

UM LABORATÓRIO VIRTUAL PARA O ENSINO DE AUTOMAÇÃO INDUSTRIAL

DANIEL S. BOTELHO¹; MARCOS V. MOREIRA²

Escola Politécnica da UFRJ, Universidade Federal do Rio de Janeiro, Departamento de Engenharia Elétrica
21.945-970 – Rio de Janeiro – RJ

¹daniel.botelho@poli.ufrj.br, ²moreira@dee.ufrj.br

Resumo: Neste artigo é proposto um laboratório virtual para simulação dos experimentos atualmente realizados na disciplina Laboratório de Automação Industrial da Universidade Federal do Rio de Janeiro. A simulação dos experimentos tem como finalidade principal proteger os equipamentos de eventuais erros de programação que os alunos possam cometer ao transformar a lógica de controle, modelada utilizando-se redes de Petri, no código de programação Ladder a ser implementado em um controlador lógico programável (CLP). O laboratório virtual é simples de ser construído a partir de softwares disponibilizados pela maioria dos fabricantes de CLP's e pode ser usado facilmente pelos alunos possibilitando o teste de seus programas de controle.

Palavras-chave: Educação em controle, Laboratório Virtual, Sistemas a Eventos Discretos.

1. INTRODUÇÃO

Com o avanço da tecnologia e com o crescimento da economia mundial, este fortemente impulsionado pelo desenvolvimento das indústrias, é fato que o mercado industrial está cada vez mais moderno. A necessidade de se ter uma indústria que produza com rapidez, mais segurança e eficácia, vem ao longo dos anos transformando os processos industriais. Para tanto, a indústria inseriu em seu processo os controladores lógicos programáveis (CLP's), automatizando assim desde seus menores até os maiores sistemas de produção. Sendo o uso de CLP's indispensável para a indústria, a formação de profissionais devidamente capacitados para projetarem controles de sistemas de automação também se faz necessária.

Diversos laboratórios podem ser encontrados na literatura para o projeto de controladores baseados em sistemas dinâmicos de variáveis contínuas (SDVC) (Basilio, 2002; Basilio e Moreira, 2004; Mendez et al., 2006; Muskinja e Tovornik, 2006; Uran e Jezernik, 2008). Contudo, poucos artigos podem ser encontrados referentes ao ensino de Sistemas a Eventos Discretos (SED), ou seja, sistemas cujo espaço de estado é um conjunto discreto e cuja evolução dos estados é dirigida por eventos (Cassandras e Lafortune, 1999) e não pelo tempo como em um SDVC.

Em Marangé et al. (2007) um laboratório para um SED é proposto em que se controla remotamente uma planta real composta de um sistema de empacotamento de comprimidos. Visando garantir a segurança do operador e a integridade dos equipamentos, é proposto um filtro de validação para proteção do sistema caso haja um erro na programação do CLP utilizado no controle do processo. Esse filtro de validação é baseado na definição de restrições lógicas que não podem ser violadas. A idéia de proteção dos equipamentos e do operador é fundamental uma vez que, em geral, no processo de aprendizagem, os alunos cometem erros de programação que podem danificar equipamentos caros, incluindo o próprio CLP.

No Laboratório de Controle e Automação (LCA) da Universidade Federal do Rio de Janeiro, é ministrada, atualmente, para alunos de graduação em Engenharia de Controle e Automação e de Engenharia Elétrica, a disciplina de Automação Industrial cujo laboratório é proposto em Lima et al. (2006). Para execução das experiências propostas em Lima et al. (2006) é utilizado um CLP modelo SLC 500 da Allen Bradley (Rockwell Automation), dois conjuntos de esteiras transportadoras com sensores fotoelétricos e um conjunto de nove lâmpadas coloridas representando lâmpadas de semáforos. Os experimentos possuem crescente grau de complexidade e são todos realizados nas esteiras transportadoras ou com as lâmpadas coloridas representando semáforos.

