

MODELAGEM E SIMULAÇÃO DE UM PROCESSO INDUSTRIAL

Adalmano O. S. M. Duarte – adalmano_duarte@yahoo.com.br

Gustavo C. Guedes – gcguedes@yahoo.com.br

Júlia D. Salgado – julia@ufpa.br

Orlando F. Silva – orfosi@ufpa.br

Pedro P. R. Junior – pauloreis01@hotmail.com

Ranieri B. Teixeira – ranieri@ufpa.br

Silvia R. C. Rosa

Universidade Federal do Pará. Departamento de Engenharia Elétrica e de Computação
Campus Universitário do Guamá
Rua Augusto Corrêa, 01
CEP 66075-110 – Belém –Pará.

Resumo: *A modelagem e a simulação de sistemas físicos são técnicas extremamente úteis para as engenharias e utilizadas em diversas situações. O domínio de tais técnicas permite a abstração das características essenciais de um sistema a um baixo custo pois a simulação, como resolução da modelagem, permite representar artificialmente um fenômeno real para a análise de seu comportamento dinâmico, comparar e identificar as soluções mais adequadas. Com a simulação, os alunos podem aplicar seus conhecimentos teóricos e aprofundá-los, bastando para tanto o domínio de uma linguagem de programação, de técnicas de cálculo numérico e da teoria associada ao problema em questão. O objetivo deste artigo é mostrar as etapas de desenvolvimento de um simulador de parte de um processo industrial, constituído de uma esteira transportadora sobre a qual recipientes devem ser preenchidos com um líquido proveniente de um silo. O controle de tal sistema seria realizado por um CLP (Controlador Lógico Programável). O ambiente de simulação usa interfaces gráficas e se baseia em programação orientada a objetos para criar um ambiente virtual representativo do processo industrial. O ambiente desenvolvido será utilizado em aulas das disciplinas de graduação em Engenharia Elétrica visando a motivação dos alunos, tanto sobre o uso de CLPs, como de técnicas de programação, modelagem e simulação de sistemas.*

Palavras-chave: *Modelagem, Simulação, Ensino, Orientação a Objetos, Java.*

1. INTRODUÇÃO

No âmbito da indústria, os controladores lógicos programáveis tem sido cada vez mais utilizados em substituição a sistemas de controle tradicionais, principalmente na implementação de sistemas de controle de operações lógicas seqüenciais, antes realizados por relés. Esta tendência é motivada pelo uso de computadores digitais e pela flexibilidade dos CLPs.

O processo consiste num conjunto de recipientes dispostos sobre uma esteira rolante, acionada por um motor. Um sensor de aproximação indica quando o recipiente alcança uma posição pré-definida abaixo de um silo que contém um determinado produto.

Através de interfaces gráficas geradas por programação orientada a objeto, será simulado o processo industrial explicitado.

2. CONTROLADORES LÓGICOS PROGRAMÁVEIS

O Controlador Lógico Programável (CLP) é um dispositivo eletrônico baseado em microprocessadores, destinado a controlar máquinas, processos industriais e outros, através do processamento dos sinais de entrada provenientes de chaves ou sensores, fornecendo como saída sinais utilizados normalmente para acionamento de algum elemento do processo. Utiliza uma memória programável para armazenar instruções e executar funções específicas como controle de energização/desenergização, temporização, controle Proporcional Integral Derivativo (PID), contagem, sequenciamento, operações matemáticas e manipulação de dados.

Os CLPs surgiram a partir de 1968 substituindo os sistemas de controle baseados em relés. Nas décadas de 70, os controladores passaram a ter microprocessadores, e a partir de 80, houve um aperfeiçoamento das suas funções, sendo utilizados também em redes de comunicação de dados.

Dentre as principais características dos CLPs tem-se:

- Linguagem de programação de alto nível, tornando o sistema bastante amigável ao operador, bem como a sua transferência e utilização em outro CLP;
- Simplificação nos quadros de painéis elétricos. Toda a fiação do comando fica resumida a um conjunto de entradas e saídas, tornando mais rápida e barata qualquer alteração necessária;
- Confiabilidade operacional, uma vez que as alterações podem ser realizadas através de programa aplicativo, a possibilidade de ocorrer algum erro é minimizada, gerando sucesso no desenvolvimento ou melhorias que se queiram implantar no processo industrial;
- Funções avançadas. Os controladores podem realizar grande variedade de tarefas de controle através de funções matemáticas, controle da qualidade, informações para relatório, e utilização em sistemas de gerenciamento da produção;
- Comunicação em rede. Através de interfaces de operação, controladores e computadores em rede permitem coleta de dados e um enorme intercâmbio de troca em relação aos níveis da pirâmide de automação.

