

# MÓDULO LABORATORIAL PARA EDUCAÇÃO EM CONTROLE, EM TEMPO REAL, BASEADO EM LINUX/RTAI

RENATO T. DE CARVALHO, HUGO R. BALDIOTI, NATHAN DE L. SILVA, FRANCISCO J. GOMES

*Faculdade de Engenharia, Universidade Federal de Juiz de Fora*  
*Rua José Lourenço Kelmer, s/n - Campus Universitário Bairro São Pedro - CEP: 36036-900 - Juiz de Fora - MG*

*E-mails:* renato.carvalho@engenharia.ufjf.br, hugo.baldioti@engenharia.ufjf.br,  
nathan.lima@engenharia.ufjf.br, chico.gomes@ufjf.edu.br

**Abstract**— The present work details the design and implementation of a low cost laboratory plant, for control engineering education, utilizing the Linux/RTAI, a FOSS system, for real time control. The developed laboratory plant is a double tanks system, that allows simultaneous control of its level and temperature variables; in this first step it was implemented the level control. It was also developed an analog-digital-analog card totally based on the microcontroller PIC 16F877. The final results showed the feasibility of Linux/RTAI as an effective real time kern, freely distributed and utilized, for developing low costs laboratories modules for control engineering education, as well encourages further developments on the developed module, including other experimental developments.

**Keywords**— Linux, RTAI, Control Education, FOSS, Real Time Control

**Resumo**— O presente trabalho tem como objetivo mostrar o desenvolvimento de um módulo laboratorial para educação em controle de processos, em tempo real, baseado no núcleo Linux/RTAI, um sistema FOSS. O módulo laboratorial consiste de um sistema de tanques duplos, que possibilita o controle de seu nível e temperatura, sendo que nesta etapa inicial foi implementado o controle de nível. Para realizar a interface do ambiente computacional com o módulo físico foi desenvolvido um sistema de aquisição e transmissão de dados, baseado no microcontrolador PIC 16F877A. Os resultados obtidos mostraram a factibilidade de se utilizar o Linux/RTAI como um núcleo de tempo real efetivo, de livre distribuição e utilização, na confecção de módulos didáticos de baixo custo para a educação em controle de processos, bem como encorajam desenvolvimentos posteriores no módulo e outros desenvolvimentos experimentais.

**Palavras-chave**— Linux, RTAI, Educação em Controle, FOSS, Controle em Tempo Real

## 1 Introdução

Governos de todas as partes do mundo começaram a lançar de iniciativas para colher os benefícios do software livre. Muitas dessas iniciativas ainda estão em fase inicial, mas há uma tendência significativa para a sua incorporação. O governo federal brasileiro é um grande incentivador do uso de padrões abertos, do licenciamento livre dos softwares e a formação de comunidades, em especial de usuários e desenvolvedores de software livre (<http://www.softwarelivre.gov.br/>). O governo federal brasileiro planeja migrar, até um percentual de 80 por cento, todos os computadores do estado e instituições de propriedade estatal para o sistema operacional Linux nos próximos três anos. A Faculdade de Engenharia da Universidade Federal de Juiz de Fora está adotando esta linha de trabalho e vem estabelecendo diretrizes para desenvolver sistemas laboratoriais de baixo custo para a educação em controle totalmente baseados em FOSS (*Free and Open Source Software*).

As vantagens ligadas ao software livre são muitas, podendo ser citada a possibilidade de desenvolver trabalhos e solucionar problemas de forma colaborativa, agregando os conhecimentos disponibilizados pelos fóruns de desenvolvedores e interagindo de

forma cooperativa com as redes ligadas aos diversos FOSS, sem qualquer custo ou contrapartida envolvida.

Outro aspecto que chama a atenção, adicionalmente à utilização e disseminação do FOSS, é o fato de que a educação em controle de processos ganha, continuamente, maior visibilidade, destacando-se o crescimento de artigos, congressos e periódicos na área, explicado por razões estruturais, ligados à sua base conceitual [Kheir,1996] e ao desenvolvimento das Tecnologias de Informação e Comunicação – TIC, com seu impacto sobre o desenvolvimento científico, tecnológico e conseqüências sobre a formação de especialistas na área [Murray,2002].

