

# AMBIENTE PARA APOIO AO APRENDIZADO DE AUTOMAÇÃO E CONTROLE BASEADO EM LÓGICA RECONFIGURÁVEL

CARLOS R. E. LIMA, RAFAEL R. DA SILVA, HEITOR S. LOPES, JULIANO DE M. PEDROSO

*Laboratório de Bioinformática, CPGEI, Universidade Tecnológica Federal do Paraná  
Av. Sete de Setembro, 3165-CEP: 80230-901- Curitiba-PR-Brazil  
E-mails: erig@cefetpr.br, hslopes@pesquisador.cnpq.br*

**Abstract**— This work describes the implementation of an environment for simulating control systems based on reconfigurable logic that can be used for supporting undergraduated engineering students. This environment aims at supplying the requirements found in the academy, as well as in industrial applications, for open systems, easily adapted to other specifications, and with accessible cost and large processing power. By exploring modularity in reconfigurable logic, it is possible to adapt the system to new project requirements.

**Keywords**— Control, automation, FPGA, reconfigurable logic

**Resumo**— Este trabalho descreve a implementação de um ambiente de simulação de sistemas de controle baseado em lógica reconfigurável, que serve de apoio ao aprendizado deste conteúdo em cursos de engenharia. O ambiente descrito visa atender à demanda encontrada principalmente no meio acadêmico, mas extensível à aplicações industriais, por um sistema aberto, de fácil adaptação, custo adequado, com grande capacidade de processamento. Através da utilização de módulos em lógica reconfigurável é possível a rápida adaptação a novas demandas de projeto.

## 1 Introdução

Com a utilização de microprocessadores e microcontroladores cada vez mais poderosos, bem como com a adoção de linguagens de programação de alto nível observa-se não apenas um aumento de desempenho, mas também uma maior facilidade de implementação dos algoritmos de controle empregados em diversos processos industriais (Clements-Jewery et al., 1996). Como tendência, observam-se produtos cada vez mais compactos, organizados de forma modular, com diferentes configurações e interfaces, servindo adequadamente para a maior parte das aplicações acadêmicas ou não de automação industrial.

Contudo, em aplicações em ambiente acadêmico, seja de graduação ou de pós-graduação e ainda em aplicações industriais onde uma combinação de alto desempenho e adaptação são desejadas, sistemas convencionais mostram-se uma solução não satisfatória. No primeiro caso os principais obstáculos são: o custo elevado dos sistemas mais sofisticados e, o mais grave, a falta de acesso completo ao projeto de hardware e software dos mesmos, ou seja, tratam-se de sistemas fechados. No segundo caso, principalmente em projetos pilotos, onde é desejado uma grande capacidade de adaptação à novas demandas de hardware e software no processo a ser controlado, os CLPs comerciais mostram-se inadequados como solução, quer pelo custos de consultoria envolvidos, quer pela incapacidade de fornecer as interfaces desejadas, ou ainda por não propiciarem a capacidade de processamento desejada. Neste aspecto é importante enfatizar que existe uma demanda crescente na indústria por soluções de controle mais complexas, o que implica em maior poder de processamento e o

eventual desenvolvimento de novas interfaces com os sensores e atuadores (Takatsu et al., 1998).

Este trabalho descreve um ambiente baseado em lógica reconfigurável, com processamento hierarquicamente distribuído, para apoio ao aprendizado de problemas de automação e controle, em especial a tarefas que demandem um alto desempenho ou ainda uma fácil adaptação às novas demandas de hardware ou software. O ambiente descrito visa atender à demanda por um sistema aberto, de fácil adaptação, custo adequado, com grande capacidade de processamento encontrada principalmente no meio acadêmico, mas extensível às aplicações industriais. Isto vem de encontro à rápida evolução das inovações tecnológicas de hardware e de software na área de sistemas embarcados, automação e robótica, onde torna-se necessário desenvolver aplicações baseadas em metodologia que levem em conta a facilidade de futuras modificações, modernizações e expansões do sistema originalmente projetado. Para atingir este objetivo, além de basear um projeto em dispositivos lógicos reconfiguráveis de alto desempenho (FPGA-Field Programmable Gate Array), optou-se pela utilização integrada de softwares de desenvolvimento bastante difundidos dentro e fora do meio acadêmico.