Os CLP's são constituídos basicamente de uma fonte de alimentação; uma Unidade Central de Processamento; uma memória; dispositivo de entrada e saída e um terminal de programação.

Com a sua implantação no controle de processos, pode-se reduzir custos dos materiais, de mão-de-obra, instalação. Também, o CLP ocupa menos espaço que os controladores, temporizadores e outros componentes de controle utilizados antes do seu advento. Quanto às linguagens de programação de CLP's, tem-se basicamente: a linguagem LADDER (lógica de relés).

3. MODELAGEM DO SISTEMA

A modelagem é uma técnica extremamente útil para as engenharias. Seja ela conceitual ou matemática, esta sugere uma representação de um sistema. Sob o ponto de vista do funcionamento, de uma melhor compreensão e de possíveis alterações de um sistema, a modelagem matemática e a simulação são essenciais. Neste tópico será abordada a modelagem matemática de um processo industrial simples.

Para compreender e controlar sistemas complexos, deve-se obter modelos matemáticos quantitativos destes sistemas. Torna-se necessário analisar as variáveis do sistema e obter um modelo matemático. Quando os sistemas sob consideração são dinâmicos, as equações que os descrevem são equações diferenciais, além disso, se estas equações puderem ser linearizadas, pode-se utilizar a Transformada de Laplace para simplificar a solução das mesmas. Na prática, a complexidade dos problemas e o desconhecimento de todos os fatores pertinentes requerem muitas vezes a introdução de hipóteses simplificadoras relativas à sua operação. Em resumo, a abordagem aos problemas de sistemas dinâmicos pode ser listada como a seguir:

1. Definir o sistema e seus componentes;
2. Formular o modelo matemático e listar hipóteses simplificadoras necessárias;
3. Escrever as equações diferenciais que descrevem o modelo;
4. Resolver as equações em função das variáveis de saída desejáveis;
5. Examinar as soluções e as hipóteses;
6. Se necessário, reanalisar ou reprojeter o sistema.

3.1. Descrição geral do sistema

O processo a ser controlado consiste essencialmente de um silo, contendo um determinado material que se deseja escoar em quantidades pré-estabelecidas em recipientes que se deslocam através de uma esteira rolante acionada por um motor DC. A quantidade de material nos recipientes é determinada por uma balança digital, enquanto que a chegada dos recipientes na posição correta para serem preenchidos é indicada por um sensor de aproximação. O sensor e a balança enviam as informações ao CLP, o qual tem a função de recebê-las e processá-las de acordo com a lógica de funcionamento desejada. A partir daí, o CLP envia sinais de comando que concretizam esta lógica.

Utiliza-se um computador digital para simulação da dinâmica dos elementos constituintes do processo, com programação baseada na linguagem Java, onde recursos de animação gráfica são empregados na visualização do mesmo e uma interface gráfica de usuário para permitir o controle da simulação. A “Figura 1” mostra uma descrição geral do sistema.

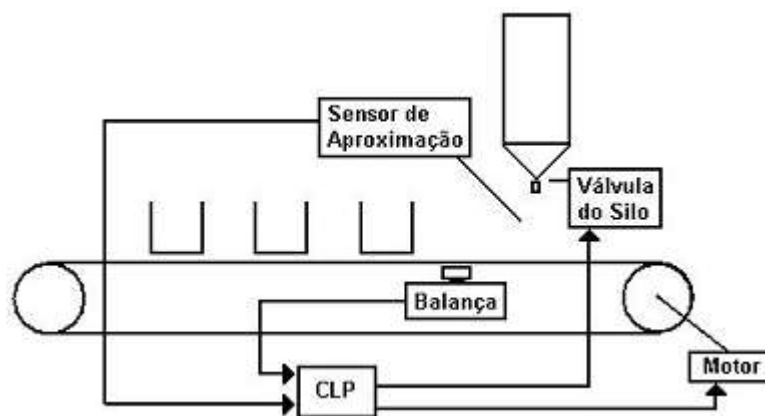


Figura 1 – Esquema geral do sistema

3.2. Modelagem Matemática dos Elementos do Processo

Silo

Um silo pode ser entendido como sendo um sistema de nível conforme mostrado na “Figura 2”.

