

## ATUALIZAÇÃO DE HARDWARE E SOFTWARE DE UM ROBÔ INDUSTRIAL

AURÉLIO DE LIMA E SILVA, LEONARDO POUBEL ORENSTEIN, FERNANDO LIZARRALDE  
ANTONIO C. LEITE, GABRIEL GLEIZER\*

\*Programa de Engenharia Elétrica / COPPE  
Universidade Federal do Rio de Janeiro  
Rio de Janeiro, RJ, Brasil

Email: aurelio@coep.ufrj.br, orenstein@peq.coppe.ufrj.br, fernando@coep.ufrj.br,  
toni@coep.ufrj.br, ggleizer@coep.ufrj.br

**Abstract**— This paper addresses the modernization process of an ABB IRB2000 industrial robot, describing the hardware and software retrofitting procedures, the wiring revision and the assembly of a new control panel. The motors, brakes and sensors are linked with servo drives, which are connected to a personal computer through a CAN-Bus network. The control algorithm is implemented over a Simulink/Matlab platform, with the development of a communication library and a man-machine interface. Experimental results are presented to validate the developed control architecture.

**Keywords**— Robotics, Retrofitting, CAN Networks, Kinematic Control

**Resumo**— Este artigo descreve o processo de modernização de um robô industrial IRB2000, apresentando os procedimentos de atualização de *hardware* e *software*, a revisão do cabeamento e a construção de um novo painel de controle. Os motores, freios e sensores são ligados a servoconversores comerciais, que são conectados a um PC através de uma rede CAN-Bus de dados. O algoritmo de controle é implementado através de uma plataforma Simulink/Matlab, a partir do desenvolvimento de uma biblioteca de comunicação e uma interface homem-máquina. Resultados experimentais são apresentados para validar a arquitetura de controle desenvolvida.

**Palavras-chave**— Robótica, *Retrofitting*, Redes CAN, Controle Cinemático

### 1 Introdução

Uma máquina ou equipamento com tecnologia obsoleta podem ser atualizados ou reformados através da aplicação de técnicas modernas de automação e da utilização de componentes e dispositivos sofisticados. Este processo, conhecido como *retrofitting* ou modernização, é muitas vezes uma solução para empresas que desejam aumentar a vida útil das máquinas, mantendo suas características periféricas com perfeito estado de conservação mecânica (Ribeiro et al., 2007).

Em grandes indústrias automatizadas é comum a substituição de robôs antigos por novos modelos a fim de tornar o sistema mais confiável, aumentando a produtividade e reduzindo os períodos de inatividade e os riscos operacionais. Devido ao baixo custo, os robôs descartados tornam-se uma alternativa viável para universidades, laboratórios e centros de pesquisa, que atuam na área de robótica aplicada. Porém, como estes robôs utilizam tecnologia obsoleta, alguns benefícios associados ao seu emprego para execução de tarefas podem não ser alcançados. Felizmente, as características mecânicas de robôs industriais não vem sofrendo mudanças significativas ao longo do tempo. As principais diferenças entre os modelos antigos e os novos estão relacionadas ao acionamento elétrico das juntas e ao sistema de controle (Lages et al., 2003; Lima II et al., 2004). Neste contexto, através de um processo de modernização tecnológica é possível agregar novas funcionalidades ao sistema robótico tornando a sua arquitetura de controle mais aberta e flexível (Bruyninckx, 2001; Lippiello et al., 2007).

Recentemente, diversos trabalhos foram publicados considerando o problema de atualização tecnológica de robôs industriais obsoletos. Em (Lages et al., 2003), detalha-se o processo de *retrofitting* de um manipulador ASEA-IRB6, compreendendo revi-

são mecânica, atualização elétrica, e substituição de controlador e *software*. Em (Becerra et al., 2004), apresenta-se o *retrofitting* de um robô PUMA 560, juntamente com a interface de *software* e a aplicação de um controle por torque computado. Uma proposta de arquitetura aberta para controle de velocidade e força de um robô industrial REIS-RV15 é descrita em (de Oliveira and Guenther, 2007), onde a comunicação entre computador e robô é feita via USB e o *software* é desenvolvido em linguagem de programação C++. Em (Becerra et al., 2004), a comunicação é feita através de uma placa Q8 com interface PCI, e a lei de controle é implementada em Simulink/Matlab.

No presente trabalho descreve-se os procedimentos de engenharia reversa utilizados para realizar o *retrofitting* de um robô industrial IRB2000 (ABB Robotics), bem como a atualização do *hardware* e o desenvolvimento do *software* para comunicação, configuração *on-the-fly*, controle e gerenciamento da interface com o usuário. Para verificar o funcionamento da arquitetura de controle proposta, uma lei de controle cinemático de posição e orientação para o robô é implementada. Resultados experimentais são apresentados para ilustrar o desempenho e a viabilidade do esquema de atualização proposto.

Uma vez que o robô IRB2000 é acionado por motores síncronos de corrente alternada (CA) trifásicos, o processo de *retrofitting* torna-se uma tarefa desafiadora, pois estes motores necessitam de controladores mais complexos, quando comparados aos empregados em motores de corrente contínua (CC) (Lages et al., 2003; Becerra et al., 2004).

### 2 Metodologia Proposta

O robô IRB2000 operava originalmente através de algoritmos de controle fornecidos pelo fabricante

(ABB Robotics), cuja característica fechada não permitia o desenvolvimento e a implementação de aplicações livres para usuários. Então, para solucionar este problema primeiramente foi realizada a especificação técnica de equipamentos comerciais para acionamento elétrico e controle de motores, a partir do conhecimento dos torques e das correntes máximas de acionamento dos motores do robô. Estes dispositivos são comumente denominados de servoconversores. Os servoconversores escolhidos foram instalados em um quadro de comando, juntamente com dispositivos de partida e proteção necessários para o acionamento dos motores (*e.g.*, disjuntores e contadores).

