

AMBIENTE DE PROGRAMAÇÃO *OFF-LINE* PARA O KIT DE ROBÓTICA ROBIX RCS-6

Carlos A. Bertotto¹
bertotto@vanet.com.br

José Luís A. Güntzel
(Orientador)
jguntzel@ulbra.tcche.br

Universidade Luterana do Brasil
Rua Miguel Tostes, 101 – 92 420-280 - Canoas, RS, Brasil
Bacharelado em Ciência da Computação, Centro de Ciências Naturais e Exatas

Resumo^{3/4} O presente trabalho descreve a implementação de um ambiente de programação *off-line* que permite ao operador programar um robô sem que este tenha que interromper a atividade corrente. Para isto, foi desenvolvido um ambiente gráfico que modela os componentes da célula de trabalho na qual o robô está inserido. Desta forma pode-se especificar, neste ambiente gráfico, uma nova tarefa e, posteriormente, enviá-la para ser processada pela célula de trabalho, reduzindo o tempo de programação de novas tarefas e permitindo a implementação de sistemas flexíveis de manufatura (FMS – *Flexible Manufacturing Systems*). Na construção deste ambiente, foram utilizados o *kit* de robótica ROBIX RCS-6, uma maquete funcional de uma célula de trabalho industrial, a linguagem de programação C e a biblioteca gráfica OpenGL.

Abstract— The main goal of the present work is the construction of an off-line programming environment that allows the operator to program a robot without interrupting the current activity. To accomplish this, a graphic environment that models the components of work cell in which the robot is inserted was developed. Thus, the programmer can create a new task, which is sent to be processed by the robot, reducing the programming time of new tasks and, by its turn, allowing an implementation of Flexible Manufacturing Systems (FSM). In the construction of this environment it is used the ROBIX RCS-6 educational kit, a functional model of an industrial working cell, the C programming language and OpenGL graphic libraries.

Keywords^{3/4} *Robotics, Off-line Programming, Computer Graphics*

1 INTRODUÇÃO

¹ Atualmente realizando mestrado no Programa de Pós-Graduação em Computação do Instituto de Informática da Universidade Federal do Rio Grande do Sul, na linha de pesquisa de Sistemas de Tempo Real.

Desde a sua introdução em ambientes industriais, os robôs têm permitido a automatização de processos repetitivos e perigosos para o ser humano. Neste contexto, os sistemas utilizados para o controle destes ambientes automáticos tem evoluído substancialmente a ponto de permitir níveis de precisão cada vez mais acurados, possibilitando o uso dos robôs em um número maior de aplicações.

Diferentemente dos sistemas de controle, os métodos de programação para a robótica industrial estão mais defasados. A maioria dos sistemas robóticos em uso utiliza o chamado **Método de Ensino** para a programação de seus robôs, o que acaba por inviabilizar o uso dos mesmos em diversas aplicações onde os lotes de produção são pequenos, já que este método exige a presença física do robô, tirando-o de operação por um tempo considerável. Na tentativa de solucionar esse problema este trabalho propõe um ambiente de programação que possibilite otimizar a criação de programas para manipuladores robóticos. Esse ambiente permite ao programador criar uma nova iteração para um lote específico de peças, sem que seja necessária a utilização de métodos de programação por ensino. Esse tipo de programação que permite programar um robô sem que o mesmo esteja fisicamente conectado ao sistema é chamado de **Método de Programação Off-line**. Além de otimizar a programação, essa abordagem permite a implementação de Sistemas Flexíveis de Manufatura (FSM), que englobam o sistema fabril, integrando estoque, expedição, projeto, produção e venda de produtos [GRO 88] [NOF 85] [MCW 86] [MCK 96].

