

## UM AMBIENTE PARA APOIO AO APRENDIZADO DE AUTOMAÇÃO E CONTROLE E SUA APLICABILIDADE NO DESENVOLVIMENTO DE COMPETÊNCIAS E HABILIDADES

**Carlos R. E Lima** – erig@utfpr.edu.br  
**Heitor S. Lopes** – hslopes@utfpr.edu.br  
**Luciano Rocha** – baracho@utfpr.edu.br  
Universidade Tecnológica Federal do Paraná  
Av. Sete de Setembro, 3165  
CEP 80230-901 – Curitiba - Paraná

***Resumo:** Este trabalho descreve a aplicação de um ambiente para apoio à aprendizagem de sistemas de controle, baseado em lógica reconfigurável, com processamento hierarquicamente distribuído, em especial em tarefas que demandem alto desempenho ou ainda uma fácil adaptação à novas demandas de hardware ou de software. O ambiente descrito visa atender à demanda por um sistema aberto de fácil adaptação, custo adequado, com grande capacidade de processamento. O ambiente foi planejado para dar suporte ao ensino de graduação e de pós-graduação, e também pode ser utilizado na indústria, permitindo a realização de experimentos de laboratórios de várias disciplinas separadamente ou como uma plataforma integrando várias disciplinas ou áreas de conhecimento. Uma aplicação do ambiente para a disciplina de Controle é apresentada, onde o ambiente é utilizado para realizar o controle da velocidade de um motor. Como exemplo de atividades integradoras, sugere-se a utilização do ambiente como uma plataforma para controle de um robô móvel. Finalmente, baseado nos exemplos apresentados, faz-se uma análise das competências que são desenvolvidas com o uso do ambiente.*

***Palavras-chave:** Ambiente de ensino de controle, Lógica reconfigurável, Plataforma de robô móvel.*

### 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 facilidade maior de implementação dos algoritmos de controle empregados em diversos processos industriais (CLEMENTS-JEWERY et al., 1996). Observa-se uma tendência a produtos cada vez mais compactos, organizados de forma modular, com diferentes configurações e interfaces, servindo satisfatoriamente à maior parte das aplicações acadêmicas de automação industrial.

Contudo, para aplicações em ambiente acadêmico, seja de graduação ou de pós-graduação e, ainda, para aplicações industriais onde uma combinação de alto desempenho e

adaptação é um requisito, sistemas convencionais não se mostram adequados. 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, são sistemas proprietários. No segundo caso, principalmente em projetos-piloto, onde uma grande capacidade de adaptação às novas demandas de *hardware* e *software* no processo a ser controlado é um requisito fundamental, as soluções comerciais se mostram inadequadas como solução. Esta inadequação decorre dos custos de consultoria envolvidos, da incapacidade de fornecer as interfaces desejadas, ou ainda da falta de 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, aplicado ao estudo de problemas de automação e controle, em especial a tarefas que demandem um alto desempenho ou ainda uma fácil adaptabilidade. 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, porém extensível a aplicações industriais. Isto vem de encontro à rápida evolução das inovações tecnológicas de *hardware* e *software* na área de sistemas embarcados, automação e robótica. Nestas áreas, torna-se necessário desenvolver aplicações baseadas em metodologias que levem em conta a facilidade de futuras modificações, modernizações e expansões do sistema originalmente projetado. Para atingir este objetivo, além de basear o 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. Dentro deste contexto, este trabalho descreve também a aplicação do ambiente para apoio ao ensino de laboratório de diversas disciplinas direta ou indiretamente relacionadas à área de controle e automação. O projeto é desenvolvido por uma equipe multidisciplinar de professores e estudantes do curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná – UTFPR, estudantes de graduação das áreas de Engenharia Industrial Eletrônica e Engenharia Industrial Mecânica. A idéia é utilizar-se dos recursos da moderna tecnologia disponível no mercado para criar um ambiente flexível e de baixo custo, para realizar tarefas de alto desempenho, visando permitir aos estudantes passarem a utilizar os recursos de lógica reconfigurável em seus projetos.

Na seção 2 faz-se uma descrição do ambiente e de como é possível utilizar-se de dispositivos FPGA existentes no mercado para projetar sistemas embarcados com lógica reconfigurável. Em seguida, apresenta-se um exemplo da utilização do ambiente em uma experiência da disciplina de controle linear onde se faz o controle de um motor CC (corrente contínua). Na seção 3 indica-se como o ambiente pode ser usado em um projeto multidisciplinar integrador, através do projeto de uma plataforma para controle de um robô móvel.