Em seguida foi realizado um estudo das principais características eletromecânicas dos motores, para configurar os servoconversores corretamente, a fim de substituir o sistema de controle original por uma malha externa de controle de posição ou velocidade, em série com uma malha interna de controle de corrente (Fitzgerald et al., 2003). Isto permitiu realizar ensaios experimentais e estudar todas as possibilidades de operação do robô, sobretudo no que diz respeito a calibração de parâmetros e a flexibilidade de sua integração com outros sistemas de atuação ou de medição. Desta forma, foi possível substituir a plataforma proprietária por uma arquitetura de controle aberta, de fácil intervenção e intercambiabilidade. Além disso, uma interface com o usuário foi desenvolvida no ambiente de programação Matlab (The MathWorks, Inc.) para facilitar a comunicação entre o sistema de controle e o operador. Esta interface foi instalada em um computador, que foi conectado ao sistema de acionamento do robô através de uma rede CAN-Bus de dados.

### 3 Retrofitting do Manipulador

Originalmente, a arquitetura de controle do robô IRB2000 permitia apenas algumas funcionalidades como, por exemplo, a especificação de trajetórias de referência, calibração de parâmetros do robô e utilização de programas pré-existentes. O sistema robótico original é composto por dois componentes principais: uma gabinete de controle e um robô manipulador. Durante o processo de *retrofitting* foi desenvolvido um novo painel de controle, moderno, compacto e mais leve, para substituir o painel original (Figura 1).

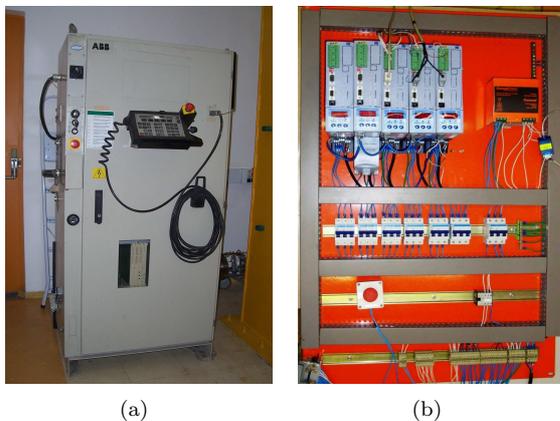


Figura 1: (a) Gabinete de controle original e (b) novo painel de controle.

A arquitetura de controle original do sistema é composta por uma placa de processamento central, responsável pelo controle do manipulador, de onde é enviado o sinal de controle para ser amplificado pela eletrônica de potência e então transmitido para os motores. A realimentação da posição angular é realizada através dos *resolvers* instalados nos eixos de cada motor. O sinal analógico proveniente dos *resolvers* é transmitido até uma placa de conversão analógico-digital (SMB, do inglês *Serial Measurement Board*) instalada na estrutura do robô, que gera um sinal digital transmitido por uma conexão serial ao controlador. Esta arquitetura foi modificada, para fornecer um maior grau de abertura e flexibilidade ao sistema robótico. O novo esquema utiliza servoconversores SCA-05 (WEG S.A.), que são responsáveis pela leitura dos *resolvers* e pelo acionamento e controle de velocidade dos motores. Estes servoconversores são conectados a um computador através de uma rede CAN-Bus de dados e neste computador são calculados os sinais de controle do robô, através de uma interface Simulink/Matlab (vide Figura 2).

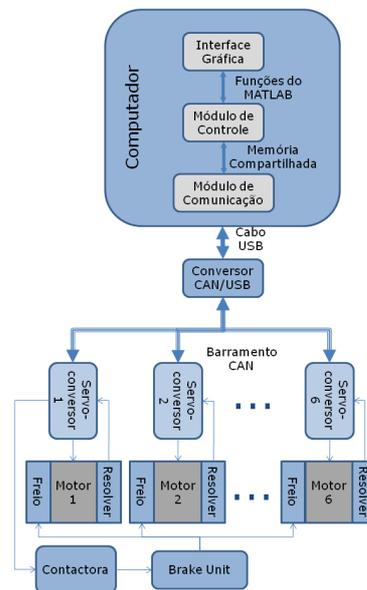


Figura 2: Diagrama em blocos da arquitetura de controle desenvolvida.

#### 3.1 Painel de Controle

O servoconversor possui um *driver* de potência necessário para o acionamento do motor e são responsáveis por algumas das verificações de segurança efetuadas pelo sistema, além de liberar o freio e fazer a leitura do *resolver*. De acordo com os motores utilizados pelo robô, foram especificados os seguintes servoconversores: o modelo SCA-05 0008, com corrente nominal de saída de 8 A para a junta 2, e o modelo SCA-05 0004, com corrente nominal de saída de 4 A para as demais juntas. A alimentação de entrada do servoconversor é trifásica e a faixa de tensão de alimentação é de 220V–230V.

Um circuito de segurança foi desenvolvido para garantir que o robô seja desabilitado através de um botão de emergência quando algum problema de mal fun-

cionamento for detectado pelo usuário. Então, o desligamento acontece automaticamente, quando ocorre um erro em alguma junta, ou via *software* quando for necessário. Este mecanismo foi implementando associando-se uma saída a relé do servoconversores ao seu estado atual, onde estes relés estão conectados em série com o botão de emergência criando uma lógica *and* entre todos os sinais habilitadores do sistema. Esta lógica é conectada a uma entrada digital de cada servoconversor, e caso um servoconversor entre no estado desabilitado, ou o botão de emergência seja ativado, todos os servoconversores são desabilitados simultaneamente. Além disso, outros equipamentos elétricos são utilizados para assegurar o funcionamento do robô como, por exemplo, uma fonte de alimentação 24V CC – 10 A (Conexel), um contato CW07-10E (WEG S.A.) disjuntores MDW-C4 e MDW-C32 (WEG S.A) e uma sinaleira para indicar o estado do sistema. Quando o servoconversor está habilitado é enviado um sinal de liberação para os freios e uma luz vermelha é acesa na sinaleira.