Neste trabalho, é proposto um laboratório virtual para simulação de sistemas a eventos discretos. Este laboratório tem por finalidade testar o código de programação desenvolvido pelos alunos antes que este seja implementado em um CLP para controlar um sistema real, evitando assim possíveis danos aos equipamentos do laboratório devidos a erros na programação do CLP. O laboratório virtual proposto neste artigo, tem como base as experiências atualmente realizadas no Laboratório de Automação Industrial da UFRJ, reproduzindo virtualmente o comportamento das esteiras transportadoras e dos semáforos, e é desenvolvido utilizando um conjunto de softwares disponíveis pela maioria dos fabricantes de CLP's.

Este trabalho está estruturado da seguinte forma: na seção 2 é apresentada a plataforma virtual de simulação; na seção 3, o Laboratório de Automação Industrial da UFRJ é brevemente apresentado; na seção 4, um laboratório virtual para simulação dos experimentos atualmente realizados no Laboratório de Automação Industrial da UFRJ é apresentado; finalmente, na seção 5, são apresentadas as conclusões.

2. PLATAFORMA VIRTUAL DE SIMULAÇÃO

A plataforma virtual de simulação foi desenvolvida utilizando um conjunto de softwares disponíveis pela maioria dos fabricantes de CLP's. Como, atualmente, no laboratório da disciplina de Automação Industrial da UFRJ é utilizado um CLP modelo SLC 500 da Allen-Bradley (Rockwell Automation), os softwares utilizados são todos da Rockwell Automation. Contudo, uma plataforma de simulação poderia ser construída utilizando-se softwares de outros fabricantes. Em Adamo et al. (2007) um laboratório virtual é sugerido para o controle de nível e de temperatura de um líquido em um tanque. Para tanto, são utilizados os softwares *LabVIEW* (National Instruments) em conjunto com os seguintes softwares da Rockwell Automation: *RSLogix 500*, para a escrita do código de programação; e *RSLogix Emulate 500*, para emular o CLP que controla o processo virtual.

Neste trabalho, os seguintes softwares foram utilizados: *RSLogix 500*, utilizado para escrever o programa de controle em linguagem Ladder; *RSLogix Emulate 500*, utilizado para emular o CLP que controla o sistema virtual; *RSView32*, que é um software para construção de sistemas supervisórios, para construir a interface gráfica; e, para realizar a comunicação entre estes softwares é utilizado o software *RSLinx Classic*.

2.1 Fluxo de Dados na Plataforma Virtual de Simulação

Os softwares utilizados na plataforma virtual de simulação devem trocar dados a todo o momento. O fluxo desses dados obedece a uma hierarquia na qual um destes softwares gerencia todos os dados que circulam na rede de automação. Os dados são enviados de um software para outro em pacotes, respeitando o protocolo de comunicação, chamado de DF1.

As redes de automação utilizam um padrão chamado de OPC (*OLE for Process Control*), que estabelece regras para a comunicação de dispositivos de campo (CLP's, sensores, válvulas, etc) com sistemas de supervisão e gerenciamento industrial. Assim, os softwares utilizam para essa troca de dados um servidor OPC. Neste trabalho, foi utilizado o *RSLinx Classic* como servidor OPC, que gerencia a comunicação entre os softwares *RSLogix 500*, *RSview32*, e *RSLogix Emulate 500*.

A figura 1 mostra o esquema das interações entre os softwares utilizados para construção da plataforma de simulação.

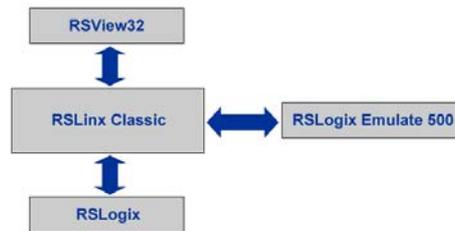


Figura 1. Diagrama de blocos da comunicação entre os softwares utilizados na plataforma virtual de simulação.

3. LABORATÓRIO DE AUTOMAÇÃO INDUSTRIAL DA UFRJ

O Laboratório de Automação Industrial da UFRJ é proposto em Lima et al. (2006) e, assim como em Moraes e Castrucci (2002), tem como objetivos principais aplicar as redes de Petri no projeto de automação baseados em modelos de SED e a prática com o hardware e o software dos CLP's.

