

# Estudo e Aplicação da Biblioteca de Construção de Jogos Allegro no Controle de um Robô Manipulador Tipo SCARA

Raquel M. Kawamoto , André L. Delai , José R. de Oliveira (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)

Faculdade de Engenharia Elétrica e de Computação (FEEC)

Universidade Estadual de Campinas (Unicamp)

Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

raquelkawa@gmail.com

{delai,jro}@dca.fee.unicamp.br

**Abstract** – The objective of this work is the study and application of the allegro C/C++ library to generate a man/machine software interface to provide a didactical tool for a SCARA robot programming. This Rhino robot model can be controlled by IBM/PC computer via RS-232 interface using a dedicated command set that allows to move the robot joints, read arm position information, record a robot programming and operate the system using joysticks.

**Keywords** – Rhino, robot, allegro, SCARA, control, interface, RS-232.

## 1. Introdução

A configuração de robô mais comumente utilizada em indústrias são os robôs manipuladores de cadeia serial, cujo sistema de controle permite governar o movimento de seus membros manualmente (o operador tem o controle direto dos movimentos) ou na forma de sequência variável (quando é possível alterar algumas das características do ciclo de trabalho) [3].

Neste contexto de robótica, o presente trabalho de iniciação científica visa o estudo de um modelo de robô manipulador tipo SCARA (*Selective Compliance Assembly Robot Arm*) [4] a partir do desenvolvimento de uma interface homem/robô de meta didática. A figura 1 exibe uma imagem do sistema Rhino SCARA onde o trabalho prático é realizado.



Figura 1. Rhino SCARA (robô e controlador).

O projeto é uma atualização do antigo sis-

tema Rhino e tem como proposta algumas melhorias tais como a operação através de *joysticks* com interface USB (*Universal Serial Bus*), armazenamento de programa de ensino em memória não volátil, um sistema de operação mais intuitivo e a execução em sistemas operacionais contemporâneos (GNU/Linux, Windows XP, Windows 7). Como produto final, espera-se obter um software didático (plataforma IBM-PC) para operação do robô Scara a ser usado no LSMR (Laboratório de Sistemas Modulares Robóticos) da FEEC/Unicamp.

## 2. Proposta

O sistema em desenvolvimento pode ser basicamente dividido em duas partes. A primeira contempla os dispositivos de entrada pertinentes ao usuário (*joystick*, *mouse* e teclado). A segunda está relacionada a comunicação via interface computador/robô (comunicação serial assíncrona). O trabalho utiliza como ferramenta a linguagem de programação C para construção dos algoritmos de interfaceamento e uso das bibliotecas Allegro e de acesso ao *hardware*, dada sua flexibilidade e amplo uso na indústria. C é tratada como uma linguagem de médio nível porque combina elementos de linguagem de alto nível com a funcionalidade da linguagem assembly [5]. O panorama geral do sistema pode ser visto na Figura 2, na qual é possível observar a hierarquia de acesso usada pelos componentes do *software*.

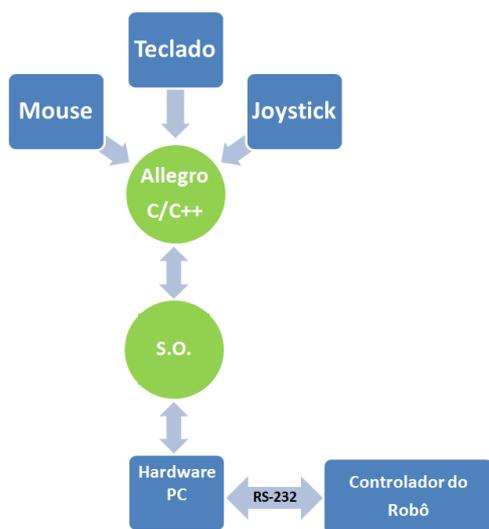


Figura 2. Diagrama Geral do Sistema.

## 2.1. A Biblioteca Allegro

A biblioteca Allegro normalmente é usada na programação de jogos, porém seus recursos e relativa facilidade de uso a torna possível de ser utilizada em outras áreas por ser nativa à linguagem C/C++, além de ser multi-plataforma, *open-source* e gratuita. A Allegro possui funções gráficas, matemáticas de ponto fixo e flutuante, funções de gerenciamento de arquivo, arquivo compacto e uma GUI (*Graphical User Interface*) e apresenta facilidade em adicionar dispositivos de entrada (teclado, *mouse* e *joystick*), além da pluralidade de sistemas operacionais suportados - como, por exemplo, DOS, UNIX, Windows, MacOSX - e da facilidade de ser configurada para qualquer tipo de compilador [1].

O compilador escolhido para a instalação da biblioteca foi o GCC em ambiente Dev C++, devido à vantagem de a instalação poder ser realizada através do Gerenciador de Pacotes da própria IDE (Integrated Development Environment).