Baseado nos exemplos de uso do ambiente, na seção 4 faz-se uma análise das vantagens e competências que o estudante pode desenvolver e das vantagens trazidas do ponto de vista curriculares para os estudantes, observando-se as diretrizes curriculares nacionais.

## **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 E ROSÁRIO, 2003). Nos sistemas tradicionais, as tarefas são executadas de forma seqüencial, com tempos de processamento que não são aceitáveis para muitas aplicações (ITO E CARRO, 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 apresentem uma grande dinâmica tecnológica. Isto é, 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 proposto 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 E 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 nas áreas de engenharia elétrica, engenharia de controle e automação e engenharia da computação.

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 desempenho. Por exemplo, algumas FPGAs mais recentes apresentam características como: grande número de pinos de entrada/saída (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 diversos módulos internos, tais como multiplicadores, blocos PWM (*Pulse Width Modulation*) e registradores dedicados a operações de alto desempenho, além de grande capacidade de memória RAM interna de alta velocidade (perto de 2 Mbits). Tais dispositivos disponibilizam mais 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*).

## 2.2 Arquitetura

A Figura 1 descreve a arquitetura utilizada através de um diagrama de blocos. 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*.

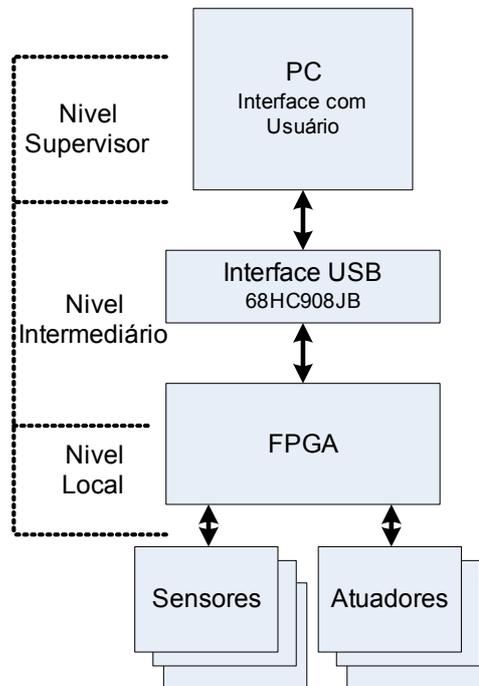


Figura 1 -Arquitetura distribuída, hierárquica e aberta implementada.

A divisão em diferentes níveis de processamento e controle permite uma distribuição melhor das diferentes tarefas possíveis, introduzindo, graças à lógica reconfigurável, a possibilidade de um grande número de processos sendo processados de forma paralela real, síncrona ou assincronamente. 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 manipulador robótico de múltiplas juntas. Aspectos de sincronização destas juntas para execução de uma trajetória qualquer são facilmente atingidos graças ao paralelismo real encontrado nos blocos implementados em FPGA. Um outro aspecto relevante é a possibilidade de introdução de novos blocos em paralelo com aqueles 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.

Na arquitetura proposta, os níveis hierárquicos são descritos como se segue:

- **Nível supervisor de processamento:** a supervisão de um ou mais sistemas de nível intermediário 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 (WILSON e HAVEWALA, 2001).

- **Nível Intermediário de processamento:** as estratégias de controle neste nível permitem a tomada de decisão com ou sem a 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 pode ser feito de forma paralela, permitindo que requisitos de tempo de execução do processamento dos sensores e atuadores sejam atendidos de forma mais fácil do que utilizando computação convencional.

### 2.3 Interface de comunicação

A transferência de dados entre um computador-base (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. É importante ressaltar que a placa permite explorar também outros tipos de comunicação por interfaces padronizadas, tais como: interface serial, paralela e Ethernet.