O ambiente proposto é descrito na seção 2, onde são abordados aspectos básicos sobre lógica reconfigurável por hardware, é descrita a arquitetura de controle utilizada e aspectos gerais da implementação do ambiente. Na seção 3 é apresentado um exemplo de aplicação usando o ambiente descrito neste trabalho. Nesta seção são também apresentados resultados das simulações do exemplo descrito. Na seção 4 são apresentadas as conclusões do trabalho.

## 2 Descrição do Ambiente

### 2.1 Lógica Reconfigurável por Hardware

Sistemas reconfiguráveis são sistemas que apresentam a característica de substituir parte de seu *software* ou *hardware* para adaptarem-se a tarefas específicas. Tais sistemas têm por objetivo obter alto desempenho com baixo custo de produção, sendo uma alternativa às máquinas de Von Neumann implementadas nos sistemas com microprocessadores (Miyazaki, 1998, Compton, 2002, Lima et Rosário, 2003). Nos sistemas tradicionais, as tarefas são executadas de forma sequencial, com tempos de processamento que não são aceitáveis para muitas aplicações (Ito, 2000; Compton, 2002). Comparados aos sistemas de *software* reconfigurável os sistemas de *hardware* reconfigurável apresentam um maior potencial em termos de desempenho e adaptabilidade (Miyazaki, 1998).

Esta proposta pode ser aplicada a sistemas, embarcados ou não, que apresentam uma grande dinâmica tecnológica, ou seja, estão sujeitos a grandes variações tecnológicas em curto espaço de tempo, seja pela demanda por novas tarefas e desempenhos, ou seja, pela disponibilidade de novas tecnologias de sensores e atuadores.

O ambiente explora a modularidade e a grande facilidade de modificação, inerentes de projetos baseados em computação reconfigurável ou lógica reconfigurável, reduzindo o tempo de manutenção e de atualização de projetos (Lima et Rosário, 2003).

São características desejadas do ambiente:

- Processamento hierárquico
- Flexibilidade,
- Custo adequado à tarefa,
- Fácil manutenção e operação,
- Grande capacidade de expansão,
- Conceito modular e aberto,
- Utilização de lógica reconfigurável.

Os seguintes problemas demandam soluções que podem ser atendidas pelo ambiente proposto:

- Redução do tempo do ciclo de desenvolvimento de um sistema de automação ou sistema embarcado,
- Capacidade de adição de novas funcionalidades a um sistema já existente,
- Utilização de ferramenta de depuração e testes de novos sistemas,
- Demanda por requisitos de tempo real mais restritivos,
- Demanda por ferramentas de ensino para a formação de recursos humanos na área de engenharia.

Uma motivação adicional para o uso de lógica reconfigurável na implementação dos módulos do ambiente desenvolvido é a crescente disponibilidade de dispositivos de grande capacidade de desempe-

no. Por exemplo, algumas FPGAs mais recentes apresentam características como: grande número de pinos de I/O (mais de 700 pinos), impedância digitalmente controlada e linhas dedicadas para operação em modo de par diferencial. Soma-se a estas características a existência de módulos internos, multiplicadores, blocos PWM (Pulse Width Modulation) e registradores dedicados para operações de alto desempenho, além de uma grande capacidade de memória RAM interna de alta velocidade (perto de 2 Mbits). Tais dispositivos operam com até 5 milhões de células lógicas, permitindo inclusive a implementação de processadores completos com programa executado em memória interna (desempenho próximo de execução em memória cache).