### 3.2 Manipulador

O manipulador é composto por seis juntas de revolução, interligadas por elos e possui seis graus de mobilidade, similar a um braço humano. A estrutura completa forma uma cadeia cinemática aberta e esta estrutura é denominada *braço antropomórfico com punho esférico*, onde as três primeiras juntas são responsáveis pelo movimento do braço, enquanto que as três últimas juntas realizam o movimento do punho.



Figura 3: Vista lateral do robô IRB2000.

O movimento do robô é comandado por motores síncronos CA trifásicos equipados com freios eletromagnéticos, e a posição das juntas é obtida através da leitura de *resolvers* acoplados aos eixos dos motores. Atualmente, o robô encontra-se instalado de forma invertida, ou seja, a sua base está fixada a um suporte acima do solo e o manipulador esta suspenso (vide Figura 3). Uma representação esquemática do robô, juntamente com a localização dos motores e os seus respectivos eixos, é apresentada na Figura 4 (ABB Robotics Products AB, 1993).

O robô possui limitações de velocidade e restrições físicas para os ângulos das juntas, de acordo com seu projeto estrutural, que podem ser consultadas na

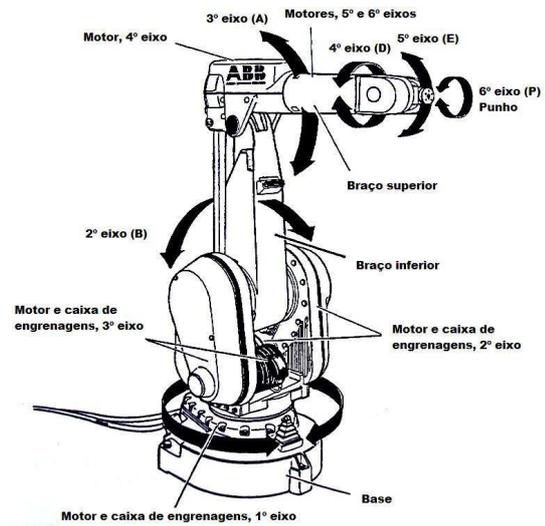


Figura 4: Desenho esquemático do robô IRB2000.

Tabela 1 (ABB Robotics Products AB, 1993). Essas informações são importantes para determinar a relação entre a leitura dos *resolvers* e os ângulos das juntas. O peso do manipulador é 350 kg e capacidade de carga máxima ou *payload* é 10 kg.

Eixo	Posição [grau]	Velocidade [grau/s]
1	± 179.9	115
2	-110 a 100	115
3	± 60	115
4	± 200	280
5	± 120	300
6	± 300	300

Tabela 1: Limites de posição e velocidade das juntas.

#### 3.2.1 Motores

O manipulador possui seis motores de corrente alternada trifásicos do tipo PMSM (do inglês, *permanent magnet synchronous motor*), que são utilizados para movimentar as juntas. Os modelos dos motores, fabricados pela ELMO MC, o torque e a corrente nominais são apresentados na Tabela 2. De acordo com o catálogo do fabricante a seguinte designação para o modelo do motor é utilizada: nome da série (PS), tamanho do motor (130), número de pólos (6 ou 4) e comprimento ativo (30, 50 ou 75). Este tipo de motor apresenta algumas vantagens quando comparado com o motor assíncrono como, por exemplo, rápida resposta dinâmica, alta relação torque e potência por volume, relação linear de torque *versus* corrente, alto fator de potência e maior eficiência (Fitzgerald et al., 2003).

Para controlar os motores, o servoconversor utiliza a estratégia de Controle Vetorial ou Controle Orientado pelo Campo (FOC, do inglês *field orientated control*). A estratégia de controle vetorial é uma técnica adotada para simplificar o controle de motores CA (síncronos ou de indução), que permite o desacoplamento entre o conjugado eletromagnético (torque) e o fluxo, semelhante à máquina CC com excitação em separado, melhorando as suas características dinâmicas quando comparadas com a resposta dos aciona-

Modelo	Junta	Torque (Nm)	Corrente (A)
PS 130/6-30-P 3278	1, 3	4.0	4.0
PS 130/6-50-P 3279	2	4.0	7.0
PS 130/4-75-P 3294	4, 5, 6	1.1	1.5

Tabela 2: Identificação dos motores do robô IRB2000.

mentos convencionais (Fitzgerald et al., 2003). Para utilizar o controle vetorial, a partir dos servoconversores, é necessário obter os parâmetros eletromecânicos (*e.g.*, constante de torque, momento de inércia, resistências e indutâncias) e o *offset* de comutação de cada motor (Bose, 1996; Pizzino, 2008). Geralmente, os parâmetros dos motores são disponibilizados na folha de dados dos motores, enquanto que o *offset* de comutação é um parâmetro específico da instalação do *resolver* no eixo do motor, e são fornecidos pelo fabricante do robô. Finalmente, é necessário realimentar a malha de controle com a posição angular obtida pelos *resolvers*.