Na construção do programa as subrotinas são inicializadas pela função `int allegro_init()`, que inicializa os recursos da Allegro, as variáveis globais e reserva de memória, sendo encerrada pela função `void allegro_exit()`, a qual retorna para o modo texto, removendo quaisquer rotinas dos dispositivos de entrada que tenham sido instaladas. Há também a função `END_OF_MAIN()`, que é uma macro para manter compatibilidade entre plataformas. As funções para qualquer dispositivo de entrada são relativamente simples.

Para um *joystick*, primeiro inicializa-se o dispositivo pela chamada da função `int install_joystick(JOY_TYPE_AUTODETECT)`, e toda vez que o *joystick* for usado e for necessário atualizar os seus valores e detectar eventos do direcional ou botões é necessária a chamada da função `int poll_joystick()`.

## 2.2. Robô Rhino SCARA

O modelo SCARA possui cinco servo-motores CC (Corrente Contínua), nomeados por A, B, C, D e E, cada um deles responsável pelo movimento de uma junta e equipado com um *encoder* óptico incremental de pista dupla que fornece a posição e o sentido da junta ao controlador. Todo o sistema possui quatro GDL (graus de liberdade), como observado no modelo 3D da Figura 3. Maiores informações podem ser obtidas no manual do proprietário [4].

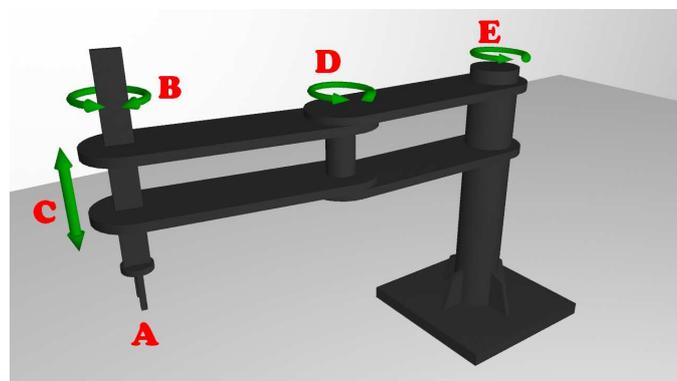


Figura 3. Modelo 3D do robô Rhino SCARA.

A configuração ou programação dos robôs é realizada conectando-se um PC a caixa controladora, que pode ser vista na Figura 1, a um computador através de uma porta serial RS-232 [2].

O robô interpreta comandos definidos pelo fabricante onde cada comando é enviado ao controlador através de palavras de sete bits (sistema ASCII de caracteres). A taxa de transferência padrão é de 9600 bps (bits por segundo), paridade par, 2 bits de parada e 7 bits de dados. Todos os comandos usados no controle do robô podem ser vistos na Tabela 1 [4].

Comando	Descrição
<return>	Envia comando
?	Retorna distância faltante para junta
A-H	Registra movimento do motor
I	Lê chaves limite C-H
J	Lê chaves limite A-B e <i>inputs</i> (1-4)
K	Lê <i>inputs</i> (5-8)
L	Liga porta auxiliar 1
M	Desliga porta auxiliar 1
N	Liga porta auxiliar 2
O	Desliga porta auxiliar 2
P	Registra bits de <i>output</i> em alto
Q	Reset do controlador
R	Registra bits de <i>output</i> em baixo
X	Aborta comando enviado ao motor

Tabela 1. Lista de comandos do robô Rhino.

Um exemplo de comando para movimentação da junta controlada pelo motor E seria o envio de **E+50 <return>**, o que faria a junta se deslocar 50 *counts* do *encoder* no sentido anti-horário. Para deslocamento de mesma proporção no sentido horário tem-se **E-50 <return>**. Uma deficiência deste sistema está no fato do controlador trabalhar unicamente com registradores de oito bits para o movimento das juntas, o que daria o máximo de 127 *counts* para cada movimento (o oitavo bit é usado para sinal). Isso não possibilita atingir toda a extensão de movimento da junta em um único comando, sendo necessário criar rotinas de software que contornem esse problema, criando grandes movimentos através da soma de pequenos deslocamentos.

### 3. Testes e Resultados

Os primeiros testes foram realizados com a biblioteca Allegro e dispositivos de I/O (*mouse*, teclado e *joystick*) para familiarização com os recursos. Uma série de pequenos programas foi desenvolvida criando estruturas de dados e também adaptando os recursos de I/O ao problema de manejo do sistema. Isso permitirá posteriormente gravar a trajetória das juntas, permitindo movimentos mais complexos por parte do robô (abordagem de aprendizado/programação por tutor).

A Figura 4 exhibe a janela de um dos programas realizados em Allegro para interação com um *joystick* de oito botões e direcional digital. Basicamente o usuário gera trajetórias retilíneas usando o

direcional do *joystick* e botões de comando para criação e deleção de novas retas. O programa também calcula as distâncias e as armazena em um vetor alocado dinamicamente e, ao final, em um arquivo em disco para posterior reconstrução das trajetórias.