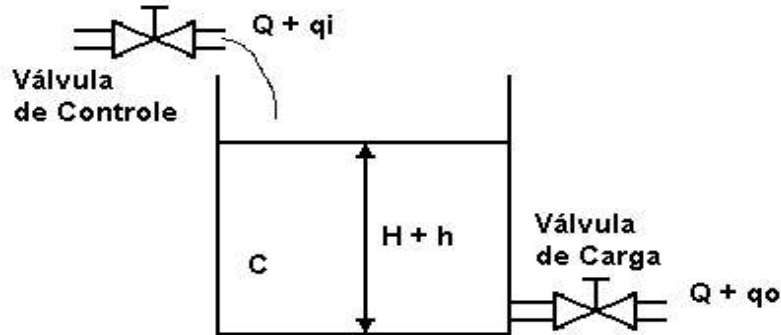


Figura 2 - Sistema de Nível

Onde se define :

Q = taxa de fluxo em regime estacionário[m^3/s];

q_i = desvio da taxa de fluxo de entrada em relação ao valor de regime estacionário[m^3/s];

q_o = desvio da taxa de fluxo de saída em relação ao valor de regime estacionário[m^3/s];

H = altura do nível em regime estacionário[m];

h = desvio na altura do nível em relação ao valor de regime estacionário[m];

$$\text{Resistência } R = \frac{\text{variação na diferença de nível, [m]}}{\text{variação na taxa de fluxo, [m}^3/\text{s]}}$$

$$\text{Capacitância } C = \frac{\text{variação no liquido armazenado, [m}^3\text{]}}{\text{variação na altura do nível, [m]}}$$

Baseado na hipótese de que o sistema seja linear (ou linearizável), e que o fluxo líquido durante um intervalo de tempo dt , é igual a quantidade armazenada adicional no tanque, então

$$C \times dh = (q_i - q_o) dt \quad (1)$$

De acordo com a referencia [1], $R = \frac{h}{q_o}$ a “equação (1)” fica

$$RC \times \frac{dh}{dt} + h = R \times q_i \quad (2)$$

considerando que não haja variação na taxa de fluxo de entrada ($q_i = 0$), a “equação (2)” torna-se

$$RC \times \frac{dh}{dt} + h = 0. \quad (3)$$

Cuja resposta temporal é dada por

$$h(t) = H_o \times e^{-t/RC}. \quad (4)$$

Em termos de vazão:

$$Q_o = \frac{H_o}{R} \times e^{-t/RC}. \quad (5)$$

Observa-se que a constante de tempo do sistema é RC.

Balança

A Vazão de saída dada pela “equação (5)” é usada para determinar o volume do líquido desejado no recipiente em um intervalo de tempo t. Então,

$$Vol = Q_o \times \Delta t \quad (6)$$

Onde Vol = volume do líquido [m³].

Na simulação da balança digital pra determinar o peso de material no recipiente, tem-se:

$$peso = m \times g \quad (7)$$

$$d = \frac{m}{vol} \quad (8)$$

$$peso = d \cdot vol \cdot g, \quad (9)$$

e substituindo vol por (6)

$$peso = d \cdot g \cdot Q_o \cdot t, \quad (10)$$

onde m é a massa do material [Kg] e d é a densidade do material [Kg/m³].

Motor DC

O motor DC para acionamento da esteira é do tipo controlado por armadura de acordo a “Figura 3”.

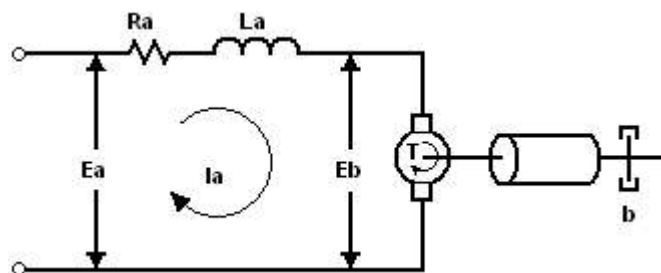


Figura 3 – Diagrama esquemático de motor DC controlado por armadura

Tendo-se a tensão de armadura Ea(s) como entrada e o deslocamento angular do eixo do motor θ(s) como saída, a função de transferência do motor, considerando desprezível a indutância do circuito da armadura [1] é dada por

$$\frac{\theta(s)}{Ea(s)} = \frac{K_{dc}}{s[\tau s + 1]} \quad (11)$$

Tomando-se como base os parâmetros do servo motor DC existente no equipamento *Mechanical Unit UM-154C*, de fabricação da *Feedback Instruments Limited*, tem-se $K_{dc} = 36,71$ e $\tau = 0,475$ seg.