A engenharia é uma profissão prática, “hands-on” e, portanto, desde os primórdios da educação em engenharia, laboratórios didáticos têm constituído um dos fundamentos da graduação e, em muitos casos, da pós-graduação. Pode-se afirmar que, anteriormente à ênfase na componente científica, a maior parte da formação do engenheiro ocorre nos laboratórios [Feisel, 2005]. Por isto, sua utilização deve se associar a uma visão clara dos objetivos educacionais pretendidos em seu emprego e na formação propiciada pelo curso. Dentre estes objetivos Peterson [2002] destaca pontos que explicitam a importância da atividade laboratorial como, p.ex., entender e utilizar modelos conceituais; coletar, analisar e interpretar dados; exe-

cutar projetos sob demanda; aprender com falhas e erros; desenvolver a criatividade; comunicar, de forma oral e por escrito, os resultados e conclusões obtidos e trabalhar em equipe.

A utilização intensiva de laboratórios, contudo, esbarra em problemas que variam desde visões conceituais distorcidas dos educadores até questões operacionais práticas, como custos envolvidos com aquisição, manutenção, operação e atualização dos equipamentos, realidades estas comuns à nossa realidade e que remetem à busca de soluções alternativas ou complementares às atividades laboratoriais, sem perda ou alteração de seus conteúdos e objetivos.

Considerando-se estes aspectos, buscou-se associar os dois pontos abordados efetuando-se o desenvolvimento de módulos laboratoriais de baixo custo, que podem ser utilizados para educação em engenharia de controle, em tempo real, operando baseados inteiramente em softwares livres, ou seja, FOSS. A rede colaborativa, a ausência de custos, a oportunidade de intercâmbio de informações constituem, entre outros, fatores que justificam plenamente esta proposta, cujos resultados iniciais, de um módulo para controle de nível, são aqui apresentados.

O trabalho está estruturado como segue: a seção 2 apresenta o sistema computacional utilizado; a seção 3 mostra o desenvolvimento do software, do interfaceamento e a planta, objeto-alvo do controle; e finalmente a seção 4 os resultados obtidos são apreciados. As conclusões, na seção 5, encerram o presente trabalho.

## 2 Sistema Utilizado

Dentre todo o repositório de softwares livres disponíveis pela internet, talvez o mais notável é o Linux, por ser, em termos de *software*, o nível mais baixo num sistema computacional e possivelmente o com maior número de colaboradores ao redor do mundo ao logo de sua história. O módulo sugerido, como veremos, faz uso desse SO (Operating System) juntamente com um módulo que torna seu kernel *full preemptive*.

### 2.1 Linux

Linux é o nome que se dá ao núcleo (*kernel*) de (SO) desenvolvido pelo finlandês Linus Torvalds em 1993, inspirado no sistema MINIX, desenvolvido pelo prof. Andrew S. Tanenbaum. Embora Linus tenha dado o pontapé inicial, e seja a ele atribuído o seu desenvolvimento, o Linux foi e ainda é fruto de esforços de milhares de usuários e programadores ao redor do mundo. Por ser distribuído sob a licença GPL [GNU, 2007], qualquer pessoa pode modificá-lo para atender às suas necessidades. Assim, ele vem sendo adotado para diversas utilidades indo muito

além de computadores, passando para celulares, tablets e vídeo-games.

Apesar de algumas poucas distribuições terem sido feitas para suportar tarefas que necessitam de execução em tempo real, o Linux é não-preemptivo [Bovet & Cesati, 2000]. Por essa razão, o escalonamento seguindo tempo real tem tempo de execução não-determinístico, o que ofusca seu uso na indústria.