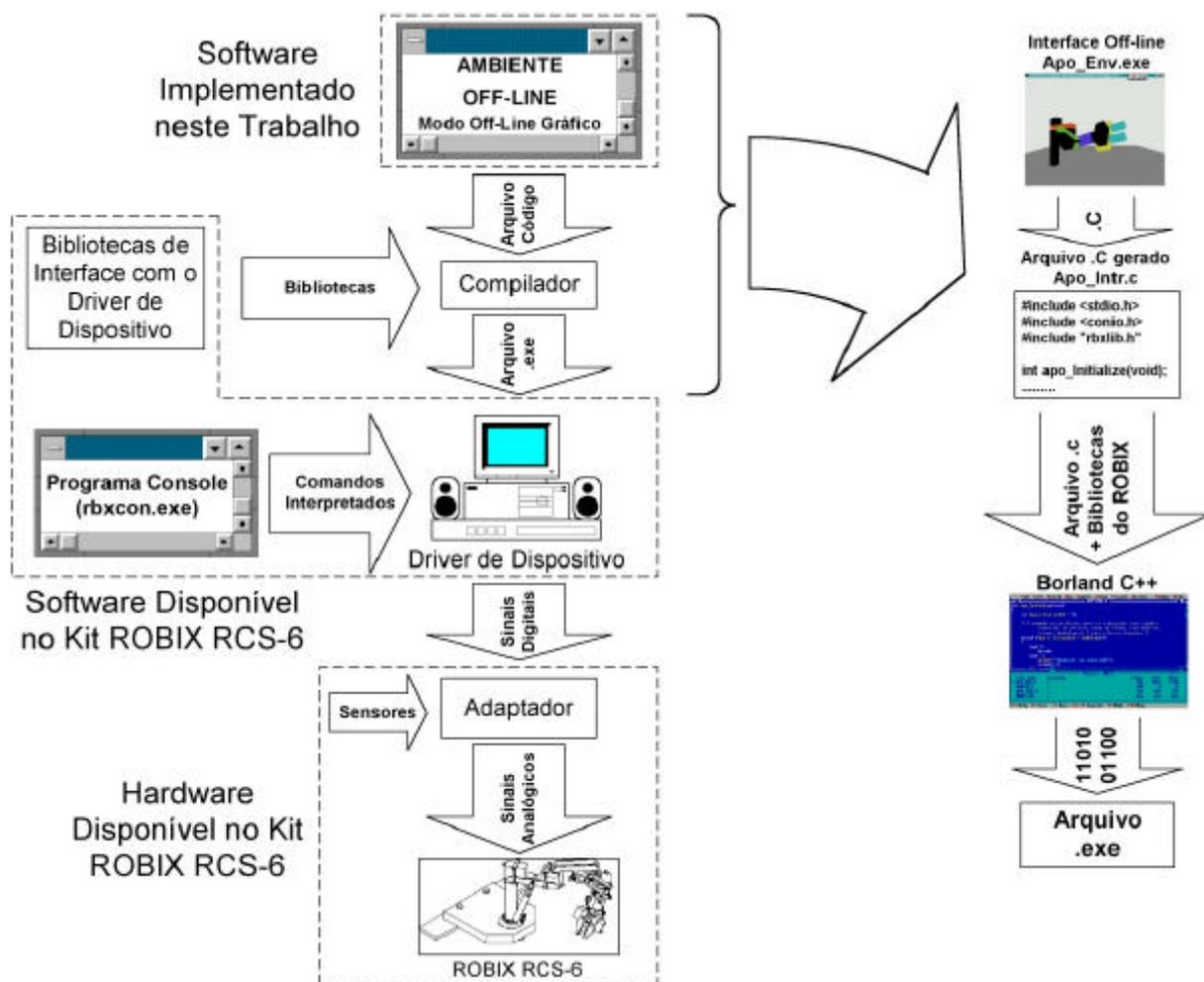


Figura 1: Diagrama esquemático completo (esq.) e detalhado (dir.) do ambiente proposto

Na construção deste ambiente, que se desenvolveu no contexto de um Trabalho de Conclusão de Curso, foi criada uma interface gráfica que implementa o sistema de programação *off-line*, onde foram modelados, a partir de uma maquete real, uma célula de trabalho industrial em escala reduzida e o *kit* educativo de robótica ROBIX RCS-6 [ADV 99]. Na figura 1, podemos visualizar o diagrama esquemático desta proposta e onde ela se insere nos módulos disponíveis do ROBIX RCS-6.

2 CARACTERÍSTICAS E COMPONENTES DO KIT ROBIX RCS-6

O ROBIX RCS-6 possui características da robótica industrial em um sistema barato, de fácil construção e educativo. Os principais módulos deste *kit* são mostrados na figura 1. Na figura 2, temos a foto do ROBIX RCS-6.