Considerando que a tensão de armadura tenha um valor constante de 2.5V, então a velocidade angular do motor, ou seja, $\omega(s) = s \times \theta(s)$ é obtida a partir da “equação (7)” e dada por

$$w(t) = 2,5 \times K_{dc} \times (1 - e^{-t/\tau}). \quad (12)$$

Esteira Transportadora

Considerando o eixo do motor conectado diretamente ao disco dentado engastado na esteira, então desprezando quaisquer não linearidades, o deslocamento de translação da esteira em um intervalo de tempo t é definido por:

$$x(t) = R \cdot \Delta t \cdot w(t), \quad (13)$$

onde R é o raio do disco [m] e $w(t)$ é a velocidade angular [rad/seg].

Sensor de Aproximação

Para simular este sensor é estabelecida a posição em que o recipiente deve ser posicionado embaixo do silo, e como a posição do recipiente é dada pela “equação (9)”, então, quando a diferença entre a posição atual e a desejada for suficientemente pequena, significa que o sensor deve indicar que o recipiente alcançou a posição desejada.

3.3. Modelo orientado a objetos

O paradigma da orientação a objetos (OO) é uma maneira natural de descrever sistemas do mundo real para programar computadores, numa tecnologia conhecida como *Programação Orientada a Objetos* (POO). Uma das principais finalidades é reduzir a complexidade de um sistema, dividindo-o em blocos que apresentam comportamentos e estados próprios, com a capacidade de comunicação entre si, os *objetos*, que usam internamente programa estruturado em um contexto restrito, não essencial.

A *Unified Modeling Language (UML)* é uma linguagem gráfica que permite a modelagem de sistemas sob o foco da orientação a objetos. Trata-se de uma linguagem complexa, com muitos recursos. Aqui será usado um conjunto reduzido desses recursos, somente o necessário para permitir uma descrição simples do sistema, sob o ponto de vista de sua codificação em linguagem Java.

A necessidade de um modelo antes da programação

Em OO, um sistema é visto como um conjunto de objetos que interagem para resolver um problema. Os sistemas apresentam duas características essenciais: 1) a *estrutura*, que descreve os objetos do sistema e seus inter-relacionamentos e 2) o *comportamento*, que descreve como o sistema muda à medida que seus objetos interagem uns com os outros.

Quando há a necessidade de se programar um sistema real, com várias entidades, precisa-se adotar um modelo que forneça orientação sobre quais objetos serão utilizados, seus tipos, o que estes objetos devem fazer, como vão se comunicar, etc, antes de se começar a escrever o programa. Caso contrário, pode-se perder tempo ajustando programas falhos, sem coesão, por serem construídos às pressas. Para ter essa orientação, utiliza-se a UML.

Nesta seção deseja-se mostrar as etapas de modelagem de um sistema real, orientada objetos, até um ponto onde seja mais fácil a etapa de codificação em Java para se obter sua simulação. Assim, o modelo da estrutura do sistema pode ser feito por dois diagramas UML, que podem ser: 1) de classes; 2) de objetos. Para modelar o comportamento utilizam-se outros três diagramas UML: 1) de mapa de estados; 2) de atividades; 3) de colaborações.

Identificando as classes do sistema

Agora, é preciso identificar as classes do modelo de simulação do sistema. É feita uma releitura da descrição geral do sistema (item 3.1) e uma captura dos substantivos ou frases substantivas que possivelmente terão participação direta da simulação, desempenhando tarefas importantes. A seguir tem-se o resultado desse processo: Silo, Líquido, Recipiente, Esteira, Motor, Balança, Sensor de Aproximação, CLP e Válvula do Silo.

Esses nomes serão, provavelmente, classes ou instâncias de classes que precisarão ser implementadas para efetivar a simulação, embora possam surgir outros, na medida em que o projeto avança.