Os experimentos sugeridos em Lima et al. (2006) possuem crescente grau de dificuldade, e são todos realizados com os seguintes equipamentos: (i) um CLP modelo SLC 500, composto de uma CPU (SLC 5/02), dois módulos analógicos, um de entrada e outro de saída, ambos com quatro canais, e um módulo de saída digital a relé; (ii) dois conjuntos de esteiras transportadoras, sendo cada conjunto composto de uma esteira deslizante feita de lona, dois sensores fotoelétricos com modo de detecção difuso, um motor interno para o arrasto da esteira e um painel, com uma fonte de 24V CC, para ligação dos componentes da esteira com o CLP; (iii) um conjunto de nove lâmpadas coloridas.

A seguir, uma breve explicação de três dos sete experimentos propostos em Lima et al. (2006) é feita, sendo um relativo ao controle das esteiras transportadas (experimento nº 7) e os outros dois relativos ao controle de semáforos (experimentos nº 4 e 6). Estes três experimentos são considerados em Lima et al. (2006) entre aqueles de maior nível de complexidade da disciplina Laboratório de Automação Industrial. Todos os demais experimentos podem ser simulados no mesmo simulador construído para esses três experimentos.

3.1 Experimento nº4: Sistema de controle de semáforos em um cruzamento

Esse experimento tem por finalidade implementar o controle de tráfego de um cruzamento. Independentemente de as vias terem tráfego nos dois sentidos, ou serem de mão única, o sistema de automação será o mesmo, desde que, no caso de fluxo nos dois sentidos, o tráfego seja interrompido ao mesmo tempo nos dois sentidos de tráfego. As exigências de projeto são as seguintes: (i) ambos os semáforos devem permanecer abertos (verde) por 30

segundos e devem ficar 5 segundos no amarelo; (ii) por questão de segurança, após um sinal fechar (passar para o vermelho) o outro só abrirá (passará a verde) 2 segundos após o primeiro ter fechado.

3.2 Experimento nº6: Semáforo para travessia de pedestres

Nesse experimento, o aluno deverá implementar um sistema de automação para executar o controle de um sistema de tráfego em um ponto de travessia de pedestre. O sistema possui dois semáforos, um para os veículos, com três lâmpadas sinalizadoras (vermelha, amarela e verde) e outro para pedestre, com duas lâmpadas sinalizadoras (vermelha e verde). O sistema a ser desenvolvido pelo aluno deve satisfazer as seguintes exigências: (i) o sinal para automóveis deverá permanecer aberto por pelo menos 1 minuto; (ii) quando o pedestre chegar à via, ele deverá apertar um botão, acionando o pedido de passagem; (iii) caso o sinal de automóveis já esteja aberto por mais de 1 minuto, esse deverá passar para o amarelo, permanecer amarelo por 5 segundos e, em seguida, passar para o vermelho; (iv) por razões de segurança, o sinal do pedestre somente mudará para verde 3 segundos após a mudança do sinal de automóveis para vermelho; (v) o sinal de pedestre deverá permanecer verde por 30 segundos; (vi) após esse tempo, a luz verde deverá apagar e a luz vermelha deverá, em seguida, piscar quatro vezes, em intervalos regulares de 1 segundo; (vii) ao final, a luz vermelha do sinal de pedestre deverá permanecer acesa e o sinal de automóveis deverá passar de vermelho para verde.

3.3 Experimento nº7: Sistema de manufatura com duas esteiras e dupla passagem do produto por um sensor

Esse experimento consiste em desenvolver um sistema de manufatura que desempenhe a seguinte seqüência: (i) quando ligadas, ambas as esteiras devem avançar no sentido de A para B; (ii) o objeto será sempre colocado no ponto A da esteira 1 e avançará no sentido da esteira 2, quando será detectado pelo sensor 1; (iii) caso a esteira 2 esteja desocupada, o objeto passará imediatamente para a esteira 2 e caso esteja ocupada, deverá ficar retido na posição do sensor 1 (desligando a esteira 1) e esperar até que a esteira 2 fique livre; (iv) na esteira 2, quando detectado pelo sensor 2, a esteira deverá parar por 10 segundos e, em seguida, avançar novamente; (v) quando detectado pelo sensor 3, a esteira deverá parar novamente, desta vez por 50 segundos; (vi) após esse tempo, a esteira deverá retroceder, levando o objeto novamente para a posição do sensor 2; (vii) quando detectado novamente pelo sensor 2, a esteira deve parar novamente por 10 segundos; (viii) após esse tempo, a esteira deverá avançar novamente; (ix) ao ser detectado pelo sensor 3, não deverá, desta vez, parar, sendo liberado da esteira 2, permitindo então que a esteira 1 entre em movimento, caso esteja parada com um objeto na posição do sensor 1, colocando um novo objeto sobre a esteira 2.

