajetória é descontinuada devido ao valor nulo da velocidade em cada ponto.

- Ponto-a-Ponto com Vias e Velocidade Não-Nula

Na abordagem anterior, as paradas no movimento do manipulador durante a execução de sua tarefa fazem com que a trajetória seja descontinuada o que produz um esforço maior para os atuadores e um desgaste mecânico nas juntas do robô. A partir disso, surgiu a necessidade de se escrever uma trajetória usando pontos via com velocidade não nula.

Nessa abordagem para calcular a polinomial cúbica da trajetória, as restrições passam a ser:

$$\text{Posição Inicial e Final} \begin{cases} \theta(0) = \theta_0 \\ \theta(t_f) = \theta_f \end{cases} \text{ e}$$

$$\text{Velocidade Inicial e Final} \begin{cases} \theta(0)' = \theta_0' \\ \theta(t_f)' = \theta_f' \end{cases}$$

Para calcular a função interpolada $f(t)$ do valor da junta no tempo, é necessário satisfazer condições de velocidade e posição, logo para cada ponto que forma o caminho, além da posição, uma velocidade é associada. Dessa forma, uma polinomial Π é calculada para cada par de pontos do caminho e para cada junta do robô de tal forma que a velocidade final de uma polinomial Π'_{i-1} seja igual a velocidade inicial da polinomial Π'_i e da mesma forma a posição final de Π_{i-1} seja igual a posição inicial de Π_i . Ou seja:

$$\Pi'_{i-1}(t_f) = \Pi'_i(t_0)$$

$$\Pi_{i-1}(t_f) = \Pi_i(t_0)$$

3. Um Ambiente para Simulação e Animação de Robôs

3.1 Especificação de Sistemas

Ao interagir e observar os resultados, além de servir como um meio de aprendizado para alunos de elétrica e mecatrônica nas disciplinas de robótica, um ambiente de simulação também permite que a simulação virtual de um robô possibilite testar aspectos como configuração mecânica e os controles a serem informados garantindo não só a confiabilidade da tarefa a ser desempenhada como também economia de tempo e dinheiro do projetista de robô. Tendo isso em mente, o objetivo desse ambiente é o de prover uma ferramenta capaz de simular alguns aspectos de robótica tais como: descrição de tipos de elo, descrição de robôs, cinemática direta e planejamento de trajetória.

- Descrição de Tipos de Elo

Para promover a reusabilidade de tipos de elo, o ambiente deve permitir que eles sejam criados para serem utilizados em vários robôs. Dessa forma, ao criar esses componentes, o projetista deve informar apenas as suas características estáticas, ou seja, aquelas que não variam cinematicamente por causa de um movimento das suas juntas. Conforme visto na seção 2.2, esses parâmetros variam dependendo do tipo de deslizamento entre as juntas. Caso seja um tipo de elo de revolução devem ser informados os parâmetros de comprimento, torção e deslocamento das juntas e se for um tipo de elo prismático devem ser informados os mesmos parâmetros exceto o último que passa a ser o ângulo entre as juntas. Além disso, o ambiente de simulação deve permitir que uma geometria possa ser informada representando o formato físico do tipo de elo e um sistema de coordenadas local para corrigir o tamanho, a posição e a orientação dessa geometria importada. Finalmente também o tipo de elo deve prover uma indicação para informar se este é uma base, um elo ou uma ferramenta.

- Descrição de Robôs

Também seguindo a seção 2.2, este ambiente de simulação deve permitir que um robô manipulador possa ser construído através da inserção ou remoção de um elo, onde o elo, por sua vez, deve seguir as propriedades do seu tipo de elo e só pode fazer parte de um robô. Adicionalmente, o ambiente deve se certificar que o elo foi inserido corretamente no encadeamento do robô levando-se em conta que uma base só pode estar na 1ª posição do encadeamento, uma ferramenta só pode estar na última posição e o elo pode estar em qualquer posições respeitando a posição de uma base ou ferramenta se houver.

- Cinemática Direta

O simulador deve permitir que o projetista possa interagir cinematicamente com o robô alterando o valor de suas juntas e que possa visualizar este efeito graficamente.