### 2.1 Arquitetura

A figura 1 descreve através de um diagrama de blocos a arquitetura utilizada. Assim, os diferentes módulos de *software* e *hardware* são estruturados em blocos independentes, através da implementação de uma arquitetura hierárquica e aberta, distribuindo as diversas ações de controle em níveis crescentes de complexidade e utilizando recursos de computação reconfigurável por *hardware*.

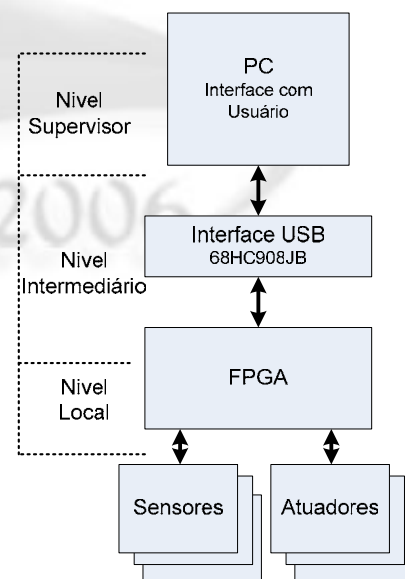


Figura 1 -Arquitetura distribuída, hierárquica e aberta implementada.

A divisão em diferentes níveis de processamento e controle permite uma melhor distribuição das diferentes tarefas possíveis, introduzindo, principalmente pela lógica reconfigurável a possibilidade de um grande número de processos sendo processados de forma paralela, de forma síncrona ou não. O paralelismo é uma grande vantagem na aplicação desta abordagem, já que permite a construção de modelos complexos muito mais próximos dos sistemas reais. Um exemplo típico desta vantagem está na simulação de um sistema com múltiplas juntas, a exemplo de um manipulador robótico. Aspectos de sincronização

destas juntas para execução de uma trajetória qualquer são facilmente atingidos graças ao paralelismo real encontrado nos blocos implementados em FPGA. Outro aspecto relevante é a possibilidade de introdução de novos blocos em paralelo com os blocos já existentes. Isto permite, por exemplo, que no mesmo manipulador robótico se introduza um bloco de ruído associado a cada junta, aprimorando o modelo do sistema.

Pode-se descrever :

- **Nível supervisor de processamento:** a supervisão de um ou mais sistemas níveis intermediários de processamento é executada através de estratégias de controle globais. Este nível também permite gerenciar e configurar tarefas específicas, estabelecer correções em função de informações obtidas da fusão de sensores, ou ainda modificar parâmetros de operação de um nível de processamento inferior. No caso de desenvolvimento de um protótipo, o controle supervisor é usado para fornecer parâmetros de operação do mesmo e facilitar a localização de erros no projeto. Este nível pode ser implementado dentro de diferentes sistemas operacionais de tempo real a exemplo de sistemas operacionais educacionais como o PET (Renaux, 1996) e o sistema Windows CE (Gerold, 1998).
- **Nível Intermediário de processamento:** as estratégias de controle neste nível permite a tomada de decisão com ou sem interferência do nível supervisor. A transferência de dados entre este nível e o nível de superior é implementada através da utilização de uma interface USB. As tarefas associadas a este nível serão executadas por rotinas implementadas em lógica reconfigurável ou combinação de microcontroladores embarcados com lógica reconfigurável. Neste nível podem também ser implementadas diferentes interfaces com o nível inferior, propiciando flexibilidade e adequação aos requisitos de projeto.
- **Nível local:** associado com as interfaces dos sensores e atuadores, atendendo à necessidade de propiciar processamento adequado das informações dos sensores e atuadores em ambiente industrial. Em lógica reconfigurável, este processamento será feito de forma paralela, permitindo que requisitos de tempo de execução do processamento dos sensores e atuadores sejam atendidos de forma mais fácil que utilizando computação convencional.