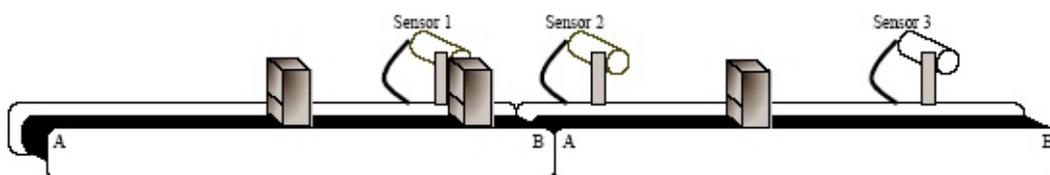


Figura 2. Sistema de manufatura com duas esteiras e três sensores. Figura retirada de Lima et al. (2006).

4. LABORATÓRIO VIRTUAL DE AUTOMAÇÃO INDUSTRIAL

Nesta seção são apresentados os passos para a construção do simulador para os experimentos apresentados na seção 3.

4.1 Interface Gráfica

O primeiro passo para construção da interface gráfica é a definição de rótulos para as variáveis que serão utilizadas no simulador. Esses rótulos são chamados de TAG's e são definidos no banco de dados denominado TAGDATABASE no *RSView32*. Se uma variável não for declarada neste banco de dados, a mesma não poderá ser utilizada pelo *RSView32*. O banco de dados de TAG's gerencia todos os endereços das variáveis que são utilizadas ao longo das telas da interface gráfica.

O segundo passo é a criação das telas da interface gráfica do simulador. O *RSView32* disponibiliza uma série de ferramentas, que quando combinadas, podem criar textos e objetos, e dar animação a estes objetos. Na figura 3 é apresentada a janela do *RSView32* para definição de propriedades associadas à animação de um objeto. Todo objeto criado na tela, apresenta propriedades, como cor, visibilidade e posição, que podem ser associadas a um TAG. Por exemplo, é possível associar a visibilidade de um objeto a um TAG, que por sua vez representa uma variável binária na lógica do simulador. Quando esta variável binária é verdadeira (=1), o objeto torna-se visível na tela; e quando é falsa (=0), o objeto torna-se invisível. No exemplo mostrado na figura 3, um TAG, *VISIBLE_CX2*, é responsável pela visibilidade de um objeto.

A figura 4 mostra a tela inicial do Laboratório Virtual de Automação Industrial criada no *RSView32*. Nesta tela o aluno deve escolher qual experimento deseja simular. Se o aluno selecionar o sistema de controle de tráfego, ele acessa uma tela onde o mesmo seleciona qual das experiências com semáforos deseja simular. Nas figuras 5 e 6, são apresentadas as telas do sistema de controle de semáforos em um cruzamento descrito na subseção 3.1 e do sistema de semáforos para travessia de pedestres descrito na subseção 3.2, respectivamente.

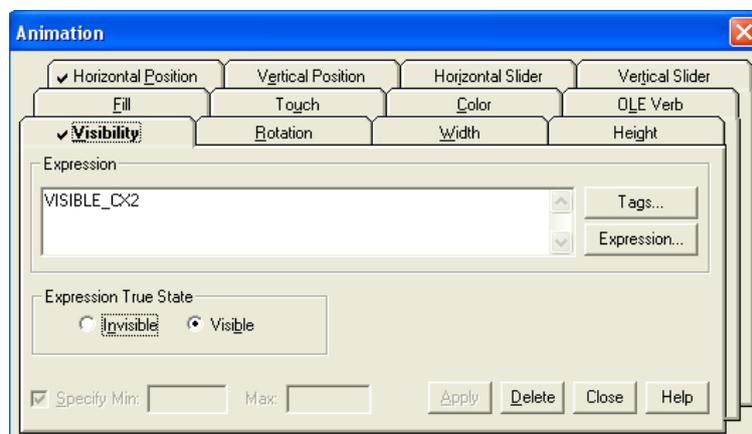


Figura 3. Propriedades de animação de um objeto no *RSView32*.