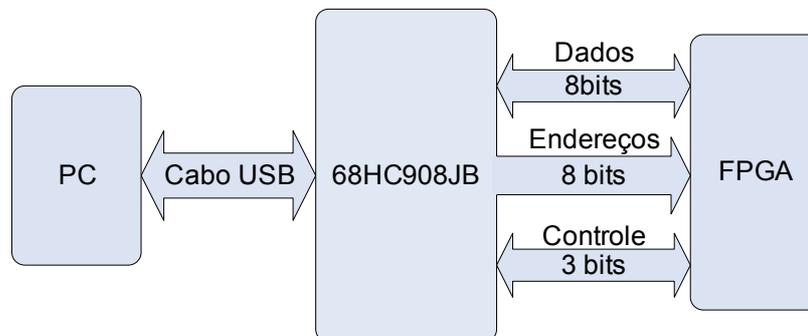


Figura 2. Diagrama de blocos da interface de comunicação utilizada.

### 2.4 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 sistemas para aplicações específicas. Em aplicações acadêmicas é utilizado para visualização, comparação e análise de resultados o *software* MATLAB (MATHWORKS, 2007)

## 2.5 Dispositivo FPGA

O dispositivo utilizado para as implementações em *hardware* foi o Stratix II EP2S15F484C3 da Altera (ALTERA, 2007). 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 padronizados, disponibilizando mais de uma centena de pinos de entrada/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, 2007). Há três motivações para este tipo de simulação:

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

## 3 EXEMPLO DE APLICAÇÃO DO AMBIENTE PARA O ENSINO DE CONTROLE

Para ilustrar a flexibilidade do ambiente, um motor CC foi implementado em *hardware* 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 na simulação de uma planta real, dado à sua robustez a ruídos.

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

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.

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

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

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

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

Para minimizar equações matemáticas, considerou-se parâmetros normalizados  $k_i$ ,  $k_d$  e  $k_{d\_c}$ , como mostrado nas equações (5), (6) e (7).

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

$$kd = \frac{2 \cdot Kp \cdot Td}{Ts \cdot (pTs + 2)} = \frac{2 \cdot Kd}{Ts \cdot (pTs + 2)} \quad (6)$$

$$kd_c = \frac{(pTs - 2)}{(pTs + 2)} \quad (7)$$

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

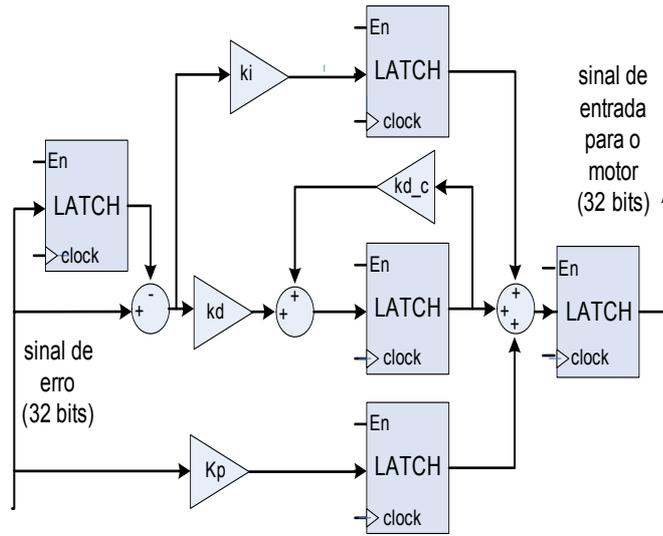


Figura 3: Diagrama de Blocos do Controlador PID.

O modelo matemático utilizado para o 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).

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{0.01}{(0.01s + 0.1) \cdot (0.5s + 1) + 0,0001} \quad (8)$$

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

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

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

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 a 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