Durante a realização de testes com o robô, foi observado um acoplamento nas duas últimas juntas do braço. Este acoplamento é devido a construção mecânica das reduções que conectam os motores às juntas e pode ser representado por

$$\theta = A_m \theta_m, \quad (1)$$

onde  $\theta \in \mathbb{R}^6$  é a posição angular das juntas,  $\theta_m \in \mathbb{R}^6$  é a posição angular dos eixos dos motores, obtida através da leitura do *resolver* e  $A_m \in \mathbb{R}^{6 \times 6}$  é a matriz de acoplamento, onde o elemento  $a_{ij}$  é a relação entre a posição angular do eixo do motor  $i$  para a posição angular da junta  $j$ . Os seguintes valores foram obtidos experimentalmente calculando-se a razão entre as posições limites definidas na Tabela 1 e a leitura da posição obtida do servoconversor nos mesmos pontos, para cada junta:

$$A_m = \begin{bmatrix} \frac{1}{74} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{43} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{75} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{40} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{35} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{35} & \frac{1}{35} \end{bmatrix}. \quad (2)$$

O motor possui também um freio eletromagnético em cada uma de suas juntas, que são desabilitados em conjunto por uma placa instalada no manipulador, denominada Unidade de Liberação do Freio (BRU, do inglês *Brake Release Unit*). Estes freios são acionados através de solenóides individuais que operam sob uma tensão de 24 V CC e consomem aproximadamente 17 W cada. Para executar este acionamento, o novo painel de controle utiliza uma fonte de alimentação 24 V CC e um contator na configuração normalmente aberta.

### 3.2.2 Resolvers

Para obter a leitura da posição angular dos motores, o robô IRB2000 possui seis *resolvers*, acoplados no eixo de cada motor. Os *resolvers* são do modelo 11-BHW-48WD/F754, trabalham com uma frequência de 4 kHz e são digitalizados pelo servoconversor com uma quantização de 14 bits, de forma que a leitura da posição varia entre 0 e 16383 pulsos. Do ponto de vista

de construção, um *resolver* é considerado como uma pequena máquina elétrica com um rotor e um estator: a bobina está no rotor, enquanto que o estator possui dois enrolamentos separados de 90 graus elétricos um do outro (SIN e COS). Excitando a bobina do rotor com uma tensão senoidal enviada pelo servoconversor (OSC), uma tensão é induzida nos enrolamentos do estator cuja magnitude depende do ângulo de rotação do eixo do motor (Sciavicco and Siciliano, 2000).

Os servoconversores utilizados trabalham apenas com o sinal analógico do *resolver*. Por isso, a estrutura de cabeamento original, que conduzia o sinal analógico dos *resolvers* para a SMB, foi modificada para uma nova configuração, onde agora os sinais analógicos são obtidos diretamente pelos servoconversores. Para garantir uma alta relação sinal/ruído, as conexões já existentes no sistema foram utilizadas na estrutura de cabeamento, e novos cabos foram criados apenas quando necessário.

## 4 Projeto do Controlador

Nesta seção apresenta-se o projeto de uma lei de controle cinemático de posição e orientação para o robô, a fim de avaliar o funcionamento da arquitetura de controle desenvolvida. Considerando o problema de controle cinemático, o movimento do robô pode ser descrito por

$$\dot{\theta}_i = u_i, \quad i = 1, \dots, n, \quad (3)$$

onde  $\theta_i$  e  $\dot{\theta}_i$  são a posição angular e a velocidade angular da  $i$ -ésima junta respectivamente, e  $u_i$  é o sinal de controle de velocidade aplicado ao  $i$ -ésimo *drive* do motor da junta.

Neste projeto, utiliza-se a convenção *standard* de Denavit-Hartenberg (DH) para determinar a função de cinemática direta do robô (Sciavicco and Siciliano, 2000). Os ângulos das juntas são obtidos através da leitura dos *resolvers* e os parâmetros DH são calculados a partir do conhecimento da estrutura geométrica do robô (ABB Robotics Products AB, 1993).

### 4.1 Cinemática direta e cinemática diferencial

Considere  $p \in \mathbb{R}^3$  a posição do efetuador com respeito a base do robô, expresso no sistema de coordenadas da base  $\{B\}$ . Agora, considere  $R$  a matriz de rotação do sistema de coordenadas do efetuador  $\{E\}$  em relação ao sistema de coordenadas da base  $\{B\}$  e  $q = [q_s \ q_v]^T \in \mathbb{R}^4$  a representação de quaternion unitário para  $R$ , onde  $q_s \in \mathbb{R}$  e  $q_v \in \mathbb{R}^3$  são a parte escalar e vetorial do quaternion respectivamente, sujeitas a restrição  $\|q\| = 1$ . Neste contexto, a postura do robô é dada pelo mapeamento de cinemática direta

$$X = \begin{bmatrix} p \\ q \end{bmatrix} = k(\theta), \quad (4)$$

onde  $k(\cdot)$  é uma função vetorial, em geral não linear, que permite a computação das variáveis no espaço operacional a partir do conhecimento das variáveis no espaço das juntas.

A equação de cinemática diferencial pode ser obtida como a derivada temporal da equação de cinemática direta em (4), isto é,

$$\dot{X} = \begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix} = J_A(\theta) \dot{\theta}, \quad J_A(\theta) = \frac{\partial k(\theta)}{\partial \theta}, \quad (5)$$

onde  $J_A$  é o Jacobiano analítico. A velocidade do efetuador  $V$  é relacionada com  $\dot{X}$  por

$$V = \begin{bmatrix} v \\ \omega \end{bmatrix} = J_R(q) \dot{X}, \quad (6)$$

onde  $v \in \mathbb{R}^3$  é a velocidade linear,  $\omega \in \mathbb{R}^3$  é a velocidade angular e  $J_R$  é o Jacobiano da representação.