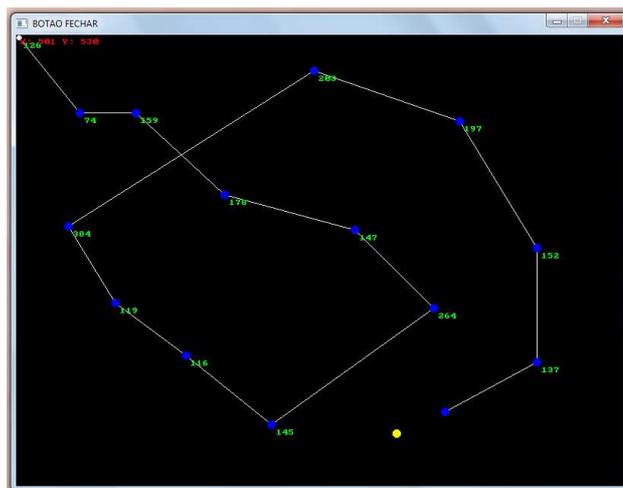
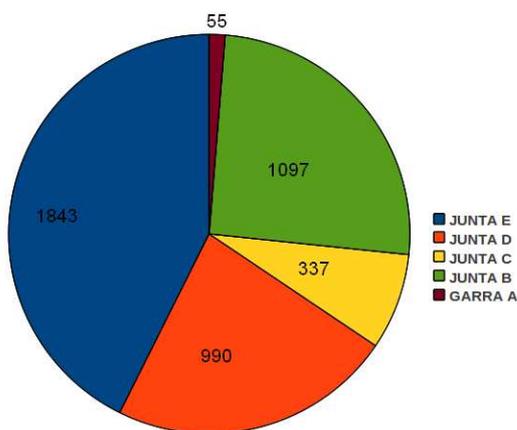


Figura 4. Janela de programa para teste de joystick.

Um problema em relação aos testes de programação do robô foi o mapeamento do intervalo de *counts* para cada junta, já que não foi possível encontrar essa informação em nenhuma documentação.

Para obter tais dados foi então necessária a criação de um programa que movimentasse as juntas do robô e calculasse a distância máxima para cada junta do braço robótico, realizando um processo de engenharia reversa. O programa basicamente mapeava cada junta 20 vezes obtendo ao fim de cada iteração um número de *counts* correspondente ao deslocamento total da junta. Cada teste retornava com pequenas diferenças devido a folga nas juntas do robô e possíveis erros de contagem. Com base nisso os dados foram submetidos ao procedimento de erro médio quadrático para obtenção de valores mais aproximados. As valores e proporções dos movimentos calculados para as juntas podem ser observados no gráfico da Figura 5.



**Figura 5. Intervalo de movimento das juntas (em counts).**

Esses dados serão essenciais para a continuidade do trabalho de programação do software que permitirá acesso de alto nível aos comandos do manipulador SCARA, isso porque tais informações implicam em condições de prever a disposição espacial das juntas do braço dentro de um erro aceitável para o objetivo didático.

O trabalho tem conclusão prevista para agosto de 2011, sendo que as próximas etapas serão a união de todos os recursos e dados obtidos em um único programa. Esse novo programa permitirá ao usuário realizar movimentos isolados e interpolados (pseudo-paralelos) nas juntas, leituras de posição e acesso a sinais externos ao controlador, o que permitirá trabalhar cooperativamente com robôs externos ao sistema.

#### 4. Conclusões

O trabalho segue com progresso semanal e dentro do cronograma estipulado para proposta de iniciação científica. Apesar das limitações do controlador do sistema Rhino SCARA, os dados obtidos e os experimentos realizados com os programas desenvolvidos até agora mostram a viabilidade de atualização e geração de uma nova ferramenta de *software* para o controle do robô manipulador. Isso trará benefícios tais como interfaces mais amigáveis, programação mais simples e intuitiva, arquivamento dos programas de movimentação criados pelo usuário e suporte a sistemas operacionais modernos.

Dessa forma o objetivo da proposta vai sendo alcançado, que consiste em proporcionar um

melhor sistema destinado ao aprendizado de robótica envolvendo robôs manipuladores de cadeia serial modelo SCARA.

#### Referências

- [1] Allegro.cc - game developing community network. <http://www.allegro.cc>. (acessado em 18/03/2011).
- [2] The rs232 standard - a tutorial with signal names and definitions. [http://www.camiresearch.com/Data\\_Com\\_Basics/RS232\\_standard.html](http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html). (acessado em 18/03/2011).
- [3] Nilson Moutinho dos Santos. Classificação geral dos robôs. departamento de informática da universidade de maringá. <http://www.din.uem.br/ia/robotica/classif.htm>. (acessado em 18/03/2011).
- [4] Rhino Robots Inc. *The XR-3 and the SCARA Manual*. Rhino Robots Inc., Champaign, Illinois, 1990.
- [5] Herbert Schildt. *C Completo e Total*. Makron Books, São Paulo, 3 edition, 1996.