Figura 4. Tela inicial do Laboratório virtual.

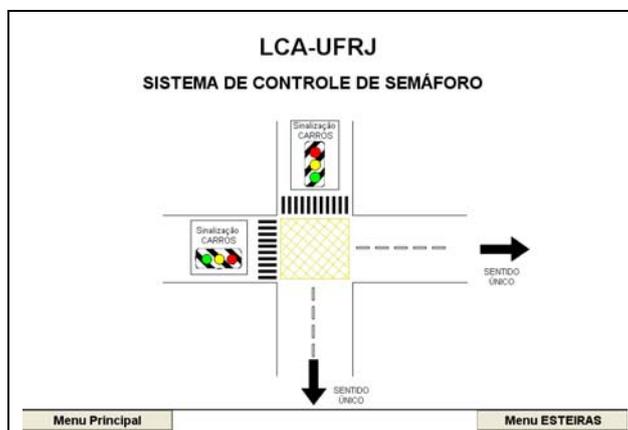


Figura 5. Tela do simulador do Sistema de Controle de semáforos em um cruzamento.

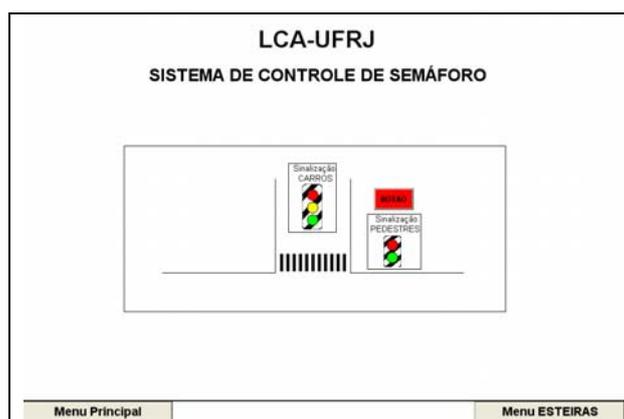


Figura 6. Tela do simulador do Sistema de Controle de semáforos para travessia de pedestres.

Se o aluno selecionar na tela inicial apresentada na Figura 4 o sistema de carregamento de esteiras descrito na subseção 3.3, é aberta a tela apresentada na figura 7.

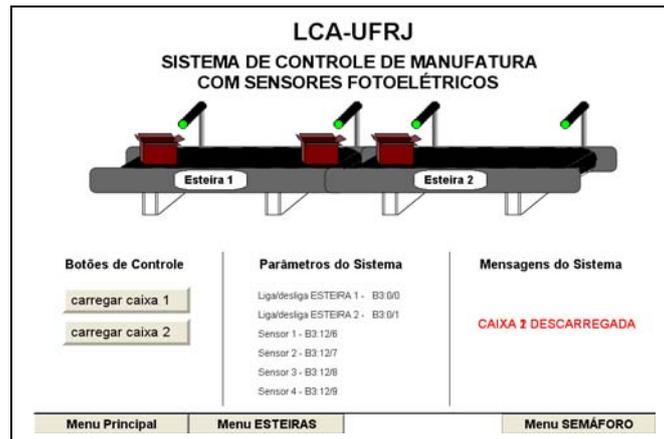


Figura 7. Tela do sistema de controle de manufatura com sensores fotoelétricos.

4.2 Diagrama Ladder da planta virtual

O simulador trabalha em conjunto com um programa Ladder, que é fornecido ao aluno para que este faça seu programa de controle dos sistemas virtuais. Para uma melhor organização do código Ladder do simulador este foi dividido em cinco subrotinas: (i) *PRINCIPAL*; (ii) *INPUT*; (iii) *ALUNO*; (iv) *OUTPUT*; (v) *SUP*. A função de cada uma dessas subrotinas é descrita a seguir.