- Planejamento de Trajetória

Finalmente, o ambiente de simulação de robôs deve permitir que uma trajetória para um robô possa ser descrita através de um caminho usando-se ponto-a-ponto com vias e velocidade não-nula descrito na seção 2.3.3. Usando-se esse esquema, o simulador deve permitir que as posições do robô em um instante de tempo sejam armazenadas no caminho com uma velocidade especificada fazendo com que a trajetória resultante seja suave e contínua.

Após a descrição do caminho e cálculo da trajetória, o ambiente de simulação também deve permitir a visualização do resultado através de uma animação.

3.2 O Projeto

Esse ambiente de simulação produz animações tridimensionais em tempo real o que exige uma linguagem de programação que seja orientada a objeto para suportar o paradigma de proposto, mas também de alta performance. Essas razões justificaram o uso da linguagem C++ para a implementação do protótipo.

Para que o código pudesse ser compilado para uma maior variedade de sistemas operacionais, utilizou-se apenas o C++ padrão e o C++ STL.[Stroustrup, 1997]

Entretanto, para o desenho da geometria dos objetos tridimensionais assim como a manipulação do mundo 3D desses objetos foi utilizado o padrão OpenGL.[Wright and Sweet, 2000] Por fim, para o projeto de interfaces gráficas e controle de eventos foi utilizada a biblioteca de desenvolvimento Qt.[Trolltech, 2002]

Com relação à organização, para o melhor gerenciamento, o ambiente de simulação foi dividido em três subsistemas. Cada subsistema agrupa um conjunto de funcionalidades afins e pode ser trabalhada independentemente. Além disso, cada subsistema é representado como uma janela filha dentro de uma janela principal da aplicação do protótipo seguindo o estilo *Multiple Document Interface*.

O subsistema *Links* é responsável pela descrição de um tipo de elo que poderá ser usado na construção de um robô assim como alteração ou exclusão daqueles já existentes.

O subsistema Robôs possui as funcionalidades da construção de um robô que é feita a partir da anexação de novos elos, criados pelo subsistema acima, na cadeia cinemática aberta do manipulador ou através da remoção de elos existentes no mesmo. Cada elo adicionado, tem o seu respectivo controle de movimento no lado direito da janela que produzirão um movimento de rotação ou translação na junta controlada por este caso o elo seja de revolução ou prismático.

O subsistema trajetórias, após a construção do robô, cuida da descrição de um caminho para a trajetória do robô.

Para descrever esse caminho, os elos do robô são posicionados através dos seus controles de movimento. Cada nova posição do robô pode ser armazenada através do botão Adicionar Posição fornecendo-se também o instante de tempo e a velocidade de cada junta nesse instante. Essa posição é o ponto do caminho ou ponto via se estiver entre a posição inicial e a posição final do robô.

Para cada inserção, alteração ou remoção no caminho do robô, a sua trajetória é recalculada automaticamente. Ao final, a trajetória pode ser verificada por meio da opção de Animação do robô.