A partir de (5) e (6), a relação entre a velocidade do efetuador e a velocidade das juntas é dada por

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \underbrace{\begin{bmatrix} J_{11}(\theta) & 0 \\ J_{21}(\theta) & J_{22}(\theta) \end{bmatrix}}_{J_{R(q)} J_A(\theta)} \begin{bmatrix} \dot{\theta}_b \\ \dot{\theta}_p \end{bmatrix} \quad (7)$$

ou, em uma forma mais compacta por

$$V = J(\theta) \dot{\theta}, \quad (8)$$

onde  $J(\theta) \in \mathbb{R}^{6 \times 6}$  é o Jacobiano do manipulador,  $J_{11}, J_{21}, J_{22} \in \mathbb{R}^{3 \times 3}$  são submatrizes jacobianas,  $\dot{\theta}_b \in \mathbb{R}^3$  é o vetor de velocidade das juntas 1, 2 e 3, correspondentes ao braço, e  $\dot{\theta}_p \in \mathbb{R}^3$  é o vetor de velocidade das juntas 4, 5 e 6, correspondentes ao punho.

Então, a partir de (8) e considerando  $\dot{\theta}$  como a entrada de controle  $u$ , obtém-se o seguinte sistema de controle:

$$V = J(\theta) u, \quad u = \begin{bmatrix} u_p \\ u_o \end{bmatrix}, \quad (9)$$

onde  $u_p \in \mathbb{R}^3$  é o sinal de controle de posição e  $u_o \in \mathbb{R}^3$  é o sinal de controle de orientação.

#### 4.2 Controle cinemático

Seja  $p_d(t)$  a posição desejada *variante no tempo* e  $R_d$  a orientação desejada *constante* para o efetuador do robô. A partir das equações de cinemática direta e cinemática diferencial, pode-se projetar uma lei de controle cinemático para alcançar simultaneamente o seguinte objetivo de controle de posição

$$p \rightarrow p_d(t), \quad e_p = p_d - p \rightarrow 0,$$

e de orientação

$$R \rightarrow R_d, \quad R_q = R R_d^T \rightarrow I,$$

onde  $e_p \in \mathbb{R}^3$  é o erro de posição e  $R_q \in SO(3)$  é o erro de orientação, ambos expressos com respeito ao sistema de coordenadas da base  $\{B\}$ . Considere  $e_q = [e_{q_s} \ e_{q_v}^T]^T$  a representação de quaternion unitário para  $R_q$ , tal que  $e_q = q * q_d^{-1}$  onde  $q_d$  é a representação de quaternion unitário para  $R_d$  e “\*” denota o operador de multiplicação de quaternions. Note que,  $e_q = [1 \ 0^T]^T$  se e somente se as matrizes de rotação  $R$  e  $R_d$  estão alinhadas (Wen and Kreutz-Delgado, 1991).

A partir da abordagem de controle cinemático (3), a seguinte lei de controle é proposta:

$$u_p = J_{11}^{-1}(K_p e_p + v_d), \quad (10)$$

$$u_o = J_{22}^{-1}(K_o e_{qv} - J_{21} \dot{\theta}_b), \quad (11)$$

onde  $K_p \in \mathbb{R}^{3 \times 3}$  é a matriz de ganho de posição,  $K_o \in \mathbb{R}^{3 \times 3}$  é a matriz de ganho de orientação e  $v_d \in \mathbb{R}^3$  é a velocidade linear desejada. Então, a partir de uma escolha apropriada de  $K_p$  e  $K_o$  como matrizes definidas positivas, pode-se mostrar que as leis de controle (10) e (11) garantem a estabilidade assintótica do erro de posição e de orientação do robô, respectivamente (Sciavicco and Siciliano, 2000).

## 5 Projeto do Software

O projeto do *software* para comunicação, controle, configuração e interface com o usuário foi desenvolvido baseado em uma arquitetura de processamento centralizada (vide Figura 2). Os servoconversores responsáveis pelo acionamento dos motores, leitura dos *resolvers* e outras funcionalidades são conectados a uma rede CAN-Bus de dados e comunicam-se através do protocolo CANopen (Pfeiffer et al., 2008).

### 5.1 Comunicação

No módulo de comunicação, um PC funciona como mestre da rede, requisitando dados relacionados ao controle do robô como, por exemplo, fração de volta, número de voltas e velocidade do motor, enviando os valores de referência de velocidade e gerenciando os estados dos dispositivos na rede.

Para realizar a operação do sistema é necessário enviar a cada tempo de amostragem um pacote de sincronização (11 bits), seis pacotes de guarda (66 bits), 6 Objetos de Dados do Protocolo (PDOs, do inglês *Protocol Data Objects*) de transmissão com 8 bytes de dados (450 bits) e 6 PDOs de recepção com 2 bytes de dados (162 bits). Somando os tamanhos dos pacotes e multiplicando pela taxa de amostragem utilizada de 1 kHz obtemos uma taxa de transferência mínima de 689 kbit/s. Como a rede CAN permite taxas de transmissão de 500 kbit/s ou 1 Mbit/s, a taxa utilizada foi de 1 Mbit/s. Por medida de segurança, se a comunicação entre o PC e os servoconversores for interrompida por mais de 10 ms os servoconversores são desabilitados e os freios são acionados. O tempo de guarda foi escolhido de modo que as juntas não se movessem mais de 3 graus no caso de falha de comunicação. Para conectar o PC ao barramento CAN foi utilizado o adaptador CANUSB (Lawicel AB). A partir da biblioteca para comunicação via CAN fornecida pelo fabricante foi criada uma biblioteca para realizar a comunicação utilizando o protocolo CANopen.

O módulo de comunicação realiza a conexão entre os servoconversores e a camada superior responsável pelo controle. Inicialmente os servoconversores são configurados para atuar na rede. Após a etapa de configuração o aplicativo de comunicação espera por conexões do módulo de controle. Uma vez finalizada a operação do sistema o aplicativo de comunicação volta a esperar por conexões do módulo de controle.