4.2.1 Subrotina PRINCIPAL

A subrotina *PRINCIPAL* é responsável pela ordem de execução das demais subrotinas. A ordem de execução obedece ao conceito de varredura em controladores lógicos programáveis (Webb e Reis, 1999), ou seja, primeiro é feita a leitura das entradas através da subrotina *INPUT*, depois é executado o código de controle desenvolvido pelo aluno escrito na subrotina *ALUNO* e, em seguida, é feita a atualização das saídas pela subrotina *OUTPUT*. As variáveis de saída da subrotina *OUTPUT* são associadas a TAGs no programa supervisorio *RSView32* simulando o comportamento do sistema real.

Para simulação do experimento do sistema de manufatura com duas esteiras descrito na subseção 3.3, é necessário ainda a subrotina *SUP*, responsável pela animação das esteiras e pelas respostas dos sensores fotoelétricos.

4.2.2 Subrotina INPUT

A subrotina *INPUT* é responsável por atualizar as entradas do sistema de controle do aluno, ou seja, informações dos sensores para o experimento do sistema de manufatura com duas esteiras da subseção 3.3, ou do botão para travessia de pedestre do experimento do sistema de controle de semáforos descrito na subseção 3.2. Para o experimento da subseção 3.1 não há a necessidade da subrotina *INPUT*, uma vez que não são utilizados neste experimento sensores ou botões.

As informações dos sensores para o experimento do sistema de manufatura com duas esteiras são fornecidas pela subrotina *SUP*. Na figura 8 é apresentado o diagrama Ladder da subrotina *INPUT* para o sistema de manufatura com duas esteiras e quatro sensores fotoelétricos.



Figura 8. Diagrama Ladder da subrotina *INPUT* do sistema de manufatura com duas esteiras e quatro sensores fotoelétricos.

4.2.3 Subrotina ALUNO

Na subrotina *ALUNO*, o aluno irá programar seu código Ladder de controle do sistema virtual, utilizando, para tanto, os endereços das variáveis de entrada e saída fornecidos na tela do simulador. Essas variáveis são associadas às entradas e saídas da planta virtual, simulando os comandos e as respostas dos sensores do sistema real, respectivamente. As entradas e saídas da planta virtual são representadas no programa Ladder por endereços de memória B3:X/X, que correspondem a variáveis binárias no programa *RSLogix 500*. Quando o aluno receber o arquivo padrão em Ladder para simular seu projeto, o mesmo deverá somente editar a subrotina *ALUNO*.

4.2.4 Subrotina OUTPUT

A subrotina *OUTPUT* é responsável por atualizar as saídas do sistema de controle do aluno, que são por sua vez, as entradas da planta virtual. Para os experimentos dos sistemas de semáforos descritos nas subseções 3.1 e 3.2, as saídas são associadas ao acionamento das lâmpadas, enquanto que para o experimento do sistema de manufatura com duas esteiras da subseção 3.3, as saídas do sistema de controle do aluno são comandos para que os motores das esteiras sejam acionados movendo as caixas postas sobre as esteiras no sentido da esquerda para a direita ou no sentido inverso. Na figura 9, é apresentado o diagrama Ladder da subrotina *OUTPUT* do sistema de manufatura com duas esteiras e quatro sensores fotoelétricos.

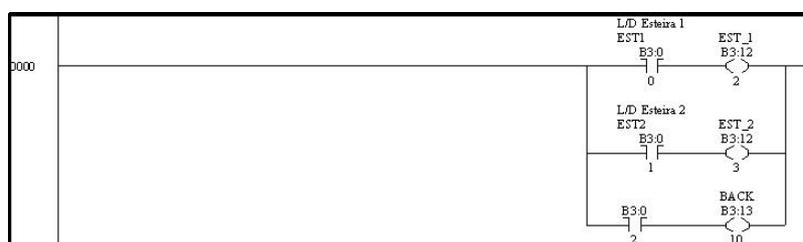


Figura 9. Diagrama Ladder da subrotina *OUTPUT* do sistema de manufatura com duas esteiras e quatro sensores fotoelétricos.

4.2.5 Subrotina SUP

Para simular os experimentos dos sistemas de semáforos não é necessária a subrotina *SUP*, uma vez que a subrotina *OUTPUT* comanda diretamente o acionamento das lâmpadas.