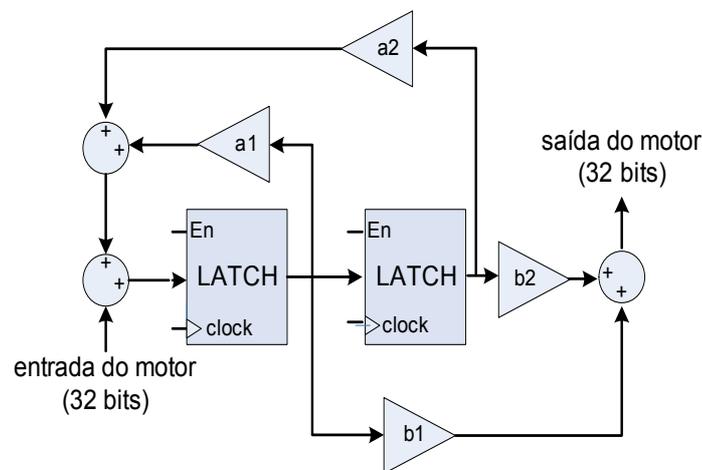


Figura 4: Diagrama de blocos do modelo do motor CC discretizado

Para confrontar o modelo do motor CC na FPGA, é analisada a sua resposta ao degrau unitário, segundo a Figura 5 - esquerda. Nesta figura são mostradas duas curvas, uma obtida através do processamento dos blocos implementados em *hardware* (curva azul) e usando a simulação do modelo em MATLAB (curva verde). 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 foram  $k_i = 2$ ,  $k_d = 12$  e  $k_d_c = 0$ . Na Figura 5 - direita 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* (curva azul) e outra obtida pela simulação no MATLAB (curva verde). Também são observados pequenos erros devido à representação numérica utilizada.

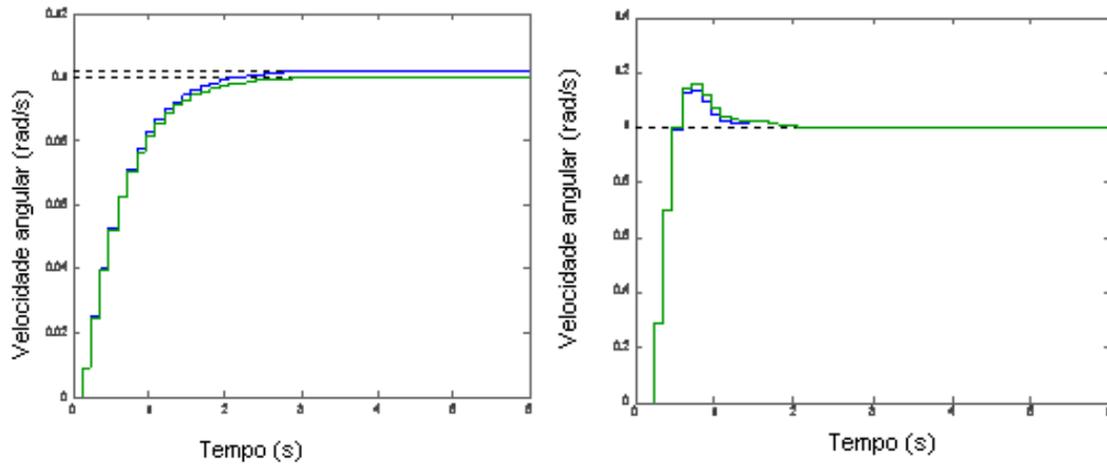


Figura 5: Esquerda resposta ao degrau do motor CC em malha aberta. Direita: resposta ao degrau do sistema em malha fechada do motor CC.

#### 4 EXEMPLO DE APLICAÇÃO DO AMBIENTE COMO PLATAFORMA PARA CONTROLE DE UM ROBÔ MÓVEL

O ambiente projetado apresenta possibilidade de ser usado em projetos integradores. Nesta seção será ilustrada a aplicação do ambiente para o controle de um robô móvel.

A arquitetura proposta é bastante geral para ser utilizada com diversos modelos de robô móvel. Como uma aplicação desta arquitetura, uma plataforma com configuração *drive* diferencial foi usada no projeto (BORENSTEIN et al, 1996). Figura 6 apresenta um modelo simplificado desta plataforma e o protótipo real desenvolvido.