## 2.2 Interface de comunicação

A transferência de dados entre o PC e a placa com dispositivo FPGA é feita através de uma interface USB 1.1 baseada em um microprocessador Motorola 68HC908JB. A figura 2 apresenta o diagrama de blocos do circuito da interface implementada.

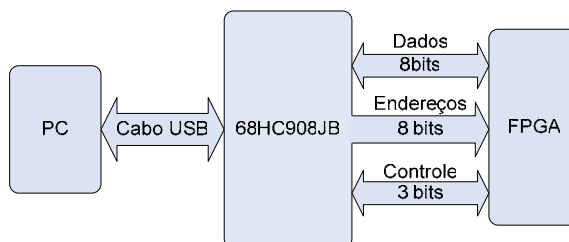


Figura 2. Diagrama de blocos da interface de comunicação utilizada.

## 2.3 Interface com o Usuário

A interface com o usuário é desenvolvida em Visual C ou Visual Basic utilizando-se de *device drivers* fornecidos por fabricante do 60HC908JB. Isto minimiza o tempo desenvolvimento de aplicativos específicos para aplicações específicas. Em aplicações acadêmicas é utilizado para visualização, comparação e análise de resultados o *software* Matlab.

## 2.4 Dispositivo FPGA

O dispositivo utilizado para as implementações em *hardware* foi o Stratix II EP2S15F484C3 da Altera (ALTERA, 2006). Este dispositivo está montado em uma placa de desenvolvimento padrão e conecta-se com o PC através da interface USB já descrita. Para eventuais sensores e atuadores externos são utilizados barramentos padrões, disponibilizando mais de uma centena de pinos de I/O para entrada e saída de dados.

Com o objetivo de avaliar a eficiência do ambiente, antes mesmo da implementação efetiva em *hardware*, diversas simulações foram feitas usando o *software* Quartus II (ALTERA 2006). Há três motivações para este tipo de simulação:

- Verificar se o sistema pode realmente responder como esperado, dentro de uma margem de erro aceitável.
- Determinar se os requisitos de tempo de processamento são aceitáveis.
- Estimar os recursos (células lógicas e memória) demandados para a aplicação.

## 3 Exemplo de Implementação

O ambiente implementado permite o desenvolvimento de inúmeras aplicações. É apresentada uma aplicação didática de um sistema de controle de velocidade de um motor DC usando um controlador PID. Para

ilustrar a flexibilidade do ambiente o motor DC é implementado através de um modelo discreto. Os resultados do experimento são transferidos para o Matlab através de um aplicativo desenvolvido em Visual C. O controlador foi discretizado tendo em vista sua possível utilização em uma aplicação de uma planta real, tendo em vista a robustez a ruídos.

Para aumentar a velocidade de processamento do bloco PID, paralelizaram-se as funções Proporcional, Integral e Derivativa. Assim em um mesmo ciclo de amostragem, as três funções são calculadas em paralelo.

A saída do controlador  $U[n]$  é a soma das saídas dos sub-blocos, como mostrado na equação (1). As ações Proporcional ( $P[n]$ ), Integrativa ( $I[n]$ ) e Derivativa ( $D[n]$ ) são descritas nas equações (2), (3) e (4), respectivamente.

$$U[n] = P[n] + I[n] + D[n] \quad (1)$$

$$P[n] = K_p \cdot e[n] \quad (2)$$

$$I[n] = I[n-1] + \frac{K_p \cdot T_s}{2 \cdot T_i} (e[n] - e[n-1]) \quad (3)$$

$$D[n] = \frac{(pT_s - 2)}{(pT_s + 2)} \cdot D[n-1] + \quad (4)$$

$$\frac{2 \cdot K_p \cdot T_d}{T_s \cdot (pT_s + 2)} \cdot (e[n] - e[n-1])$$

Para minimizar equações matemáticas, consideramos parâmetros normalizados  $ki$ ,  $kd$  e  $kd\_c$ , como mostrado nas equações (5), (6) e (7).