### 2.2 RTAI

RTAI [www.rtai.org] é um acrônimo para a expressão “*Real-Time Application Interface*”, um *path* para kernel *vanilla* (“puro”). Seu desenvolvimento se deu a partir do *Politecnico di Milano*, quando foi necessário realizar simulações de cálculos aeroespaciais utilizando Linux. Com esse *path* é possível diminuir o tempo de execução e troca de tarefas e diminuir o tempo de resposta à interrupções, tornando o sistema Linux preemptivo sem a necessidade de migrar de distribuição.

O RTAI possui uma API amigável que permite desenvolver rotinas de tempo real em C/C++ utilizando o GCC. Com um HAL (*Hardware Abstraction Layer*) que elimina funções *latency killers* nativas do Linux, como impressão de caracteres no dispositivo de saída e operações de leitura/escrita em FIFOs, e com modificação no escalonador do kernel, o RTAI torna o uso do Linux possível em operações na indústria de um modo geral.

## 3 Desenvolvimento

### 3.1 O módulo laboratorial

O módulo laboratorial desenvolvido é constituído por dois tanques (um reservatório e o outro controlável), duas bombas, duas válvulas (que compõe a válvula seletora de fluxo), um radiador (planejado para futuras aplicações de controle de temperatura), um computador e uma placa de comunicação entre o computador e o sistema físico, desenvolvida como parte integrante deste trabalho. Para sua execução foram utilizados materiais convencionais, de baixo custo, disponíveis no mercado. Assim, os tanques são tambores de alumínio, as bombas são automotivas de combustível e o sistema de refrigeração utiliza um radiador convencional. Os itens de maior custo são os sensores, de nível e temperatura (figura 2).

### 3.2 O sistema mecânico do módulo

O sistema mecânico da planta é constituído pelas bombas e pela válvula seletora de fluxo. O nível do tanque controlável é realizado pelas bombas, sendo uma responsável por encher o tanque e a outra por esvaziar. No sistema mecânico completo da planta existem dois fluxos independentes de líquido, um referente ao controle de nível e outro ao controle de

temperatura (nesta aplicação foi utilizado o fluxo referente ao nível do tanque).

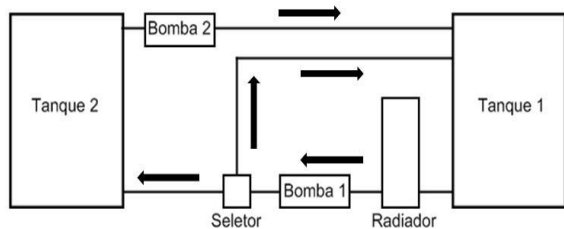


Figura 1. Esquema simplificado da planta com os fluxos

O fluxo controlado foi o referente à malha que engloba, nessa ordem, o tanque 2, a bomba 2, o tanque 1, o radiador, a bomba 1, o seletor e novamente o tanque 2. Nesse caso vale ressaltar que o tanque 2 é o reservatório, o tanque 1 é o controlável, ou seja, onde se deseja ter o controle do nível, o radiador fica desligado (não interferindo no sistema). Existem duas formas de esvaziar o tanque 1 para se efetuar o controle de nível, sendo que ambos dependem do posicionamento dos tanques. Se o tanque 2 estiver numa altura menor que o tanque 1 pode-se economizar energia e esvaziar o tanque 1 pelo princípio dos vasos comunicantes, sem a necessidade do uso da bomba. Caso os tanques estejam na mesma altura a bomba 1 é ativada para esvaziar o tanque 1. Se houver necessidade de esvaziar com uma velocidade maior o tanque controlável, a bomba 1 é ativada, independentemente da altura dos tanques.