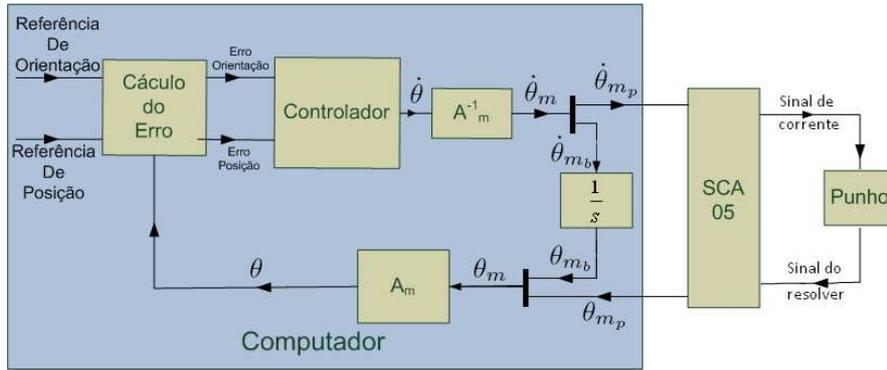


Figura 5: Diagrama em blocos do esquema de controle cinemático.

5.2 Módulo de controle

O módulo de controle foi desenvolvido em Simulink/Matlab (The MathWorks Inc.) pois nesta plataforma é possível projetar algoritmos de controle rapidamente através de *scripts* e blocos de suas bibliotecas. Para executar os modelos em tempo real foi utilizado o QuaRC (Quanser Inc.) por ser totalmente integrado ao Simulink e possuir uma biblioteca com diversas ferramentas como, por exemplo, funções para comunicação entre processos utilizando *Universal Resource Identifier* (URI) e blocos para a aquisição do sinal de orientação de um Wiimote (ou Wii Remote), o controlador do Nintendo Wii (Nintendo Co. Ltd.). O protocolo utilizado para realizar a comunicação entre processos foi o de Memória Compartilhada por ter o menor tempo de execução.

Para executar o controle do robô foi desenvolvida uma biblioteca no Simulink com blocos personalizados para modularizar a implementação das leis de controle (Figura 6). Estes blocos consideram as características do robô como, por exemplo, a sua região de trabalho, a limitação de velocidade das juntas e a redução dos motores. Foram criados também blocos para calcular as cinemáticas direta e diferencial do robô IRB2000 e blocos das juntas, que são responsáveis por receber os dados dos motores e enviar as referências para a camada de comunicação. Os detalhes dessa biblioteca estão registrados em (de Lima e Silva Jr, 2010).

5.3 Interface com o usuário

A interface homem-máquina (IHM), desenvolvida no Matlab, permite ao usuário realizar tarefas simples com o robô como, por exemplo, o controle independente das juntas, o controle da posição e orientação do efetuador e calibração de parâmetros (vide Figura 7). A IHM foi projetada para ser amigável ao usuário e permite a visualização de informações importantes para a operação do robô como, por exemplo, posição e velocidade das juntas, corrente dos motores, orientação e posição do efetuador e estado do sistema. A taxa de atualização dos dados da IHM é de 0.1 Hz para não comprometer o desempenho do algoritmo de controle.

Uma das novas funcionalidades que foi implementada devido a abertura da arquitetura de controle é o controle da orientação do punho do robô, por um dispositivo Wiimote. O Wiimote comunica-se com o computador via protocolo Bluetooth e o bloco do

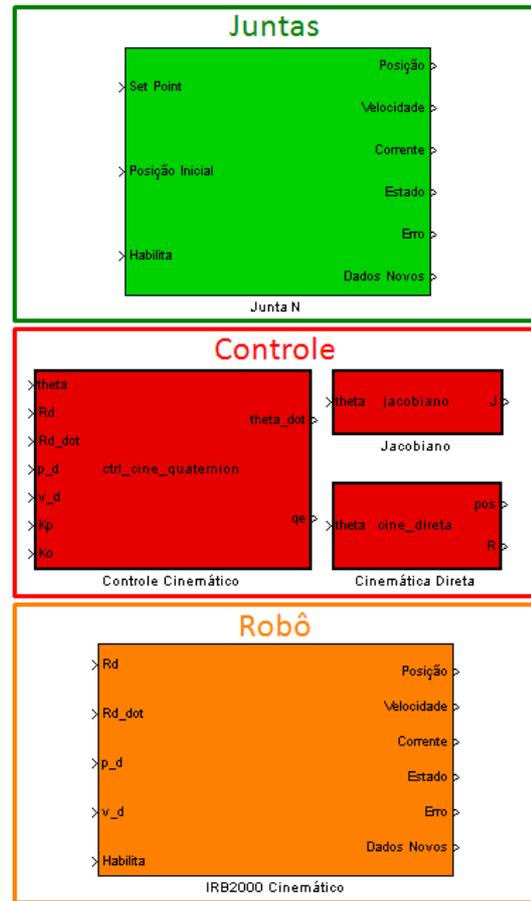


Figura 6: Biblioteca do Simulink para o robô IRB2000.

QuaRC responsável pela aquisição de sinal retorna o estado de todos os botões e a aceleração da gravidade no referencial do controle. O sinal da gravidade é medido através de acelerômetros e o sinal obtido somente permite calcular o ângulos de *roll* e *pitch* do *joystick*. O ângulo de *yaw* é gerado pelas setas direcionais para a esquerda e direita.