Assim, a subrotina *SUP* é utilizada somente na simulação dos sistemas de manufatura com as esteiras transportadoras. O código Ladder da subrotina *SUP* não é apresentado neste trabalho por uma questão de espaço.

Para a simulação dos sistemas com as esteiras, é importante que o aluno possa, na tela do simulador, escolher o momento do carregamento de caixas na esteira 1. Além disso, para testar a funcionalidade do programa de controle do aluno escrito na subrotina *ALUNO*, é necessário permitir um número mínimo de duas caixas circulando simultaneamente no sistema. Isso é feito por dois botões denominados *CARREGAR CAIXA 1* e *CARREGAR CAIXA 2*, apresentados na tela do simulador da figura 7. Quando o aluno seleciona o carregamento da primeira caixa no sistema, a caixa torna-se visível e o botão *CARREGAR CAIXA 1* torna-se inativo. De forma análoga, o mesmo ocorre com a segunda caixa e com o botão *CARREGAR CAIXA 2*.

A subrotina *SUP* é responsável também por dar movimento às esteiras virtuais dependendo dos comandos fornecidos pelo código Ladder de controle do aluno. Se o aluno comandar o acionamento das esteiras em um determinado sentido, é necessário que este movimento seja visualizado na tela do simulador. Na subrotina *SUP* a posição e velocidade das caixas nas esteiras são simuladas por temporizadores, onde o tempo acumulado destes fornece a posição das caixas em uma escala que varia de zero até um valor pré-determinado no temporizador, e a base de tempo do temporizador fornece a velocidade das caixas nas esteiras. É importante ressaltar que, como visto na seção 4.1, para que no simulador seja mostrado o movimento das caixas, o tempo acumulado dos temporizadores deve ser declarado como um TAG no *RSView32*. Este TAG, por sua vez, deve estar associado à propriedade de animação de posição horizontal da caixa na tela do simulador, e deve ser definido de forma que quando o acumulador alcance o valor pré-definido do temporizador, a caixa alcance o final da esteira.

Na planta real do experimento das esteiras da subseção 3.3 são utilizados quatro sensores fotoelétricos de modo difuso de detecção. Uma vez compreendido que os sensores são acionados em função da posição das caixas nas esteiras, torna-se simples condicionar na lógica Ladder do simulador quando os sensores deverão ser acionados. Como a posição das caixas em cada esteira é dada pelo tempo acumulado dos temporizadores, em uma escala que varia de zero a um valor pré-definido, associados, respectivamente, ao início e ao fim das esteiras, os sensores do simulador são acionados em função de determinados intervalos de tempo acumulado dos temporizadores.

Cada experiência proposta em Lima et al. (2006) possui uma característica própria de controle, e é razoável que o controle programado pelo aluno, e simulado na planta virtual, apresente erros. Para alguns erros de programação, identificados como os mais comumente cometidos pelos alunos, o aluno pode não só visualizar no simulador o mau funcionamento do sistema, como também é alertado de seu erro pelo sistema de simulação através de uma mensagem de erro na tela. As situações de erro de programação pelo aluno são identificadas pelo código Ladder do simulador, que através de um endereço de memória possibilita a visualização na tela do sistema de simulação. Por exemplo, se o programa do aluno para o controle do experimento descrito na subseção 3.3 permite a passagem de uma caixa da esteira 2 para a esteira 1, um endereço booleano de memória evolui ao seu nível lógico VERDADEIRO e uma mensagem de erro como a mostrada na figura 10 é apresentada na tela do simulador. Isto facilita



Figura 10. Mensagem de erro gerada na tela do simulador para o sistema de manufatura com esteiras e dupla passagem do produto por um sensor.

4.3 Implementação da lógica de controle no CLP

Uma vez simulado, o programa do aluno deverá ser implementado no CLP a fim de controlar o sistema real. Para isto, algumas poucas modificações devem ser feitas ao longo do programa do aluno. As modificações se limitam ao endereçamento das entradas e saídas da experiência simulada. No programa do simulador são utilizados endereços de memória (B3:X/X) para as entradas e saídas do sistema fornecidos aos alunos, enquanto no programa a ser implementado no CLP, são utilizados endereços diferentes (I:X/X; O:X/X) referentes aos módulos de entradas e saídas digitais e analógicas. Para implementar o sistema no CLP o aluno deve substituir os endereços de memória referentes a entradas, por seus respectivos endereços nos módulos de entrada digital e analógica. Analogamente, o mesmo tratamento deverá ser adotado com as saídas do sistema.