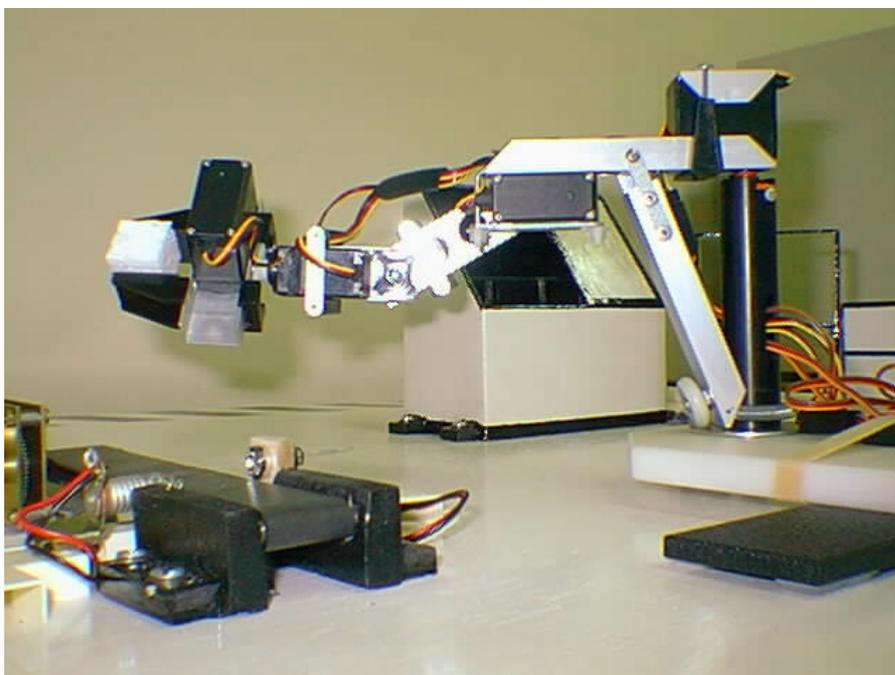


Figura 2: ROBIX RCS-6

O adaptador do kit ROBIX RCS-6 é um dispositivo eletrônico que controla as funções do robô e conecta-se a um computador compatível com IBM PC, através de uma porta paralela. Possui seis saídas para controle de servomotores de aplicação em aeromodelos. Apresenta também oito entradas analógicas, com um conversor analógico-digital de oito canais e oito bits, que permite a conexão de sensores. Possui também sete chaves digitais que comandam a comutação de cargas externas, além de duas saídas que fornecem 150mA de corrente contínua para a alimentação de pequenos motores e lâmpadas.

A interface ou programa console (rbxcon.exe) é um ambiente de programação que permite programar os movimentos do robô. O programa console é executado no sistema operacional DOS (*Disk Operating System*), provendo um ambiente de programação industrial para robôs de exploração. Este programa possui uma linguagem de *script*, a qual permite controlar o robô facilmente. O programa console implementa o modo de ensino (*Teach Mode*) para programação sem que comandos tenham que ser digitados, além de exibir continuamente os parâmetros do robô e as leituras dos sensores. Esta interface não foi utilizada por tratar-se de uma interface incompatível com o trabalho proposto, sendo substituída por uma interface gráfica, que implementa o ambiente *off-line*.

O *driver* de dispositivo (*rbxdrv.exe*) provê a interface de baixo nível com o adaptador do ROBIX RCS-6. Este programa é automaticamente instalado na memória quando o programa console é executado. Ele permite que o controle do ROBIX seja feito por meio do programa console ou das bibliotecas de interface com o *driver* de dispositivo, incluídas no *kit*. O *driver* de dispositivo permite uma coordenação automática dos movimentos simultâneos dos servos, ajustando automaticamente a velocidade dos mesmos para realizarem movimentos suaves, permitindo que todos iniciem e parem ao mesmo tempo.

As bibliotecas de interface com o *driver* de dispositivo, que são referenciadas neste artigo simplesmente por **bibliotecas do ROBIX**, possuem várias rotinas avançadas que permitem programar o robô diretamente. Estas rotinas permitem o controle completo do ROBIX RCS-6, possibilitando o monitoramento e o controle de suas ações, parâmetros, estados, sensores, relatórios de erros, carga de arquivos, etc. As bibliotecas do ROBIX permitem a programação condicional requerida, quando o comportamento do robô é dado pela leitura de sensores provenientes do exterior.