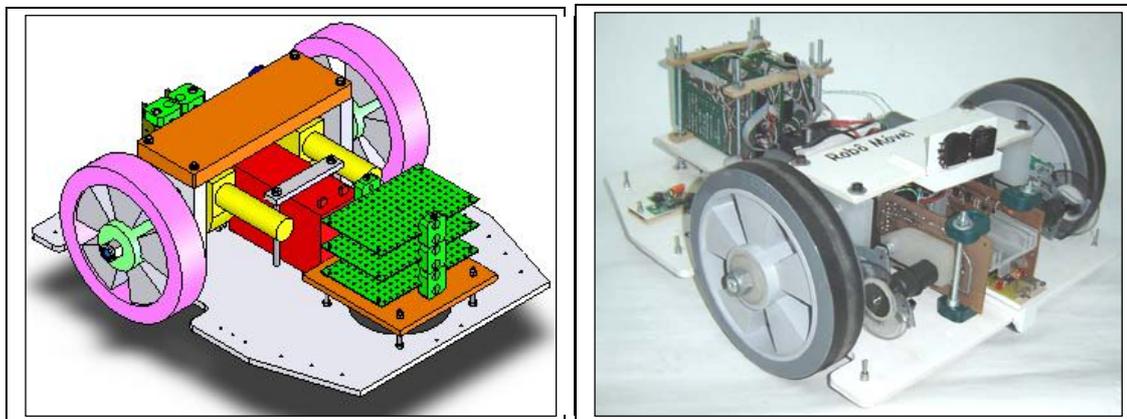


Figura 6: Plataforma do robô móvel: projeto e protótipo real desenvolvido

A Figura 7 detalha a arquitetura proposta aplicada ao protótipo desenvolvido. O nível supervisor é construído utilizando-se uma interface baseada no sistema operacional Microsoft Windows CE (WILSON & HAVEWALA, 2001), operando em um computador portátil remoto. O enlace de comunicação entre este nível e nível embarcado foi implementado através de uma rede local sem fio (*wireless-LAN*).

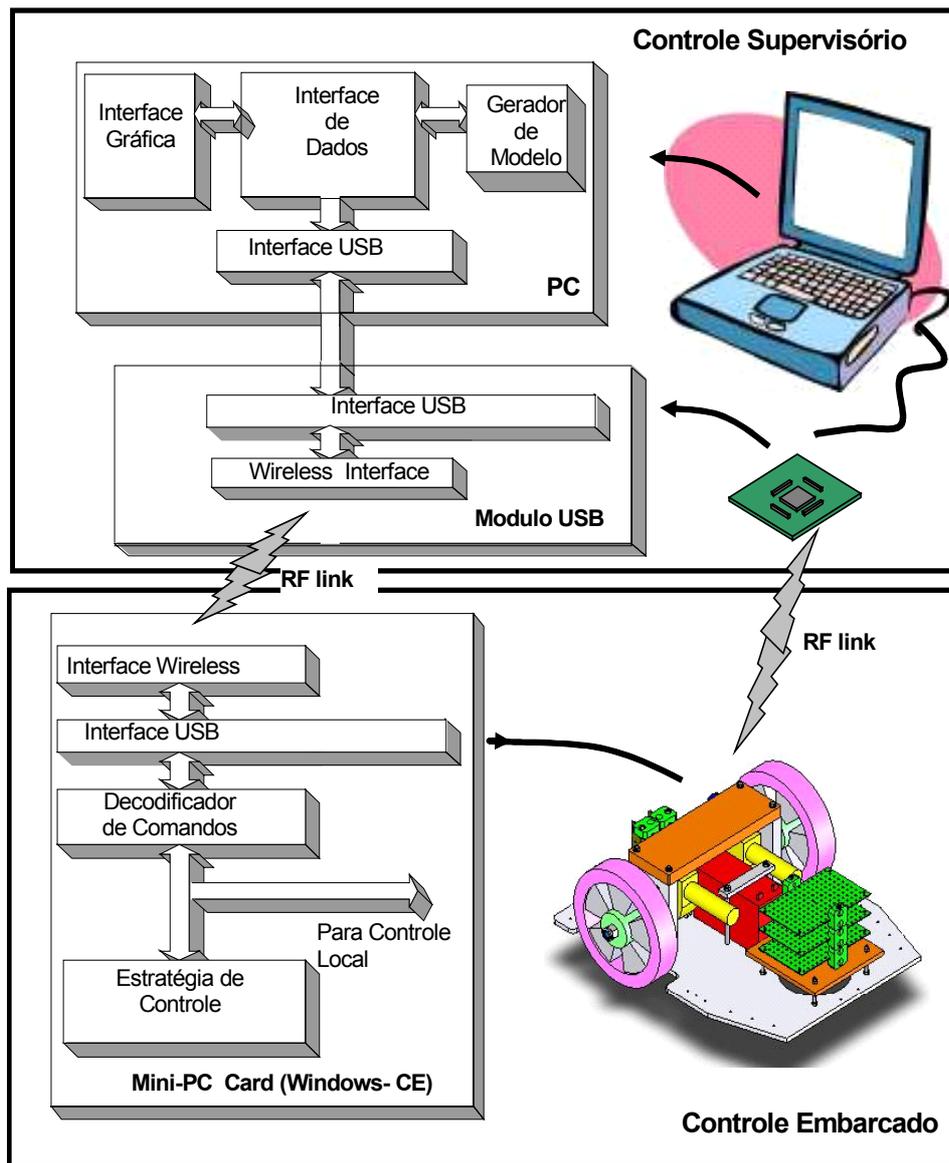


Figura 7: Detalhes dos blocos que compõe o sistema embarcado os níveis de controle de supervisão