$$ki = \frac{K_p \cdot T_s}{2 \cdot T_i} = \frac{K_i \cdot T_s}{2} \quad (5)$$

$$kd = \frac{2 \cdot K_p \cdot T_d}{T_s \cdot (pT_s + 2)} = \frac{2 \cdot K_d}{T_s \cdot (pT_s + 2)} \quad (6)$$

$$kd\_c = \frac{(pT_s - 2)}{(pT_s + 2)} \quad (7)$$

O diagrama de blocos do controlador implementado em hardware é mostrado na figura 3.

O modelo matemático utilizado para o motor CC é representado pela equação (8), na sua forma contínua. Usando uma discretização *zero-order hold*, com taxa de amostragem ( $T_s$ ) de 120ms obtém-se a equação (9).

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{0.01}{(0.01s + 0.1) \cdot (0.5s + 1) + 0.0001} \quad (8)$$

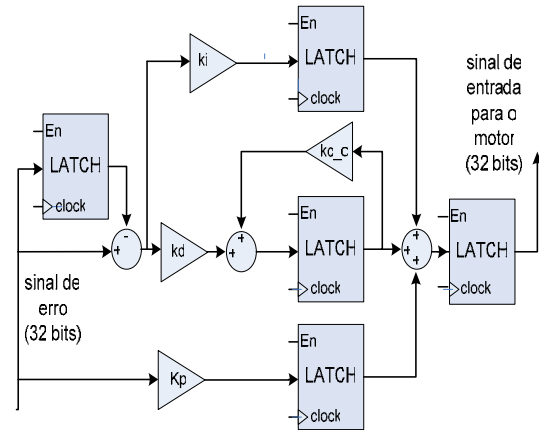


Figura 3. Diagrama de Blocos do controlador PID.

$$\frac{\dot{\theta}(z)}{V(z)} = \frac{0,0092z^{-1} + 0,0057z^{-2}}{1 - 1,0877z^{-1} + 0,2369z^{-2}} \quad (9)$$

Finalmente, implementa-se a equação (9) representando-a por equações a diferenças, segundo a equação (10).

$$\begin{aligned} \dot{\theta}[k] = & 0,0092 \cdot V[k-1] + \\ & 0,0057 \cdot V[k-2] + \\ & 1,0877 \cdot \dot{\theta}[k-1] - 0,2369 \cdot \dot{\theta}[k-2] \end{aligned} \quad (10)$$

Como neste projeto a representação das variáveis em hardware é feita por uma string de bits é necessária uma conversão dos valores reais para valores binários. Na Tabela (1) é mostrada esta conversão associada aos parâmetros do diagrama de blocos da figura 4, representado segundo uma forma canônica.

Tabela 1. Conversão dos parâmetros adotada para representação de números reais como inteiros.

Valores Reais associados aos parâmetros	Valor Convertido
b1 = 0,0092	151 = 0x0097h
b2 = 0,0057	93 = 0x005Dh
a1 = 1,0877	17821 = 0x459Dh
a2 = 0,2369	3881 = 0x0F29h

nica.

Para confrontar o modelo do motor CC na FPGA, é analisada a sua resposta ao degrau unitário, segundo a figura 5. Nesta figura são mostradas duas curvas, uma obtida através do processamento dos blocos implementados em hardware e usando a simulação do modelo em Matlab. O valor de regime no Matlab foi de 0,1 e o valor implementado em FPGA ficou um pouco maior. Este erro era esperado, já que os blocos em *hardware* usaram representação dos números como inteiros.