6 Resultados Experimentais

Os experimentos foram realizados através de uma interface desenvolvida em Simulink/Matlab, utilizando a comunicação descrita na seção 5 e método de integração de Euler com frequência de amostragem

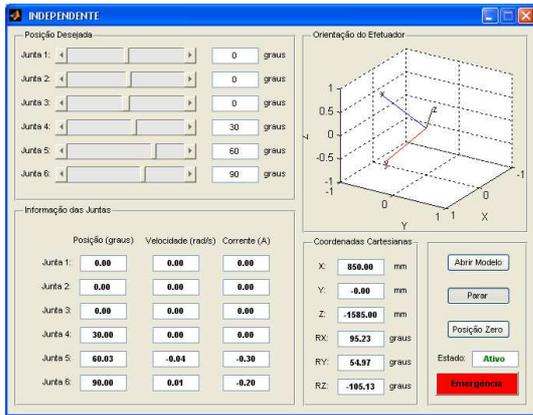


Figura 7: Interface Gráfica

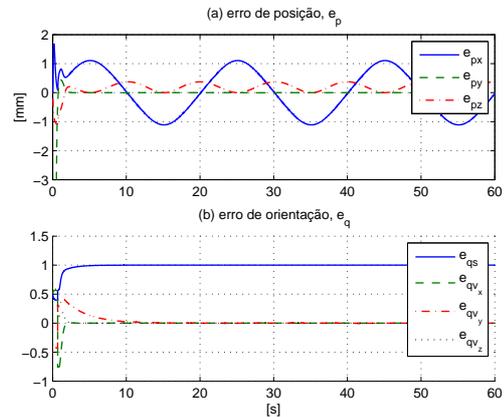


Figura 8: Erros de posição e de orientação.

de  $1\text{ kHz}$ . Esta frequência de amostragem foi escolhida de forma a ser dez vezes mais rápida (Åström and Wittenmark, 1996) que a constante de tempo da malha fechada de posição do motor mais rápido (Orenstein, 2010). Para facilitar a visualização do movimento do robô, a representação de ângulos de Euler ZYZ (Sciavicco and Siciliano, 2000) foi utilizada para descrever a referência de orientação do efetuador.

O experimento consiste na regulação da orientação  $\phi = [\alpha \ \beta \ \gamma]^T$  para  $\phi_d = [-\frac{\pi}{2} \ \frac{\pi}{2} \ \frac{\pi}{2}]^T \text{ rad}$  com o efetuador realizando uma trajetória circular de raio  $300\text{ mm}$ , no plano  $YZ$  centrado no ponto  $p_0 = [850 \ 0 \ -1585]^T \text{ mm}$  e com uma frequência de  $0.05\text{ Hz}$ . A posição inicial para os ângulos de todas as juntas do robô foi zero. A trajetória a ser seguida foi definida de forma a maximizar o movimento das juntas do punho. Os ganhos dos controladores foram:  $K_p = 10\text{ I mm s}^{-1}$  e  $K_o = \text{diag}\{0.4, 0.066, 0.6\} \text{ rad s}^{-1}$ .

Devido às restrições no espaço físico onde o robô está instalado, não foi possível realizar experimentos com segurança atuando nas três primeiras juntas. Para contornar esta limitação, estas juntas foram simuladas em uma estratégia similar à técnica *Hardware In the Loop* (HIL) (Godoy et al., 2008; Isermann, 2008). Na simulação as três primeiras juntas foram representadas como integradores puros. Para implementar o controlador foram utilizadas as malhas de controle de velocidade e de corrente, internas do servoconversor. Os resultados obtidos podem ser observados nas Figuras 8–11, respectivamente.

A posição e a orientação do efetuador foram obtidas a partir dos ângulos das juntas, simulados e medidos através da cinemática direta do robô. Analisando a Figura 8(a), observa-se que o erro máximo de posição foi em torno de  $1\text{ mm}$ . O controle de orientação apresentou resultados satisfatórios e a parte vetorial do quaternion do erro convergiu para um valor próximo de zero, conforme ilustrado na Figura 8(b). A Figura 9 ilustra a convergência dos ângulos de Euler para os valores desejados. A regulação de posição no eixo  $X$  e o rastreamento da trajetória no plano  $YZ$  são apresentadas nas Figuras 10 e 11 respectivamente.

### 7 Conclusões e Trabalhos Futuros

Este trabalho descreve os procedimentos utilizados no processo de atualização tecnológica (ou *re-*

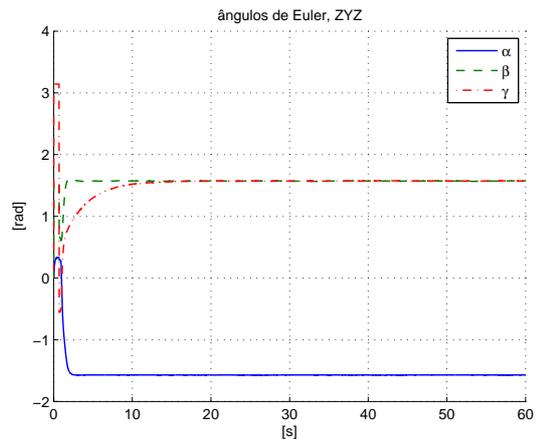


Figura 9: Ângulos de euler ZYZ.

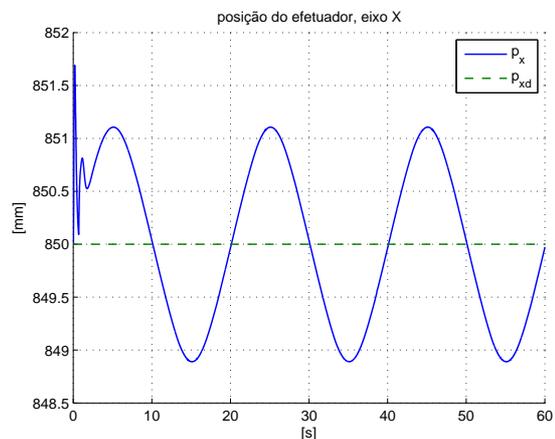


Figura 10: Posição do efetuador no eixo X.

*etrofitting*) de um robô industrial, bem como o *hardware* e o *software* desenvolvidos para controle, configuração e comunicação com um PC executando Simulink/Matlab. O *retrofitting* mostrou-se uma alternativa viável para laboratórios e centros de pesquisa, uma vez que foi possível reutilizar a estrutura mecânica, os sensores e os atuadores do robô para o desenvolvimento de uma arquitetura de controle totalmente aberta.