Para realizar o movimento nas juntas, foram incluídos seis servomotores Hitec HS-422. Eles são controlados por pulsos que são convertidos em informação posicional, codificada pela largura do pulso. Neste tipo de servos o eixo tem uma gama de rotação de 180 graus ou meia volta. Para mover o eixo do servo, o adaptador manda seqüências de pulsos de controle, onde a largura destes pulsos irá aumentar, quando o movimento for horário, e diminuir, quando o movimento for anti-horário.

2.1 Como o Movimento é Computado pelo Driver de Dispositivo

Robôs podem tremer e vibrar quando iniciam e param o movimento. Enquanto a vibração é característica das máquinas em geral, técnicas modernas de projeto vem reduzindo drasticamente a vibração nas máquinas em uso.

No caso do ROBIX RCS-6, a técnica de controle conhecida por “Perfil de Velocidade Trapezoidal” é utilizada pelo driver de dispositivo para produzir inícios e paradas suaves para cada servo, o que é feito mediante o ajuste dos parâmetros de aceleração/desaceleração e velocidade máxima. O *driver* de dispositivo calcula o tempo necessário para cada porção do movimento de acordo com os parâmetros configurados para cada servo. Isto explica porque estes valores devem ser modificados e otimizados para cada servo com o objetivo de diminuir a vibração.

3 DESENVOLVIMENTO DO AMBIENTE DE PROGRAMAÇÃO OFF-LINE USANDO OPENGL

O objetivo do presente trabalho foi a criação de um Ambiente de Programação *Off-line* utilizando os módulos incluídos no *kit* ROBIX RCS-6. O desenvolvimento deste ambiente foi dividido em quatro etapas distintas.

Na etapa inicial, foram definidas as bases do trabalho proposto, ou seja, a forma do robô, suas particularidades e sua operação, a fim de realizar uma configuração ótima, que se enquadrasse nos objetivos do trabalho. Esta fase consistiu na montagem do robô, verificando as peças prováveis e testando-as com configurações variáveis, possibilitando ao robô, alcançar objetos e manipulá-los com facilidade. A configuração do robô utilizada neste trabalho é a do tipo articulada, por apresentar maior liberdade de movimentos. O robô possui cinco graus de mobilidade, sendo que três deles pertencem ao punho, dando à garra movimentos completos e em todas as direções. O corpo do robô possui apenas dois graus de mobilidade, permitindo movimentos compostos, sem torná-lo muito complexo.

A segunda etapa consistiu na criação de um maquete funcional de uma célula de trabalho industrial real. Nesta maquete, que aparece na figura 3, podem ser realizados os mais variados testes com o ambiente proposto como a simulação de novas situações, criação de diferentes iterações e até mesmo testando o comportamento do robô e da célula em situações de risco. Inicialmente, foi necessária a decisão sobre quais componentes seriam colocados na maquete. Decidiu-se colocar uma esteira de transporte de carga que, com a ajuda de um sensor foto-elétrico (foto-transistor), pode movimentar-se e parar quando a peça barra a passagem de luz; um trono mecânico, um pequeno armário e uma caixa de expurgo de materiais. O próximo passo era adquirir e preparar uma superfície que pudesse comportar bem o robô e os componentes da maquete. Foi escolhido um compensado de 1m x 80cm que foi devidamente preparado e pintado. As peças da maquete foram feitas utilizando-se madeira do tipo balsa. Depois de prontos, os modelos foram fixados na maquete, permitindo que as ligações elétricas fossem feitas.