Figura 2. A planta real

### 3.3 A placa de comunicação

Para se realizar o controle em tempo real foi necessário efetuar o desenvolvimento de uma placa de comunicação que realizasse o processo de interface entre o computador e o módulo laboratorial utilizado. Para isto, foi utilizado um microcontrolador PIC 16F877. A comunicação da placa com o computador é através da porta serial RS-232 que é um padrão para troca serial de dados binários. Para que isso fosse possível foi utilizado um circuito eletrônico que

converte os sinais vindos da porta serial do computador, que variam entre  $-12V$  e  $+12V$ , para sinais compatíveis com o padrão TTL de  $5V$  do microcontrolador. Uma foto do sistema desenvolvido pode ser vista na figura 3 a seguir:

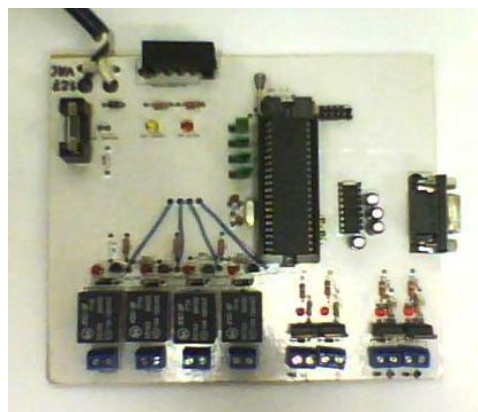


Figura 3. Interface de comunicação desenvolvida

Para o fechamento da malha foram utilizados dois sensores ópticos de distância, modelo GP2Y0A21YK, cuja saída analógica em função da distância de reflexão é mostrada na figura 4 que apresenta comportamento totalmente não linear, o que introduz complexidades adicionais para implementação do controle em malha fechada para o sistema.

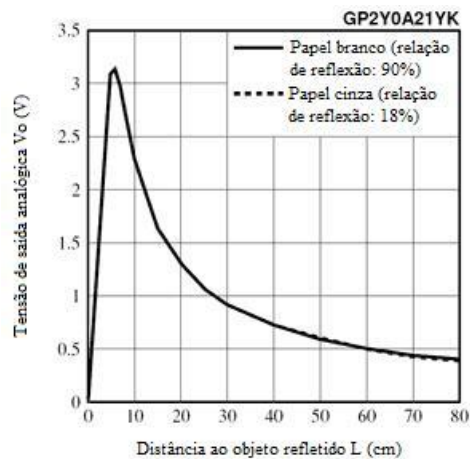


Figura 4. Gráfico do Sensor- Tensão de saída pela distância

O microcontrolador efetua a aquisição destes dados analógicos, efetua sua digitalização e envia para o computador, onde são processados, em tempo real pelo RTAI sendo a resposta enviada para os atuadores, utilizando para isto o canal digital analógico da interface desenvolvida. Os atuadores usados para o controle de nível são duas válvulas de 12VCA, e duas bombas 12VCC. Para controlar a vazão das bombas é utilizado o PWM (*Pulse Width Modulation* ou Modulação de Largura de Pulso) que é uma maneira simples de se controlar motores. Vale ressaltar que a placa desenvolvida permite, adicionalmente à utilização dos quatro atuadores, que outros quatro

possam ser utilizados, sendo dois com alimentação 12VCA e os outros dois com alimentação 12VCC.

### 3.3.1 Microcontrolador

Na placa de aquisição desenvolvida foi utilizado, como base, o microcontrolador PIC16F877A, desenvolvido pela Microchip. Pode-se observar na Figura 5, o diagrama de pinos do microcontrolador utilizado, que possui 8 conversores ADC de até 10 bits, 2 módulos CCP (Comparação, Captura e PWM) e periférico de comunicação serial.

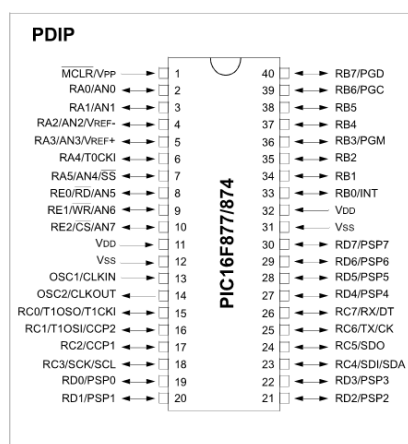


Figura 5. Diagrama de Pinos - PIC16F877A

No sistema desenvolvido, o microcontrolador é responsável por converter os valores analógicos dos sensores em valores digitais, enviar e receber os dados do computador através da porta serial e acionar todos os atuadores da planta.