Identificando atributos e comportamentos de classe

As classes possuem *atributos* (dados), que representam o estado de um objeto e *operações* (comportamentos), que estabelecem o modo pelo qual os objetos se comunicam; diz-se que o comportamento de um objeto representa a sua colaboração para o sistema.

Identificam-se os atributos das classes lendo a descrição do sistema e retirando palavras ou frases descritivas. Nessa fase é muito útil considerar o modelo matemático do sistema, pois o mesmo determina vários atributos de cada componente do sistema.

Identificam-se as operações das classes a partir das ações que os objetos executam. Por exemplo, a esteira *transporta* os recipientes; o CLP *abre* a válvula do silo, etc. Também, nessa fase é essencial ler o modelo matemático; pois as equações estabelecem como os dados de uma classe mudam de estado. Naturalmente, as classes podem ter várias operações e atributos, embora seja possível descobrir novos.

Na “Figura 4” são mostradas as classes CLP, Esteira e Balança com alguns de seus atributos e operações, modelados segundo a UML.

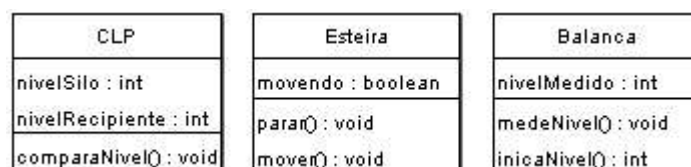


Figura 4 – atributos e operações de classes

Essas representações serão mais bem esclarecidas na próxima subseção.

Diagramas de classes e objetos

As classes representam uma abstração de entidades com características comuns, os objetos. Estes ocorrem, então, como instâncias específicas de classes. A UML procura refletir isso através dos diagramas de classes e de objetos.

Diagramas de classes modelam a estrutura estática do sistema, enquanto os diagramas de objetos representam essa estrutura em um momento específico da execução do programa. Por isso, diz-se que diagramas de objetos representam uma “fotografia” do sistema. Na “Figura 5” podem ser vistas as classes que foram identificadas até aqui, modeladas segundo a UML.

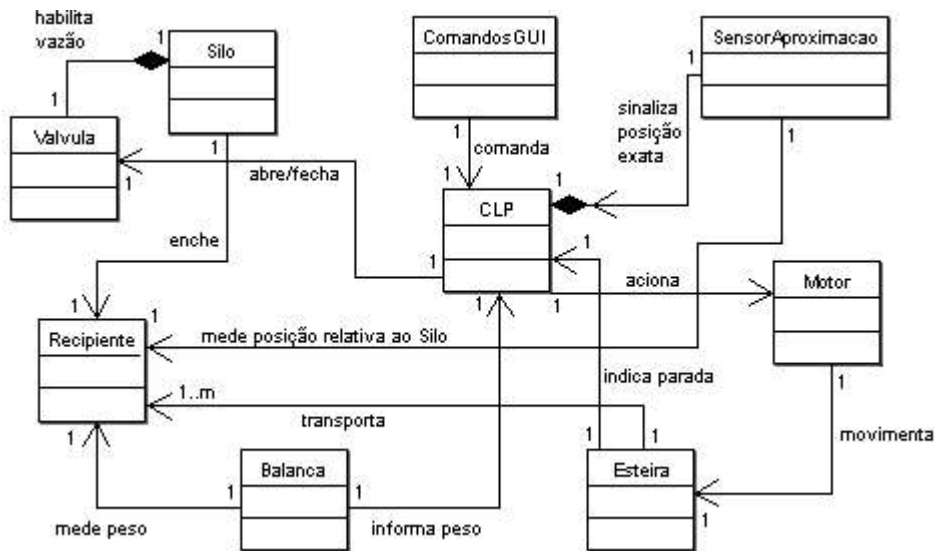


Figura 5 – Diagramas de classes do subsistema industrial

Cada classe é representada por um retângulo dividido em três regiões. A primeira contém o nome da classe; a intermediária contém os atributos da classe e a última contém as operações da classe. Essas regiões foram vistas com seus respectivos conteúdos na seção anterior, pode se útil revê-las.