5. CONCLUSÃO

Neste trabalho é apresentado um laboratório virtual para o ensino de automação industrial tendo como base os experimentos atualmente realizados na disciplina de Laboratório de Automação Industrial da Universidade Federal do Rio de Janeiro. O laboratório virtual possui uma concepção simples e pode ser facilmente utilizado pelos alunos como uma etapa intermediária entre a modelagem do sistema a evento discreto e a implementação do código de controle no Controlador Lógico Programável para controlar o sistema real, evitando assim que erros de programação ocasionem danos aos equipamentos. Além disso, outros experimentos mais complexos podem ser simulados utilizando a mesma idéia proposta neste trabalho, sem que haja a necessidade do sistema real e do próprio controlador, o que possibilita uma maior flexibilidade para o ensino de sistemas a eventos discretos. Este simulador tem sido utilizado na disciplina de Laboratório de Automação Industrial com boa aceitação por parte dos alunos.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAMO F., ATTIVISSIMO F., CAVONE G. E GIAQUINTO N. SCADA/HMI systems in advanced educational courses, **IEEE Transactions on Instrumentation and Measurements**, v. 56, n. 1, p. 4–10, 2007.

BASILIO J. C. A Laboratory for a first course in control systems, **International Journal of Electrical Engineering Education**, v. 39, p. 54–70, 2002.

BASILIO J. C. E MOREIRA M. V. State-Space parameter identification in a second control laboratory, **IEEE Transactions on Education**, v. 47, p. 204–210, 2004.

CASSANDRAS C. G. E LAFORTUNE S. **Introduction to Discrete Event Systems**, Boston: Kluwer Academic Publishers, 1999.

LIMA S. T. S., DIAS J. R. S. E BASILIO J. C. Um laboratório para um curso de automação industrial utilizando redes de Petri, In: XVI CONGRESSO BRASILEIRO DE AUTOMÁTICA, 2006, Salvador. **Anais**. Salvador: UFBA, 2006. p. 3200–3205.

MARANGÉ P., GELLOT F. E RIERA B. Remote Control of Automation Systems for DES courses, **IEEE Transactions on Industrial Electronics**, v. 54, n. 6, p. 3103–3111, 2007.

MENDEZ J. A., TORRES S., ACOSTA L., SIGUT M. E MORENO L. A control engineering laboratory based on a low-cost non-linear plant, **International Journal of Electrical Engineering Education**, v. 22, p. 205–215, 2006.

MORAES, C. C. E CASTRUCCI, P. L. Um programa didático em automação insutrial, IN: XIV CONGRESSO BRASILEIRO DE AUTOMÁTICA, 2002, Natal. **Anais**. Natal: UFRN, 2002, p. 1397-1402.

MUSKINJA N. E TVORNIK B. Swinging up and stabilization of a real inverted pendulum, **IEEE Transactions on Industrial Electronics**, v. 53, n. 2, p. 631–639, 2006.

URAN S. E JEZERNIK K. Virtual laboratory for creative control design experiments, **IEEE Transactions on Education**, v. 51, n. 1, p. 69–75, 2008.

WEBB, J. W. E REIS, R. A. **Programmable Logic Controllers: principles and applications**, Upper Saddle River: Prentice Hall, 1999.

***Abstract:** In this paper, it is suggested a virtual laboratory for the simulation of experiments that are currently carried out at the laboratory of the Industrial Automation course of the Federal University of Rio de Janeiro. The main objective of the simulation of the experiments is to protect the equipments of programming errors that the students can eventually make transforming the control logic, modeled using Petri nets, in the Ladder control code to be implemented in a programmable logic controller (PLC). The virtual laboratory can be simply constructed with softwares that are available by the majority of PLC's manufacturers, and it can be easily used by the students for testing their control codes.*

***Key-words:** Control education, Virtual laboratory, Discrete Event Systems.*