### 3.3.2 PWM

A modulação por largura de pulso ou PWM (*Pulse-Width Modulation*) é uma técnica comumente utilizada em circuitos digitais para controlar a velocidades de motores CC; outra técnica possível para controlar a velocidade de motores seria o controle analógico, mas para o presente trabalho optou-se pela técnica PWM, pois esta implementação foi possível utilizando apenas um bit, enquanto uma técnica analógica exigiria bits adicionais, e consequentemente um conversor digital/analógico.

O PWM gera um ciclo de ondas quadradas, como mostrado na Figura 6, e a variação da largura do pulso possibilita que ocorra uma variação da tensão média aplicada ao motor, conseguindo assim controlar sua rotação. Esta técnica, pela simplicidade e facilidade de implementação, foi selecionada para o presente trabalho.

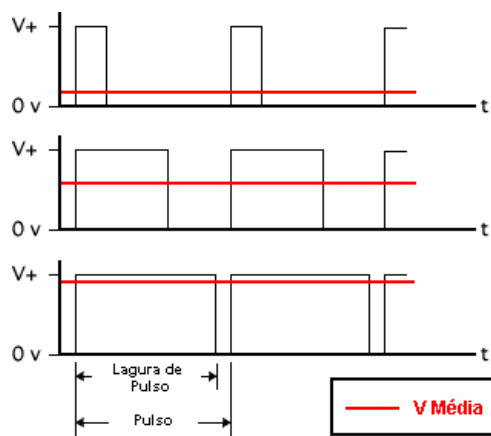


Figura 6. Exemplo de PWM

### 3.4 Sensor

O sensor de distância utilizado nesse trabalho foi escolhido por sua facilidade de uso, pois o fabricante garante seu funcionamento entre as distâncias de 10 a 80 centímetros do líquido e gera valores analógicos de tensão em sua saída correspondentes à distância, de acordo com a figura 4. Trata-se de um sensor IR (infra-vermelho), mais econômico que um sonar e mais eficiente que outros IR's alternativos. Maiores informações podem ser obtidas em: <http://www.pololu.com/catalog/product/136>

A relação matemática entre a tensão de saída do sensor versus a distância a que se encontra do objeto refletido (mostrada na figura 4) é claramente não linear, porém esta dificuldade pode ser atenuada utilizando-se a relação entre a tensão de saída e o inverso da distância medida, o que gera uma curva praticamente linearizada na faixa de trabalho do sensor. De acordo com o fabricante, uma relação linear mostra a distância do objeto refletido é aproximadamente igual a um fator de escala constante ( $\sim 27 \text{ V.cm}$ ) dividido pela tensão de saída do sensor.



Figura 7. Sensor IR utilizado

### 3.5 O sistema de controle

O programa de controle do Módulo Laboratorial é composto por 3 partes distintas integradas. A primeira é constituída por um processo executado no espaço do usuário que é responsável pelo processamento dos sinais gerados pelos sensores através de um filtro noise-spike. Essa parte também possui um PID digital através do qual a resposta às bombas pode ser tratada de forma a otimizar o controle do nível. A segunda parte é uma GUI (*Graphical User Interface*)

que torna o controle por parte do usuário amigável por mascarar os recursos de programação da parte anterior através de uma interface gráfica construída com widgets GTK (GPL). Nessa interface, o operador pode inserir as constantes do PID ( $K_p$ ,  $K_i$ ,  $K_d$ ), o nível de líquido desejado e o controle manual. Também é possível visualizar, em uma segunda janela, o gráfico da evolução do nível do líquido, conseguido através da biblioteca PLplot (GPL). A terceira parte é um conjunto de FIFOs que são responsáveis pela comunicação entre a GUI e o processo de controle da planta (uma FIFO para a comunicação GUI-Processo, e outra FIFO para a comunicação no sentido contrário).