As classes estão relacionadas por meio de associações, indicadas pela linha que as une. Os números junto às linhas expressam valores de multiplicidade, que indicam quantos objetos de uma classe participam da associação. As associações podem ser nomeadas e uma seta pode indicar seu sentido. No diagrama “Figura 4”, a associação entre as classes Silo e Recipiente pode ser lida assim: “Um objeto da classe Silo *enche* um objeto da classe Recipiente”. Observe que se considerou a multiplicidade dos objetos. O pequeno losango colocado na classe Silo indica que essa classe tem um relacionamento de agregação com a classe Válvula, ou seja, um relacionamento todo/parte. A classe que tem o losango é o todo (Silo) e a classe na outra extremidade à parte (Válvula). A expressão “tem um” define a agregação. Assim, diz-se que o Silo *tem uma* Válvula.

Diagramas de mapa de estados

Os objetos em um sistema têm estados que mudam em resposta a mensagens enviadas por outros objetos no sistema. Essa mudança de estado é refletida pela mudança dos valores de certos atributos de classes. Por exemplo, o CLP envia a mensagem “abreValvula” e um objeto da classe Válvula, que atende à mensagem, altera o valor de seu atributo do tipo boolean *aberta* para *true*, um valor adequado. Isso ocorre entre os demais objetos do sistema. Em UML os estados são representados, em um diagrama de estados, por um retângulo com cantos arredondados, conforme mostra a “Figura 6” para o atributo movendo, do tipo boolean de um objeto da classe Esteira.

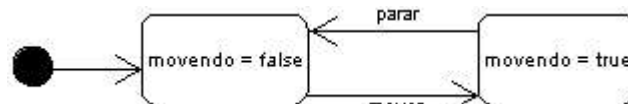


Figura 6 – Diagrama de mapa de estados para um objeto de Esteira

O nome do estado ocorre dentro do retângulo; um círculo cheio, com uma seta, indica qual o estado inicial do objeto. As setas entre os estados representam as transições de estado

que ocorrem, como visto anteriormente, em resposta a uma mensagem. O nome da mensagem que causa a transição é escrito junto à linha da respectiva transição.

Diagramas de atividades

Para se ter um entendimento claro do o que um objeto faz, suas atividades são especificadas através de diagramas de atividades, que modelam o fluxo de trabalho de um objeto durante a execução do programa. Trata-se de um fluxograma que especifica as ações que um objeto deve executar e em que ordem.

Em UML, as atividades de um objeto são representadas em elipses. A “Figura 7” mostra o diagrama de atividades de um objeto da classe Silo.



Figura 7 – Diagrama de atividades de um objeto de Silo.

O nome da atividade ocorre dentro da elipse. Uma linha, com seta na extremidade, liga duas atividades, indicando a ordem em que as mesmas devem ser executadas. O círculo cheio indica a primeira atividade e círculos concêntricos indicam a última atividade. O losango indica um desvio condicional e as frases entre colchetes indicam que a transição ocorre se a condição especificada for satisfeita.

Diagramas de colaborações

Em OO os objetos se comunicam para realizar uma tarefa. Quando essa comunicação é feita entre dois objetos dizemos que eles *colaboram*. A colaboração consiste numa mensagem particular que um objeto de uma classe envia para um objeto de outra classe invocando uma operação desse outro objeto. A UML especifica os diagramas de colaborações para modelar esse tipo interação. Com esses diagramas, consegue-se mais uma representação do comportamento do sistema, obtendo-se informações sobre como os objetos interagem. A “Figura 8” mostra um diagrama que representa as interações entre objetos das classes CLP, Motor e Esteira.

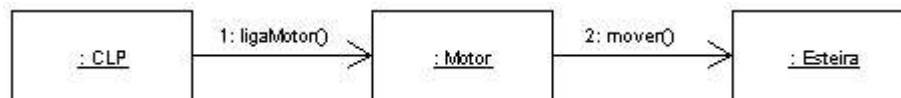


Figura 8 – Diagrama de colaborações entre objetos de CLP, Motor e Esteira.

A notação “: CLP” indica que se está considerando um objeto da classe CLP. As mensagens são passadas entre os objetos na direção da seta que os une. Os números indicam a ordem de envio das mensagens. O nome sobre a seta é de uma operação do objeto receptor, que deverá responder à mensagem, de acordo com o que lhe foi pedido.

3. SIMULAÇÃO

Segundo PEDGEN (1994) “Simulação é o processo de projetar um modelo de um sistema real e conduzir experimentos neste modelo com o propósito de entender o comportamento do sistema ou avaliar várias estratégias para sua operação”.