O controlador PID foi parametrizado com Referência igual a 1,  $K_p$  igual a 15,  $K_i$  igual a 33,33 e  $K_d$  igual a 1,44. Como o modelo foi simulado sem ruído, o fator de correção não foi utilizado. Aplicando-se as equações (5), (6) e (7), os parâmetros no FPGA fo-

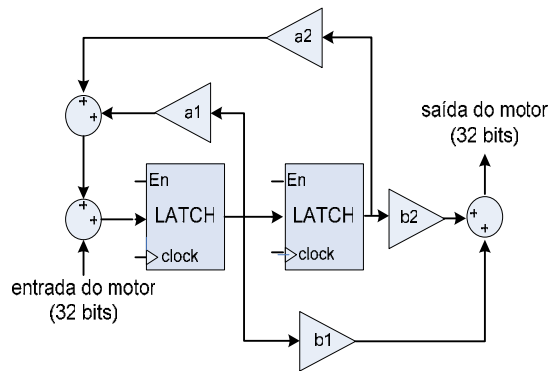


Figura 4. Diagrama de blocos do modelo do motor CC discretizado.

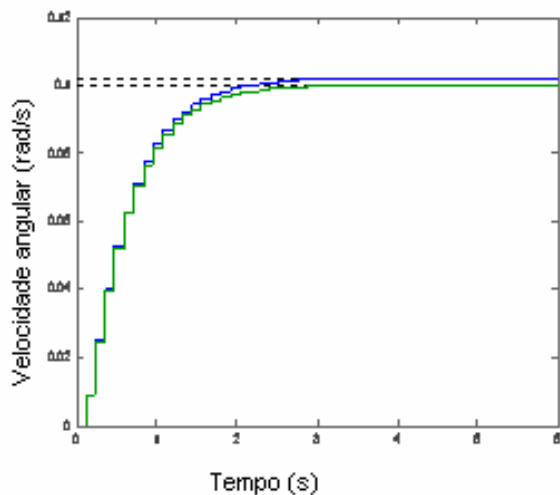


Figura 5. Resposta ao degrau do Motor CC em malha aberta. A curva em verde é o resultado no Matlab e a curva em azul é o resultado do hardware implementada na FPGA.

ram  $k_i = 2$ ,  $k_d = 12$  e  $k_d_c = 0$ . Na figura 6 é apresentada a resposta do motor usando o controlador PID ajustado com estes parâmetros. Mais uma vez são confrontadas duas curvas: uma resultante do processamento em hardware e outra obtida pela simulação no Matlab. Também são observados pequenos erros devido à representação numérica utilizada.

#### 4 Conclusão

Ambientes de desenvolvimento baseados em lógica reconfigurável por software apresentam características adequadas para execução de projetos de automação e controle. Eles permitem que o controle de um processo industrial seja feito de modo a aproveitar as características de baixo consumo, alta velocidade de operação, capacidade de integração, flexibilidade e facilidade de programação dos dispositivos lógicos reprogramáveis.

Ferramentas que venham a permitir a rápida implementação prática de um projeto de automação e controle não são apenas uma comodidade técnica, mas são também essenciais para a viabilidade econômica deste projeto. Isto é válido para a empresa e

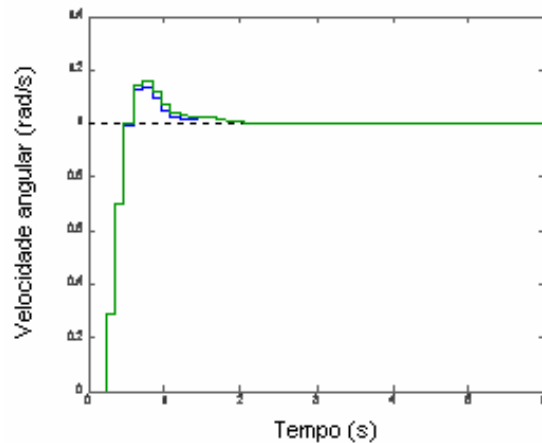


Figura 6. Resposta ao degrau do sistema em malha fechada do Motor CC. A curva em verde é o resultado do Matlab, e a curva em azul é o resultado da hardware implementada na FPGA

para a universidade e centros de pesquisa, cada vez mais sujeitos as competições do mercado.