### 3.6 O PID

Para obter melhor uso da planta, foi utilizado um PI digital, sendo executado em tempo real na tarefa principal do sistema de controle da planta. Seu código é uma implantação da equação

$$\Delta V_n = k_p (1 + k_i) e_n - k_p e_{n-1},$$

onde  $\Delta V_n$  representa a variação necessária na vazão  $V$  da bomba no instante  $n$ , e  $e_{n-1} = h_f - h_{n-1}$  o erro, definido como a diferença entre o nível desejado e o nível medido pelo sensor no instante  $n-1$ .

### 3.6 A Interface Gráfica

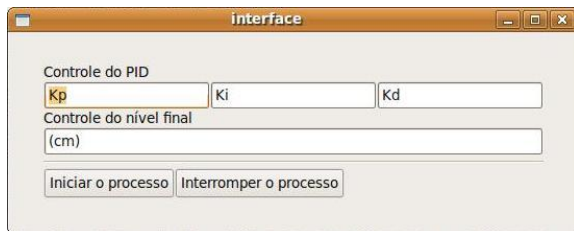


Figura 8. Interface gráfica do controle da bomba

No âmbito do controle, baseado em Linux/RTAI, as informações são adquiridas através da porta serial, processadas em tempo real e então enviadas para o atuador, efetuando o controle da bomba, em função dos dados adquiridos. Utilizou-se, nesta primeira tentativa de aplicação do sistema Linux/RTAI o controle PWM anteriormente descrito para o controle do nível do tanque. A implementação do sistema envolveu, entre outras dificuldades, a questão dos ruídos associados às malhas de controle de nível. Observou-se, assim, muita oscilação do sinal proveniente do sensor, decorrente não só de ruídos elétricos do sistema e do sensor, mas também da oscilação natural do líquido. (Esta situação pode ser vista na figura 10).

Na forma de um infograma, o núcleo do sistema em tempo real Linux/RTAI possui a seguinte estrutura (figura 9):

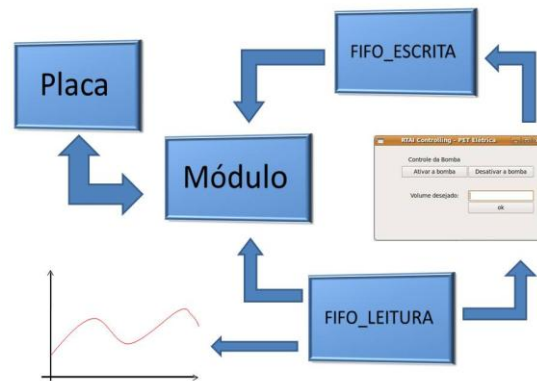


Figura 9. Infograma básico

A informação contida na “FIFO\_ESCRITA” é uma sequência de 48 bits na seguinte forma:

Tabela 1. Protocolo de escrita na FIFO

PWM (16)	Iniciar/interromper (8)	
$K_i$ (8)	$K_p$ (8)	$K_s$ (8)

A compreensão correta do protocolo é tarefa do Módulo de Controle.

As tarefas executadas pelo núcleo de tempo real podem ser visualizadas na seqüência de comandos a seguir, na forma de um pseudo código, que detalha as operações realizadas:

Módulo:

início:

```
(8 bits) dado
(8 bits) comando
ler_porta_serial(dado)
escreve_fifo_escrita(dado)
lê_fifo_leitura(comando)
envia_porta_serial(comando)
```

fim;

```
RTIME now;
rt_set_periodic_mode();
rtf_create(FIFO, 4000);
task = rt_task_init(nam2num("SERTST"), 1, 0, 0);
res = rt_sopen(COM1, BAUD, NUMBITS,
STOPBITS, RT_SP_PARITY_ODD,
RT_SP_NO_HAND_SHAKE,
RT_SP_FIFO_DISABLE);

rt_spclear_rx(COM1);
rt_spclear_tx(COM1);
rt_make_hard_real_time();
tick_period =
start_rt_timer(nano2count(TICK_PERIOD));
now = rt_get_time();
rt_task_make_periodic(&rt_task, now + tick_period,
```

```

tick_period);
    rtf_get(FIFO, &buffer, sizeof(buffer));
    controle[0] = buffer;
    rt_spread_timed(COM1, &controle[0], 1,
    nano2count(MS100));

```

## 4 Resultados

Foram realizados diversos testes para se verificar as condições de operação do sistema desenvolvido. Pode-se registrar algumas dificuldades associadas ao ajuste do sensor e dos ganhos do sistema, dada os ruídos presente nas malhas de controle de nível.