#### 4.1 Nível de controle embarcado

No nível de controle embarcado duas tarefas são implementadas: o decodificador de comandos e o controle lógico. A primeira tarefa decodifica os comandos vindos da interface de comunicação embarcada (do nível de controle supervisão), permitindo a execução de ações diferentes através dos dados recebidos. A segunda tarefa gera os sinais de controle para as interfaces dos atuadores, ambas localizadas no nível de controle. Estratégias de controle são implementadas no bloco de controle lógico.

O protótipo usa a FPGA descrita na seção 2.5, onde, neste caso, os algoritmos são implementados em VHDL ou linguagem Graphic na plataforma Quartus II da Altera. As estratégias de controle embarcado no bloco de controle lógico foram desenvolvidas usando linguagem C++, em um ambiente SOPC (*System-On-a-Programmable Chip*) ou através do uso dos blocos implementados no *hardware* (ALTERA, 2007).

## 4.2 Nível de controle local

O diagrama de blocos do nível de controle local é mostrada na Figura 8, incluindo todas as interfaces necessárias com sensores e atuadores (motores). Do ponto de vista da arquitetura aberta, a implementação destes controladores locais deve ser suficientemente geral de modo a permitir uma fácil reconfiguração da lei de controle em relação às especificações do usuário, ou em relação à tarefa executada pelo robô.

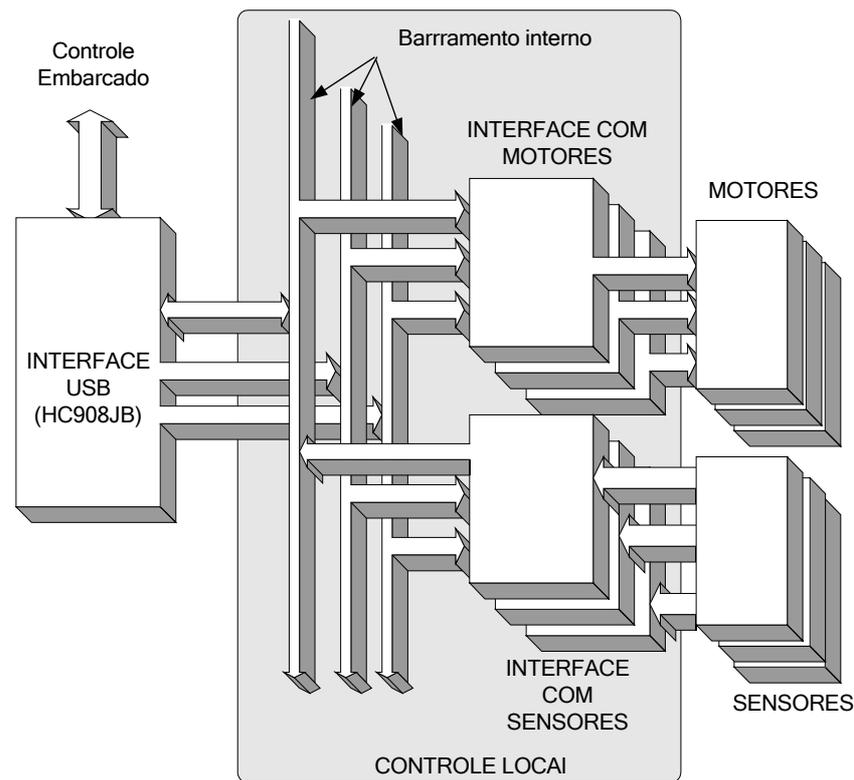


Figura 8: Diagrama de blocos do nível de controle local

## 5 ANÁLISE DA QUESTÃO PEDAGÓGICA RELACIONADA AO AMBIENTE E DAS COMPETÊNCIAS DESENVOLVIDAS

### 5.1 Utilização do ambiente para apoio ao ensino de graduação e de pós-graduação

Características principais que tornam este ambiente promissor 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.
- Velocidade de operação – além das características 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 *softwares*.

- Atualização tecnológica – O projeto proposto apresenta-se como oportunidade de domínio de tecnologias associadas a sistemas microprocessados, sistemas operacionais, interfaces, lógica reconfigurável, entre outras, colocando o estudante a par do estado da técnica e da arte na área de engenharia.