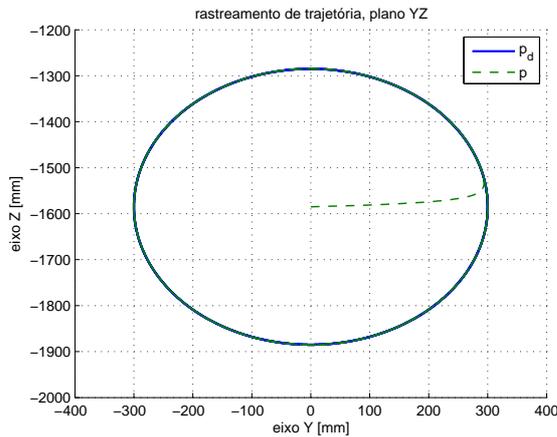


Figura 11: Rastreamento de trajetória no plano YZ

Servoconversores comerciais foram utilizados para realizar a leitura dos *resolvers* e comandar os motores que atuam nas juntas do robô através da estratégia de controle vetorial. Um algoritmo de controle cinemático foi implementado para a regulação da posição e da orientação do efetuador do robô. Resultados experimentais foram apresentados para validar a arquitetura de controle desenvolvida.

As propostas de trabalhos futuros de acordo com as idéias apresentadas aqui são: (i) a construção de uma rede de sensores fotoelétricos acoplada aos freios dos motores para a estabelecer um perímetro de segurança em torno da região de trabalho do robô; (ii) reestruturação do espaço físico onde robô está localizado para possibilitar a atuação em todas as juntas; (iii) desenvolver um módulo de comunicação que envie os dados dos servoconversores através de uma porta TCP/IP, possibilitando o controle do robô remotamente via internet.

### Agradecimentos

Este trabalho foi parcialmente financiado pelo CNPq.

### Referências

- ABB Robotics Products AB (1993). *Product Manual IRB 2000 M93A*.
- Åström, K. J. and Wittenmark, B. (1996). *Computer-Controlled Systems: Theory and Design*, 3rd edn, Prentice Hall.
- Becerra, V. M., Cage, C. N. J., Harwin, W. S. and Sharkey, P. M. (2004). Hardware retrofit and computed torque control of a puma 560 robot updating an industrial manipulator, *IEEE Control Systems Magazine* **24**(5): 78–82.
- Bose, B. K. (1996). *Power Electronics and Variable Frequency Drives - Technology and Applications*, 1st edn, Wiley-IEEE Press.
- Bruyninckx, H. (2001). Open robot control software: the OROCOS project, *IEEE International Conference on Robotics & Automation*, Piscataway, NJ, pp. 2523–2528.

- de Lima e Silva Jr, A. (2010). Desenvolvimento de um sistema de controle em rede utilizando CAN e uma interface homem-máquina para um robô industrial, *Projeto final de graduação*, Departamento de Engenharia de Controle e Automação, UFRJ.
- de Oliveira, A. S. and Guenther, R. (2007). Uma proposta de arquitetura aberta para o controle de posição e força em robôs manipuladores, *8th Simpósio Brasileiro de Automação Inteligente*, Florianópolis, SC.
- Fitzgerald, A. E., Kingsley Jr, C. and Umans, S. D. (2003). *Electric Machinery*, 6th edn, McGraw-Hill, Inc.
- Godoy, E. P., Inamasu, R. Y. and Porto, A. J. V. (2008). Proposta de Utilização de Hardware-in-the-Loop no Desenvolvimento de Sistemas de Controle via Redes CAN, *VIII Conferência Internacional de Aplicações Industriais*, Poços de Caldas, MG.
- Isermann, R. (2008). Mechatronic systems – Innovative products with embedded control, *Control Engineering Practice* **16**(1): 14–29.
- Lages, W. F., Henriques, R. V. B. and Bracarense, A. Q. (2003). Arquitetura aberta para retrofitting de robôs, *Manet Notes Workshop*, Bragança Paulista, SP.
- Lima II, E. J., Torres, G. C. F., Castro, C. A. C., Bracarense, A. Q., Henriques, R. V. B. and Lages, W. F. (2004). Sensing for retrofitting of an industrial robot, *11th IFAC Symposium on Information Control Problems in Manufacturing*, Salvador, BA.
- Lippiello, V., Villani, L. and Siciliano, B. (2007). An open architecture for sensory feedback control of a dual-arm industrial robotic cell, *Industrial Robot: An International Journal* **34**(1): 46–53.
- Orenstein, L. P. (2010). Atualização de hardware e software de um manipulador industrial, *Projeto final de graduação*, Departamento de Engenharia de Controle e Automação, UFRJ.
- Pfeiffer, O., Ayre, A. and Keydel, C. (2008). *Embedded Networking with CAN and CANopen*, 2nd edn, Copperhill Technologies Corporation, RTC Books.
- Pizzino, C. A. (2008). *Controle não-linear adaptativo sensorless de um motor de indução baseado na corrente rotórica*, Master's thesis, Programa de Engenharia Elétrica - COPPE/UFRJ.
- Ribeiro, A. S., Almeida, A. G. S., de Souza, M. B. and Lima II, E. J. (2007). Metodologia para implementação de retrofitting de controladores de equipamentos de automação de processos, *8th Congresso Iberoamericano de Ingeniería Mecánica*, Cusco, Peru.
- Sciavicco, L. and Siciliano, B. (2000). *Modelling and Control of Robot Manipulators*, 2nd edn, Springer-Verlag Inc.
- Wen, J. T. and Kreutz-Delgado, K. (1991). The attitude control problem, *IEEE Transactions on Automatic Control* **36**(10): 1148–1162.