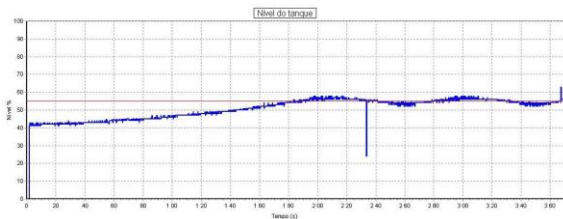


Figura 10. Resposta para alteração de nível - PWM

A figura 10 mostra o resultado obtido para o controlador no modo PWM quando foi inserida uma perturbação na forma da alteração de referência do tanque 1, que foi alterada de 20% para 40% do valor final. Pode-se observar que o sistema demorou cerca de 50s para alcançar a nova referência, valor este compatível com as constantes do tempo do módulo desenvolvido e componentes utilizados. Pode-se também observar a presença de ruídos nos sinais amostrados, bem como alguns “picos”, provenientes, provavelmente, de oscilações no sinal do sensor.

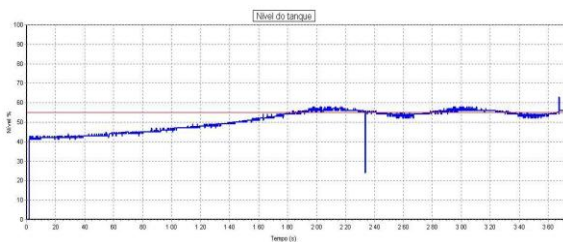


Figura 11. Resposta para uma alteração de nível - PI

O gráfico 11 mostra a resposta do sistema para uma alteração do nível de 40% para 55%, mas utilizando agora uma estrutura PI no algoritmo velocidade. Dadas as sintonias utilizadas, observa-se um comportamento muito parecido com o anterior, com estrutura PWM. Observa-se ainda que os resultados apresentam os ruídos característicos anteriormente observados. A figura 12 mostra a ação da bomba e o erro do sistema em relação à referência.

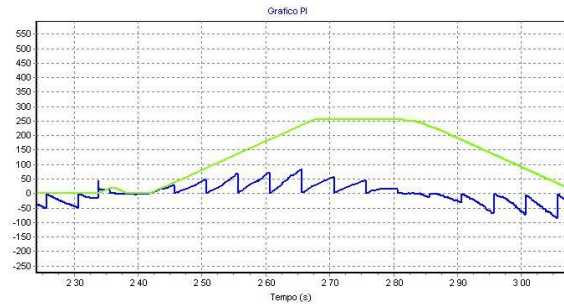


Figura 12. Resposta para uma alteração de nível - PI

## 5 Conclusão

Pode-se concluir pela total factibilidade do desenvolvimento de módulos de baixo custo, para operação em tempo real, que podem ser utilizados na educação em engenharia de controle. Merece destaque o fato que o sistema operacional utilizado, o núcleo Linux/RTAI, é FOSS, sendo, portanto, de livre distribuição e utilização, o que representa um ganho adicional para a proposta.

Adicionalmente às facilidades financeiras de obtenção e utilização do software livre no presente desenvolvimento, há que se caracterizar os aspectos positivos do FOSS, haja vista a construção do conhecimento compartilhado e o trabalho em rede. Assim, diversas dificuldades surgidas ao longo desta implementação puderam ser superadas através de troca de informações, compartilhamento de módulos, e mesmo auxílio “on-line” da comunidade ligado ao FOSS, particularmente ao Linux/RTAI. Fica claro que é possível realizar funções de forma equivalente ou até melhor do que as realizadas com softwares proprietários, geralmente utilizados em muitas das aplicações atualmente existentes nos laboratórios universitários.