Com essas características a plataforma pode ser utilizada nos cursos de graduação para apoiar o ensino de diversas disciplinas práticas de laboratório, como eletrônica digital e microprocessadores, eletrônica de potência, controle linear, sistemas embarcados, engenharia de *software*. O principal impacto nos estudantes é a mudança de paradigma como consequência da migração do modelo de *hardware* para hardware configurável em sistemas embarcados. Da mesma forma, denomine colunas/linhas em tabelas, com respectivas unidades, caso aplicável.

## **5.2 Utilização da plataforma para projetos integradores do currículo**

A plataforma pode ser utilizada para a realização de projeto que integrem o conhecimento de diversas disciplinas, áreas ou mesmo cursos de engenharia. Estes projetos vêm de encontro às diretrizes curriculares nacionais, que determinam que em cada curso de engenharia devam existir trabalhos de síntese e integração dos conhecimentos e ser estimuladas atividades complementares, como trabalhos de iniciação científica ou desenvolvimento de protótipos. (Artigo 5, parágrafos 1º e 2º das Diretrizes Curriculares Nacionais) .

Nos Estados Unidos projetos integrando competências são incluídos no currículo dos cursos de graduação em engenharia. Um exemplo deste tipo de projeto pode ser encontrado no Trinity College (VERNER & ALGHREN, 2004), onde os estudantes se matriculam em uma disciplina oferecida no primeiro ano, onde realizam um projeto semestral em equipe no qual projetam e constroem um robô para combate de fogo e participam de uma competição internacional de robôs. Os estudantes matriculados na disciplina são orientados por membros de um Grupo de Estudo de Robótica, formado por estudantes de todos os anos do curso.

Aplicação da plataforma para o desenvolvimento de plataforma de controle para robôs móveis, conforme descrito na seção 3, é um tipo de projeto com características multidisciplinares, enquadrando-se perfeitamente naquilo que recomendam as diretrizes curriculares para a engenharia. O desenvolvimento de sistemas embarcados para robôs móveis exige conhecimento multidisciplinar: modelagem, controle, automação, sistemas de potência, transmissão de dados, sensores, processamento de sinais, engenharia de software e inteligência artificial.

Diversas competências e habilidades gerais indicadas nas diretrizes curriculares podem ser desenvolvidas com este tipo de projeto, relacionadas na tabela 3, baseado nas diretrizes curriculares nacionais (CONSELHO NACIONAL DE EDUCAÇÃO, 2002). A tabela não esgota todas as competências e habilidades gerais para o engenheiro relacionadas nas diretrizes, mas pretende apenas indicar aquelas julgadas mais diretamente relacionadas com o ambiente de ensino proposto. As numerações na coluna da direita da tabela correspondem à numeração das competências e habilidades das diretrizes curriculares.

Tabela 3: Competências e habilidades desenvolvidas com o uso do ambiente segundo as diretrizes curriculares nacionais

Competências e Habilidades	Ambiente de Ensino	Nº
Aplicar conhecimentos matemáticos, científicos, tecnológicos e instrumentais à engenharia;	Aplicação de conhecimentos de matemática, sistemas lineares, controle, comunicações, microprocessadores e lógica reconfigurável e uso das plataformas com lógica reconfigurável;	I
Projetar e conduzir experimentos interpretar resultados;	Experimentos ligados às disciplinas, como controle de motor CC ou como partes do projeto integrado;	II
Conceber, planejar, elaborar e analisar sistemas e produtos;	Projetos integradores, como a plataforma de controle do robô móvel	III
Planejar, supervisionar, elaborar e coordenar projetos e serviços da engenharia;	Coordenar um projeto integrado multidisciplinar envolvendo grupos de estudantes de diversos cursos, num concurso de robôs, por exemplo;	IV
Identificar, formular e resolver problemas de engenharia;	Diversos problemas de engenharia ligados ao projeto do robô, como por exemplo, algoritmos de programação para realizar o controle da plataforma do robô;	V
Desenvolver e/ou utilizar novas ferramentas ou técnicas;	Uso de recursos tecnológicos atualizados, por exemplo, sistemas microprocessados, FPGAs, ambientes de simulação, softwares de programação, e uso de lógica reconfigurável, entre outros;	VI
Avaliar criticamente a operação e a manutenção de sistemas;	Avaliar as condições para o funcionamento dos projetos;	VII
Comunicar-se eficientemente nas formas escrita, oral e gráfica;	Comunicar-se com os colegas de projeto, comunicar-se com o público, elaborar artigos, participar de discussões;	VIII
Atuar em equipes multidisciplinares: aprender a trabalhar em equipe, no caso do projeto do robô móvel;	Projetos integradores envolvendo trabalho em equipe;	IX
Compreender e aplicar a ética e responsabilidade profissional;	Desenvolvimento de valores éticos com a participação de projetos e trabalhos em grupo e em prováveis concursos;	X
Avaliar a viabilidade econômica de projetos de engenharia.	Os projetos podem envolver escolha de componentes e recursos, envolvendo custos, por exemplo.	XII