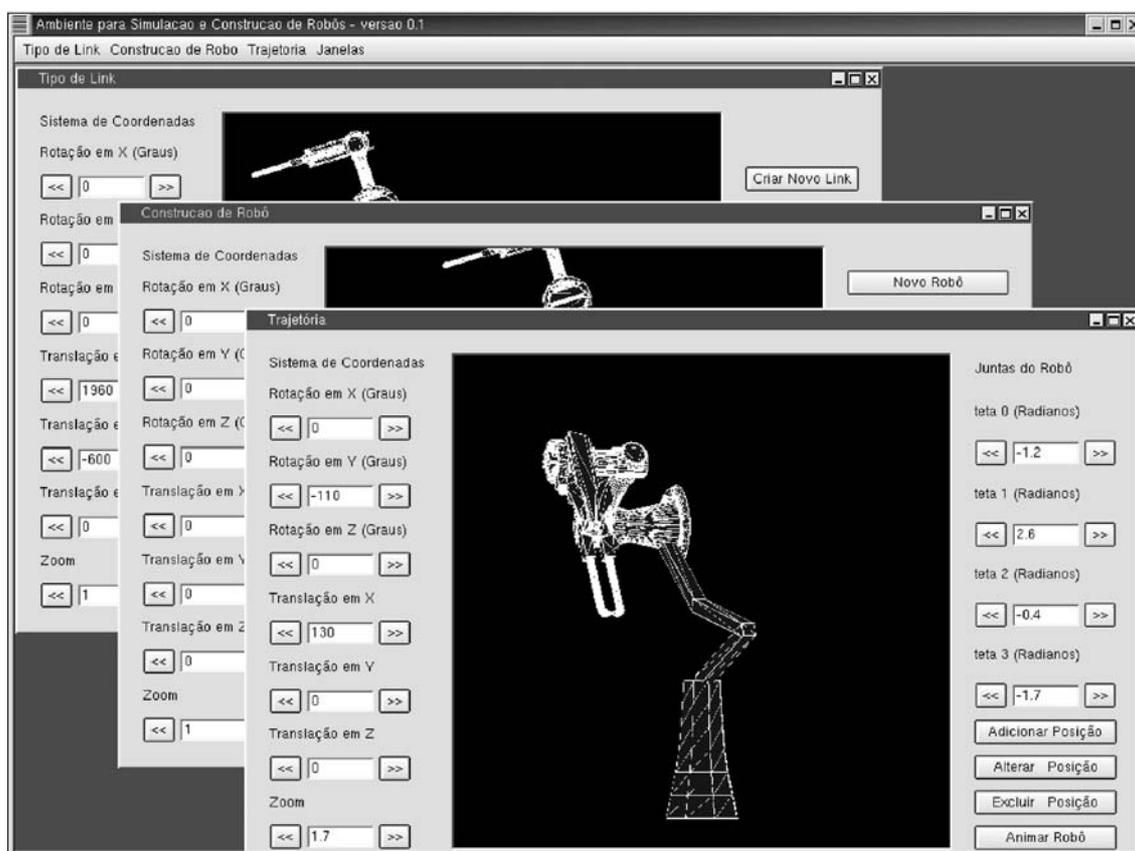


Figura 3. Janela Principal do Ambiente

4. Conclusões

Este tipo de ambiente de simulação para robôs mostra-se especialmente importante para qualquer estudante ou profissional que deseja entender o funcionamento cinemático dos robôs. Para um estudante, essa simulação vai ajudar na compreensão mais sólida dos conceitos de robótica por permitir a ele uma visão experimental e prática do assunto.

Podem ser adicionadas outras capacidades ao sistema como dada a posição do atuador final, calcular o valor do deslocamento ou inclinação das juntas através da resolução de um sistema de equações de cinemática inversa. Através desse sistema o estudante poderia visualizar o robô e verificar rapidamente a resposta desse sistema de equações. Além disso, vários itens nesse sistema podem ser estendidos nos seguintes aspectos:

- O desenho do robô pode ser feito utilizando recursos mais poderosos do OpenGL como textura e iluminação produzindo uma imagem mais realista;
- Na criação de um Tipo de Link, ainda não foi pensada uma interface para calcular iterativamente um sistema de coordenadas de tal forma que a origem desse sistema coincida com a primeira junta do Tipo de Link;
- É necessário que seja catalogada uma lista de arquivos com informações de geometria de vários elos de robôs diferentes proporcionando mais opções na construção do robô.

Finalmente, deve-se lembrar que, participar desse projeto trouxe vários benefícios, pois devido a sua característica multidisciplinar possibilitou conhecer várias áreas: Robótica, Computação Gráfica e Engenharia de Software.

5. Referências

- “Easy-rob simulator.”, disponível em http://www.easy-rob.de/m4_3.htm, acesso em 22 de dezembro de 2003
- “JRobot simulator”, disponível em <http://jrobot.sourceforge.net>, acesso em 10 de janeiro de 2004
- “RoboSim Simulator”, disponível em <http://ieee.uow.edu.au/~daniel/software/RoboSim/>, acesso em 10 de janeiro de 2004
- Craig, J. J. (1989), “Introduction to Robot Mechanics and Control Second Edition.”, Addison-Wesley.
- Mortenson, M. E. (1999), “Mathematics for Computer Graphics Applications”, Industrial Press Inc, New York, NY, EUA.
- Sciavicco, L. and Siciliano, B. (1996). “Modeling and Control of Robot Manipulator”, McGraw-Hill, Nápoles, Itália.
- Stroustrup, B. (1997), “The C++ Programming Language Third Edition”, Addison-Wesley, Murray Hill, NJ, EUA.
- Trolltech (2002). “Qt Reference Documentation.”, <http://doc.trolltech.com/2.3/index.html>
- Wright, R. S. and Sweet, M. (2000). “OpenGL Super Bible Second Edition.”, Waite Group Press, Indianápolis, IN, EUA.