Uma questão interessante sobre a utilização do Linux/RTAI é que os códigos ao final do trabalho não ficaram tão complicados e difíceis de entender porque, além da preocupação com a elaboração de um programa com características didáticas, de fácil compreensão, a linguagem utilizada para a programação facilitou no processo de desenvolvimento, tornando o código menor do que o esperado.

Os resultados obtidos por este trabalho, que terão continuidade no âmbito do curso de Engenharia Elétrica da UFJF, estimulam a utilização dos FOSS, nas diversas aplicações, quer nos ambientes laboratoriais acadêmicos, quer nos ambientes industriais, para desenvolvimento ou treinamento de operadores.

## Agradecimentos

Agradecimento ao Programa de Educação Tutorial do MEC/SESu pelo suporte para o desenvolvimento deste trabalho.

## Referências Bibliográficas

- “A Comunidade de Software Livre do Governo Federal – Portal Software Livre”, Disponível em: <http://www.softwarelivre.gov.br/comunidade-no-governo>
- “about\_rtai – RTAI – Official Website”, Disponível em: [https://www.rtai.org/index.php?module=pagema ster&PAGE\\_user\\_op=view\\_page&PAGE\\_id=2 &MMN\\_position=2:2](https://www.rtai.org/index.php?module=pagema ster&PAGE_user_op=view_page&PAGE_id=2 &MMN_position=2:2)
- “Ações – Portal Software Livre”, Disponível em: <http://www.softwarelivre.gov.br/planejamento-cisl/acoes>
- “International Free and Open Source Science Foundation”, Disponível em: <http://www.ifossf.org/>
- Feisel, L. D. and A. J. Rosa, “ *The Role of the Laboratory in Undergraduate Engineering Education*”, **Journal of Engineering Education**, January, p. 121-130, 2005.
- “FOSS Government Policy/Strategic Importance of FOSS – Wikibooks, collection of open-content textbooks”, Disponível em: [http://en.wikibooks.org/wiki/FOSS\\_Governmen t\\_Policy/Strategic\\_Importance\\_of\\_FOSS](http://en.wikibooks.org/wiki/FOSS_Governmen t_Policy/Strategic_Importance_of_FOSS)
- “Free Software Foundation”, Disponível em: <http://www.fsf.org/>
- GNU Project. Disponível em: [http://en.wikipedia.org/wiki/GNU\\_operating\\_system](http://en.wikipedia.org/wiki/GNU_operating_system). Acessado em 22/02/2010
- “Internacional Open Source Network”, Disponível em: <http://www.iosn.net/>
- Kheir, N. A, K. J. Åström, D. Auslander, K. C. Cheok, G. F. Franklin, M. Masten and M. Rabins, “*Control Systems Engineering Education*”, *Automatica*, vol.32, n.2, pp.147-166, 1996.
- Murray, M. R., editor, “*Future Directions on Control, Dynamics and Systems Report*”, Control in an Information Rich World, California Institute of Technology, June, 2002.
- “Open Source Initiative”, Disponível em: <http://opensource.org/>
- Peterson, G. D, Feisel, L. D., “*A Colloquy on Learning Objectives For Engineering Education Laboratories*”, Proc.American Society for Engineering Education, Annual Conference&Exposition, 2002
- “Pololu - Sharp GP2Y0A21YK0F Analog Distance Sensor 10-80cm”, Disponível em: <http://www.pololu.com/catalog/product/136>
- “Portal Software Livre”, Disponível em: <http://www.softwarelivre.gov.br/>
- “Richard Stallman's Personal Home Page”, Disponível em: <http://www.stallman.org/>
- Sedra, A. S. and Smith, K. C., 2007. *Microeletrônica*. 5th ed. Pearson Prentice Hall.
- “The Free Software Definition”, Disponível em: <http://www.gnu.org/philosophy/free-sw.html>