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

ALTERA CORPORATION. Disponível em: <<http://www.altera.com>> . Acesso em: 25 de junho de 2007.

BORENSTEIN, J.; EVERETT, H. R.; FENG, L. **Navigating Mobile Robots: Sensors and Techniques**. Wellesley: A. K. Peters, Ltd., 2006

CLEMENTS-JEWERY, K.; JEFFCOAT, W.; ROGERS, S. **The PLC Workbook—Programmable Logic Controllers Made Easy**. London: Prentice Hall, 1996

COMPTON, K. Reconfigurable computing: a survey of systems and software. **ACM Computing Surveys**, v. 34, n. 2, p. 171 – 210, 2002

CONSELHO NACIONAL DE EDUCAÇÃO. Resolução n. 11 de 11 de Março de 2002. Dispõe sobre as diretrizes curriculares nacionais para os cursos de engenharia. **Diário Oficial da República Federativa do Brasil**, Brasília, 11/mar/2002

ITO, S.A.; CARRO, L. A comparison of microcontrollers targeted to FPGA-based embedded applications. In: PROCEEDINGS OF THE IEEE 13<sup>TH</sup> SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN PROCEEDINGS, 2000, p. 397 – 402

LIMA, C.R.E.; ROSÁRIO, J.M. Reconfigurable architecture proposal applied to mobile robots. In: PROCEEDINGS OF THE 11<sup>TH</sup> INTERNATIONAL CONFERENCE ON ADVANCED ROBOTICS, ICAR 2003, n. 2, p. 601 – 606, 2003

MIYAZAKI, T. Reconfigurable systems: a survey. In: PROCEEDINGS OF IEEE DESIGN AUTOMATION CONFERENCE, ASP-DAC '98, 1998, p. 447 – 452

RENAUX, D.P.B. PET – A Small Real-Time Support System for Microcontrollers without Virtual Memory. **Technical Report**. Laboratory for Embedded Systems Innovation and Technology - CEFET-PR, Curitiba, 1996.

TAKATSU, H.; ITOH, T.; ARAKI, M. Future needs for the control theory in industries – report and topics of the control technology survey in Japanese industry. **Journal of Process Control**, v. 8, n. 5-6, p. 369 – 374, 1998.

THE MATHWORKS INC. Disponível em: < <http://www.mathworks.com/>>. Acesso em: 25 de Junho de 2007.

VERNER, I. M.; AHLGREN, D.J. Conceptualising educacional approaches in introductory robotics. **International Journal of Electrical Engineering Education**, v. 41, n.3, p. 183 – 201, 2004.

WILSON, J.Y.; HAVEWALA, A. **Building Powerful Platforms with Windows CE**. Reading: Addison-Wesley Professional, 2001

## **A LEARNING ENVIRONMENT FOR CONTROL AND AUTOMATION AND ITS APPLICABILITY IN DEVELOPPING COMPETENCIES AND ABILITIES**

**Abstract:** : *This work describes de aplication of educacional enviroment for automation and control laboratory experiments, based on configware logic and hierarquically distributed processing, specialy for high performance tasks or easily adaptable to hardware and software demands. The described environment aims to be an open system, easily adpatble, low cost, with great processing capability. This environment was projected in support to undergraduate and graduate curriculum, althoug it can be use al in industry, allowing several practical experiments for diferente courses, or as plataform integrating several courses or even areas from distinct engeneering modalities. An application example for a control course is presented in which de learning environment is used speed motor control. As an integrating project, the environment is used as in a mobile robot platform. Finally, we discuss the curricular competencies evolved in the environment context.*

**Key-words:** *learning control environmment, reconfigurable logic, mobile robot platform*