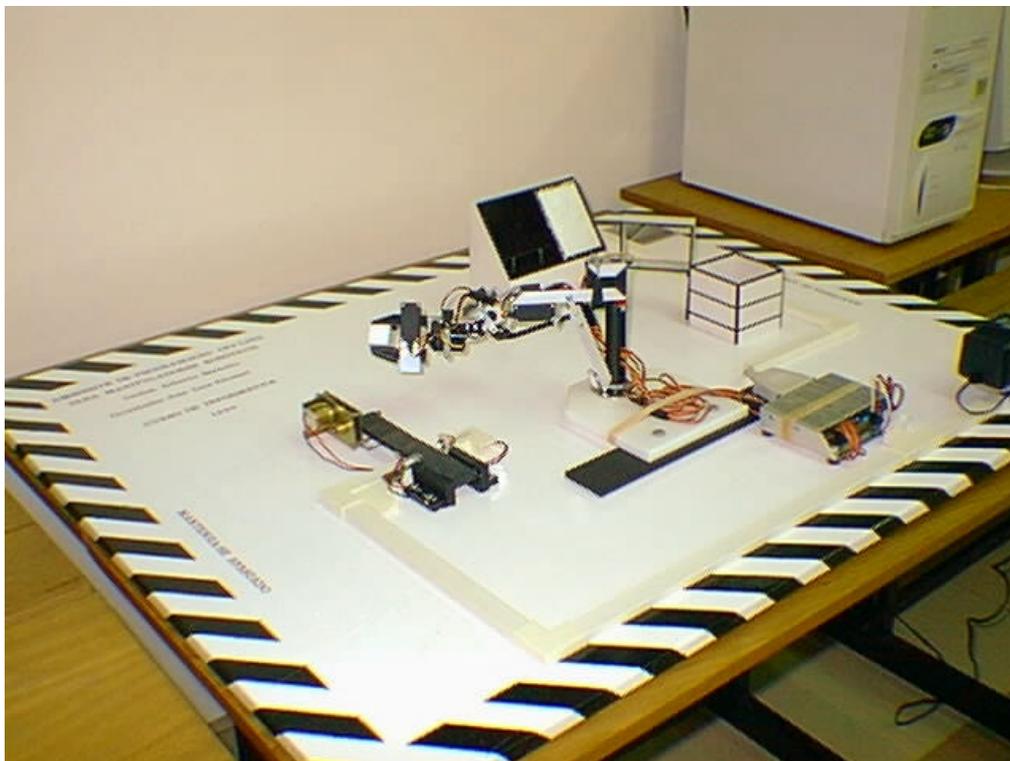


Figura 3: Maquete funcional do sistema apresentado

Na terceira etapa foi implementado um grupo de funções que corresponde a uma camada de software para a comunicação entre a interface gráfica do ambiente de programação *off-line* e as rotinas das bibliotecas do ROBIX. Estas funções encapsulam uma série de rotinas das bibliotecas do ROBIX e prestam um serviço que consiste em realizar a ligação entre a interface gráfica e o *driver* de dispositivo. Estas funções possibilitam realizar uma gama de ações que são agrupadas conforme o estado atual do robô. O termo **função**, neste artigo, refere-se ao grupo de funções criadas para a interface gráfica de programação conseguir interagir com o *driver* de dispositivo.

Na quarta e última etapa foi implementada a interface gráfica em OpenGL [SIL 99] [THE 99] [WOO 97], que é mostrada na figura 4. A programação *off-line* para robôs necessita de uma interface que simule o funcionamento completo de um ambiente de produção como uma célula de trabalho. Além deste ambiente simulado, o programa deve conhecer as particularidades desta célula e do próprio robô. Dentre estas particularidades temos a

seqüência de funcionamento de máquinas como esteiras de carga e tornos mecânicos, forma física do robô, potência e gama de movimentos dos propulsores das juntas, dentre outras. Para permitir que o programa conhecesse estas particularidades do ambiente real, foi necessário além da construção da interface gráfica, uma série de estruturas que encapsulam as formas, procedimentos e comportamento dos vários componentes da maquete real e também do robô. Neste ponto houve a necessidade da resolução de um problema importante que poderia inviabilizar a construção deste sistema: a diferença entre as medidas de movimento em OpenGL, que são expressas em graus e as do ROBIX, que são expressas em pulsos de motor. A solução encontrada foi a utilização de conversão de unidades. O resultado desta conversão foi então usado para transformar cada movimento na interface do ambiente *off-line* em pulsos de motor que o driver de dispositivo do ROBIX pudesse compreender e executar.



Figura 4: Interface do Ambiente de Programação *Off-line*

A programação neste ambiente é feita utilizando-se a mesma filosofia que qualquer sistema robótico que utilize programação por ensino, com a diferença que o robô está desenhado virtualmente na tela e não está *on-line* com o computador controlador no momento da programação. Para realizar a programação existe um grupo de teclas, que simulam uma interface de programação por ensino (*Teach Box*) e permitem ao programador criar iterações compostas, oferecendo facilidade de manipulação e a possibilidade de correção de possíveis erros, sem que estes sejam transmitidos ao programa que controla o robô fisicamente.