Com este trabalho espera-se também contribuir para a difusão dos sistemas baseados em lógica reconfigurável, bem como, através de um exemplo prático, demonstrar sua aplicabilidade no ambiente acadêmico. A utilização de interfaces gráficas e VHDL no projeto permitem a modularidade e reusabilidade, onde diferentes blocos são projetados separadamente para posterior integração. As diversas ferramentas integradas no ambiente permitem a visualização de possíveis erros de projeto, antes mesmo da execução física do mesmo. Outra vantagem é a característica de hierarquização, que facilita o trabalho em equipe. A utilização de lógica reconfigurável contribui para verificar os limites da aplicação desta alternativa de implementação de algoritmos em problemas práticos que exijam grande poder de processamento. O projeto desenvolvido é ainda uma ferramenta de apoio didático em laboratórios de graduação e de pós-graduação de Engenharia Elétrica da UTFPR.

Alguns aspectos promissores do ambiente desenvolvido são:

- Flexibilidade – Há uma grande variedade de configurações possíveis na implementação de soluções para diversos problemas de automação e controle.
- Ambiente aberto – Permite que seus blocos componentes sejam totalmente acessados e eventualmente modificados para agregar novas soluções.
- Facilidade de expansão – Novos recursos podem ser facilmente adicionados a um projeto, permitindo a adição de novas funcionalidades.

Este trabalho abre perspectivas para a realização dos seguintes trabalhos futuros, entre outros:

- Fácil expansão – Novos módulos podem ser gradativamente adicionados ao sistema original permitindo que novas funcionalidades

sejam implementadas, permitindo inclusive, no caso de lógica reconfigurável, que isto seja feito sem a necessidade de realizar um novo projeto de hardware.

- Velocidade de operação – Além da característica de operação em paralelo, presente em algoritmos implementados com lógica reconfigurável, a execução de módulos em hardware apresenta tempos de processamento muito menores se comparados com a mesma implementação em software.
- Domínio tecnológico – O projeto proposto apresenta-se como oportunidade de domínio de tecnologias associadas a sistemas microprocessados, sistemas operacionais, interfaces, lógica reconfigurável, entre outras.

### Referências Bibliográficas

- Altera Corporation (2006), URL: <http://www.altera.com>, 2006.
- Clements-Jewery, K., Jeffcoat, W. e Rogers, S. (1996). *The PLC Workbook–Programmable Logic Controllers Made Easy*. Prentice Hall, London, UK.
- Compton, K. (2002). Reconfigurable computing: a survey of systems and software. *ACM Computing Surveys*, **34**(2) 171 – 210.
- Ito, S.A. e Carro, L. (2000). A comparison of microcontrollers targeted to FPGA-based embedded applications. *Proceedings of the IEEE 13th Symposium on Integrated Circuits and Systems Design Proceedings*, pp. 397 – 402.
- Lima, C.R.E. e Rosário, J.M. (2003). Reconfigurable architecture proposal applied to mobile robots. *Proceedings of the 11th International Conference on Advanced Robotics, ICAR 2003*, Vol. 2, pp.601 – 606.
- Miyazaki, T. (1998). Reconfigurable systems: a survey. *Proceedings of IEEE Design Automation Conference, ASP-DAC '98*, pp. 447 – 452.
- Renaux, D.P.B. (1996). *PET – A Small Real-Time Support System for Microcontrollers without Virtual Memory*. Technical Report, Laboratory for Embedded Systems Innovation and Technology - CEFET-PR.
- Takatsu, H., Itoh, T. e Araki, M. (1998). Future needs for the control theory in industries – report and topics of the control technology survey in Japanese industry. *Journal of Process Control*, **8**(5-6), pp. 369 – 374.