Para realizar a simulação de sistemas adota-se basicamente duas estratégias: simulação analógica e via computador digital. Uma terceira possibilidade seria a utilização dos dois recursos simultaneamente que é denominada simulação híbrida.

A simulação digital ou computacional consiste no processo de representar o modelo de um sistema em um computador e então experimentar como o modelo responderia a seguinte pergunta “o que aconteceria se (...)?”. Observa-se que é uma ferramenta excelente ao apoio na tomada de decisões uma vez que provê meios de responder atividades do projeto e operação de processos ou sistemas complexos de manufatura, com a visualização de sistemas do “mundo real”. Consiste nas seguintes etapas:

- Modelagem: construção do modelo;
- Experimento: Aplicar variações sobre o modelo construído, realizando alterações;
- Validação: Comparar os resultados obtidos com o modelo e a realidade.

4.1. Simulação de um CLP

Levando-se em conta que um CLP é, grosso modo, um dispositivo que recebe estímulos e responde a estes (pois possui um programa armazenado em sua memória), torna-se simples pensá-lo como o que ele realmente é: um objeto. A forma como ele responde aos estímulos corresponde ao seu comportamento (seus métodos), e o programa armazenado determina seus dados (atributos).

Para simulá-lo, uma representação virtual de seus métodos e atributos é feita utilizando-se as ferramentas discutidas neste trabalho (UML e Java). Determina-se com que outros objetos o CLP deve interagir (exemplos: verificar o estado do sensor de aproximação, ativar o motor da esteira, abrir ou fechar a válvula do silo) e de que forma isto deve acontecer. Por exemplo, se o sensor de aproximação sinaliza que o recipiente se encontra na posição adequada para o enchimento, então o objeto CLP deve: chamar o método “desligaMotor” do motor da esteira; chamar o método “abreValvula” da válvula do silo, para que o recipiente seja cheio; aguardar a sinalização do objeto “balança”, que indicará que o recipiente já contém o volume desejado do líquido; chamar o método “fechaValvula” da válvula do silo; reativar o motor (através do método “ligaMotor” de um objeto Motor) para que o ciclo se repita. Assim, modela-se e simula-se o CLP, de uma maneira dedicada, ou seja, apenas os serviços que ele oferece para a execução do processo considerado.

4.2. Simulação e ensino

O ensino em Engenharia consistia basicamente na transmissão de conhecimento, em que os alunos se tornavam meros receptores de informações. Essas informações eram memorizadas, repetidas através de mecanismos e aplicadas em técnicas para solução de problemas. Hoje, com a revolução tecnológica, o ensino teve que sofrer mudanças. Soluções para problemas considerados difíceis na Engenharia deveriam ser apresentados com rapidez. Assim, a simulação entra como uma importante ferramenta para lidar com essas situações, já que ela está associada a um software que imita o comportamento dinâmico de um sistema, fazendo a iteração entre os objetos que constituem o modelo e fornecendo soluções.

A simulação cria um ambiente agradável e de fácil manipulação, motivando a aprendizagem. Com ela, os alunos podem aplicar seus conhecimentos teóricos e aprofundá-los, bastando para tanto o domínio de uma linguagem de programação e domínio da teoria associada ao problema em questão.

4.3. Simulação do sistema em Java

Partindo-se do modelo UML, inicia-se o processo de codificação do programa. Com os diagramas UML, há uma grande facilidade em se determinar aspectos da estrutura e do comportamento do sistema. Quase de imediato conhecem-se fatores que determinam maior velocidade e redução de erros no momento de se escrever o programa em Java, tais como: a quantidade de objetos que participam de uma associação; as classes que devem ser implementadas; as interações entre objetos que precisam ser feitas; as transições de estados que um determinado objeto deve passar; as atividades que um objeto deve realizar, entre outras.

O resultado dessa codificação será um programa que quando executado refletirá o modelo subjacente. Mas, uma vez que o objetivo é a simulação, torna-se indispensável o uso de ferramentas gráficas; tanto para visualizar os elementos do sistema como para permitir o controle da simulação. A “Figura 9” mostra exatamente esse aspecto para a interface de simulação do programa resultante do modelo.