A interface gráfica de programação detecta o movimento comandado pelo teclado e o transforma em valores numéricos inteligíveis ao driver de dispositivo do ROBIX RCS-6, adicionando a esses valores, as rotinas correspondentes. Esses valores e as rotinas, que representam as posições válidas do robô, junto com o número de iterações que se deseja que o robô realize, vão sendo gravadas em um arquivo texto, com o formato de um código escrito em linguagem C. Uma vez concluída a programação de movimentos no ambiente gráfico, o arquivo gerado deve ser compilado, resultando em um programa executável que poderá ser enviado diretamente para ser processado pelo robô.

4 CONCLUSÃO

Os sistemas de programação *off-line* são uma excelente alternativa na programação de robôs utilizados na otimização de processos de fabricação de bens de consumo, mas os sistemas disponíveis atualmente chegam a custar centenas de milhares de dólares. Assim, este trabalho apresenta a implementação de um sistema *off-line* de simulação e controle de robôs manipuladores com os conceitos desta abordagem, e utilizando componentes relativamente baratos, permitindo a instituições o usarem como ferramenta de ensino e desenvolvimento de simulações de sistemas fabris. Apesar disso, este trabalho é um protótipo que tem limitações, no que se refere a sua portabilidade, já que seu uso se restringe apenas para o *kit* ROBIX RCS-6 e com a configuração proposta. Para permitir que mais robôs pudessem utilizar a mesma interface seria necessário que fossem padronizados os controladores destes robôs e estes tivessem um grupo de comandos básicos que poderiam ser configurados através de modelos matemáticos e físicos [WRI 97] [GRO88]. Porém, como este trabalho tem uma abordagem didática, servindo principalmente para o ensino da robótica em universidades e escolas, esta limitação não seria importante.

Como continuação do que foi implementado, existem várias possibilidades que poderiam incrementar a funcionalidade deste sistema. Dentre elas, podem ser citadas:

A construção de um Gerador de Ambientes para Programas Off-line. Neste gerador, o programador poderia criar vários componentes de uma célula de trabalho a partir de primitivas básicas e salvá-las em bibliotecas. Quando fosse necessária a construção de uma nova célula, o programador simplesmente iria arrastar e soltar os componentes das bibliotecas para a área de trabalho, posicionando-os conforme o ambiente real.

A implementação de um controle de vários robôs, cada um conectado a uma máquina. Em várias aplicações, é necessário mais do que um robô para realizar a tarefa especificada. Cada robô é conectado em um computador e se faz necessária uma comunicação entre as máquinas para sincronização e coordenação entre eles.

BIBLIOGRAFIA

- [ADV 99] Advanced Design Inc. <http://www.robix.com>, Maio de 2000.
- [GRO 88] GROOVER, Mikell P. [et al]. Robótica Tecnologia e Programação. São Paulo: McGraw-Hill, 1988.
- [MCK 95] McKERROW, Phillip John. Introduction to Robotics. 3^a ed. Wollongong: Addison-Wesley, 1995.
- [MCW 86] McWATERS, J. F., STORR, A.(Ed.).Off-line Programming of Industrial Robots. Amsterdam: North-Holland, 1986.
- [NOF 85] NOF, Shimon Y.(Ed.). HandBook of Industrial Robotics. West Lafayette: John Willey & Sons, 1985.
- [ROB 95] ROBIX RCS-6 User Guide and Project Book, Advanced Design, Inc., 1995.
- [SIL 99] Silicon Graphics Inc. <http://www.sgi.com>, Maio de 2000.
- [THE 99] The Industry's Foundation for High Performance Graphics. <http://www.opengl.com>, Maio de 2000.
- [WOO 97] WOO, Mason; Neider, Jackie; Davis, Tom. OpenGL Programming Guide. 2^a ed. Addison-Wesley, 1997.
- [WRI 97] Workshop on Intelligent Robotics (1997, aug.4-5: Brasília, DF); Proceedings, Porto Alegre: SBC, 1997.