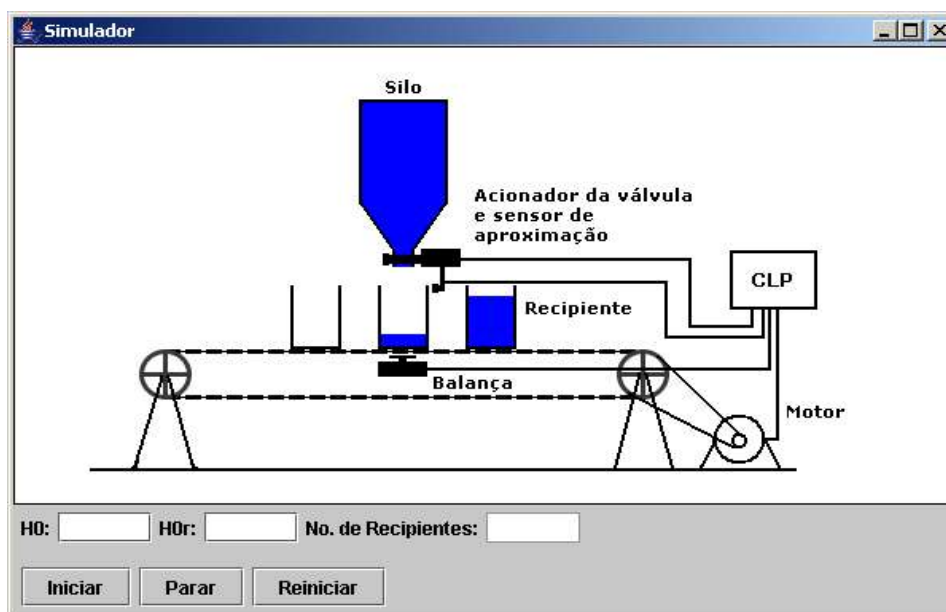


Figura 9 – Interface de simulação do sistema em Java.

6. CONCLUSÕES

Neste artigo foram mostradas técnicas de modelagem de sistemas para determinar uma melhor compreensão e de possíveis alterações do sistema proposto. Partindo de tal foram aplicadas técnicas de programação orientada a objeto para simular um controlador lógico programável num processo produtivo, criando um ambiente virtual com uma interface gráfica do processo, aonde os usuários poderão inserir dados referentes à planta industrial virtual, e o programa simulará o comportamento do processo.

O ambiente desenvolvido será utilizado em aulas das disciplinas de graduação em Engenharia Elétrica visando a motivação dos alunos, tanto sobre o uso de CLPs, como de técnicas de programação, modelagem e simulação de sistemas.

7. REFERÊNCIAS BIBLIOGRÁFICAS

BARREIROS, J. A. L., SILVA, O. F. & VIANA, S. A., **Modelamento Matemático Contínuo e Discreto de um Servo-Motor DC Através de Técnicas Off-Line e On-Line.** 1997.

DEITEL, H. M. & DEITEL, P. J., **Java, Como Programar.** – 4ªed – Porto Alegre: Bookman, 2003.

DORF, R. C., **Sistemas de Controle Moderno.** Rio de Janeiro: LTC, 1998.

FREITAS, P. J., **Introdução à Modelagem e Simulação de Sistemas com Aplicações em Arena.** Florianópolis: VISUAL BOOKS LTDA, 2001.

PEDGEN, C. D.; SHANNON, R.; SADOWSKI, R. P. **Introduction to simulation using SIMAN.** 2. 2d. New York, McGraw-Hill, 1994.

SANCHO, J. M. (Org.) **Para uma tecnologia educacional.** Porto Alegre: ARTMED, 1998.

***Abstract:** The modeling and the simulation of physical systems are extremely useful in engineering, being used in many situations. Such techniques allow a low cost abstraction of the essential features of a system because the simulation, as an implementation of a model, allows artificial representation of a real phenomenon, intending on the analysis of its dynamics, comparison and identification of the most adequate solution. Through simulation, students can verify theory and deepen their knowledge. For such a task, only some background in programming and numerical calculation techniques as well as the theory related to the problem in question are required. The aim of this paper is to show the development stages of a simulation system of part of an industrial process, which is constituted of a conveyer belt, over which recipients must be filled with a liquid from a silo. The control of such a system would be carried out by a PLC (Programmable Logical Controller). The simulation environment uses graphical interfaces and is based on object-oriented programming in order to create a representative virtual environment of the industrial process. Such an environment will be used for teaching in electrical engineering undergraduation courses for stimulating the use of PLCs, programming techniques, system modeling and simulation.*

Key words: Modeling, Simulation, Education, Object-Orientation, Java.