

DESENVOLVIMENTO DE AMBIENTE DE SIMULAÇÃO DE SISTEMA FLEXÍVEL DE MANUFATURA UTILIZANDO SOFTWARE SCADA

Horácio Medina da Rocha, horaciomedina@gmail.com¹

Lucas Nasciutti Prudente, lukaznp@yahoo.com.br¹

Eduardo José Lima II, eduardo@demec.ufmg.br²

¹Engenharia de Controle e Automação, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627 - Pampulha, CEP 31270-901, Belo Horizonte, Minas Gerais, Brasil

²Departamento de Engenharia Mecânica, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627 - Pampulha, CEP 31270-901, Belo Horizonte, Minas Gerais, Brasil

Resumo: *Sistemas Flexíveis de Manufatura (FMS - Flexible Manufacturing System) são extremamente caros, estruturalmente complexos e ocupam muito espaço físico. Uma planta, mesmo que didática, pode custar centenas de milhares de dólares, o que, aliado à complexidade e falta de espaço, muitas vezes torna impraticáveis pesquisas acadêmicas e aulas práticas utilizando tais sistemas. Este trabalho tem como objetivo simular uma FMS utilizando software SCADA, de forma que a planta virtual seja capaz de representar todas as ações de uma planta real, podendo ser empregada como uma ferramenta poderosa para estudo e pesquisa sobre automação. O software de simulação SCADA foi escolhido para facilitar a integração desta planta virtual com um CLP (Controlador Lógico Programável), que é responsável pelos comandos enviados à planta e pelos sinais recebidos da mesma. Para o projeto da simulação foram utilizados diagramas SFC (Sequential Function Chart) considerando que a atividade de cada componente da FMS pode ser representada por estados com transições e ações bem definidas, já que todos os sinais de entrada e saída utilizados são discretos. Outro fator que torna o uso do SFC ideal é o fato do software SCADA utilizado (Elipse E3) permitir a representação das transições dos estados através de condições booleanas e que cada componente da planta pode ser representado por objetos independentes, utilizando as funcionalidades da programação orientada a objeto. A simulação desenvolvida se mostrou fiel à planta real, permitindo o teste de programas do CLP e a verificação de técnicas de controle de planta.*

Palavras-chave: *Sistema de Manufatura Flexível, Simulação, SCADA*

1. INTRODUÇÃO

Sistemas Flexíveis de Manufatura (*Flexible Manufacturing System - FMS*) surgiram a partir de modificações nos sistemas de manufatura, como reação às necessidades específicas do mercado, sendo a principal respostas rápidas às necessidades de mudança do produto (atualizações, alterações nas funcionalidades, variedade, entre outras). Porém para que isto possa ser obtido com eficiência, estes sistemas são na maioria das vezes complexos tanto no nível dos equipamentos (normalmente de alta tecnologia) como no seu controle e operação.

Em sistemas flexíveis de manufatura, a implementação de sistemas supervisórios SCADA (*Supervisory Control And Data Acquisition*) e a programação de CLP's (Controladores Lógico Programáveis) são realizadas com base na experiência do programador e em seu conhecimento das etapas da manufatura. Essa implementação, porém, normalmente não é realizada seguindo uma metodologia formal, o que facilita a ocorrência de erros, bem como ineficiência e ociosidade de recursos. Adiciona-se a isso o fato de sistemas flexíveis de manufatura lidarem com um grande número de variáveis (tipos de matérias primas, tarefas a serem executadas, tipos de produtos acabados etc). A utilização de uma metodologia formal para a modelagem e a geração das regras para o sistema supervisório permitiria a diminuição de erros durante a implementação, a redução da ociosidade de recursos, bem como possibilitaria a utilização de técnicas de análise do funcionamento do sistema, como verificação de *dead-locks* etc.

Contudo, para se validar as metodologias estudadas, torna-se necessário o uso de plantas reais.

Este trabalho tem como objetivo principal simular um sistema flexível de manufatura (FMS) conhecido, utilizando para isto um software SCADA instalado num computador e um CLP. Com isto tem-se uma planta fisicamente fácil de operar, já que está simulada em um computador, e que não apresenta problemas físicos, como por exemplo, falhas elétricas e falhas mecânicas (estes podem ser simulados e testados em momentos mais oportunos). A planta virtual pode

então ser utilizada para o estudo e testes de metodologia de controle de sistemas flexíveis de manufatura integrados via sistema supervisório.

2. REVISÃO BIBLIOGRÁFICA

2.1. Laboratório virtual

A simulação de processos a cada dia se torna mais utilizada e importante no meio acadêmico e laboratorial, já que a evolução tecnológica anda a passos largos, e acompanhá-la, modificando as plantas reais, se torna complicado e caro. Existem plantas didáticas simplificadas com o controlador integrado a venda no mercado, porém elas não dão a oportunidade ao aluno de utilizar materiais utilizados no mercado real.

O enfoque apresentado em Galvez et al. (2006) segue uma filosofia diferente. A planta é implementada virtualmente em um computador pessoal, utilizando Visual Basic da Microsoft® como linguagem de programação, e a comunicação entre a planta virtual e o mundo real é realizada através de uma interface que utiliza o CLP ZAP500 da Hitecnologia®.

Como parte da filosofia, o processo simulado também é feito da maneira mais próxima do real possível, fazendo-se painéis de terminais dentro dos padrões industriais, tais como faixa de tensão e corrente dos sinais. A principal vantagem dessa configuração é que o usuário tem a oportunidade de interagir diretamente com o sistema de controle real (externo) e com uma planta virtual que se apresenta através de um painel de terminais (“conectados” aos seus sensores virtuais) exatamente como em uma planta real.

Em Galvez et al. (2006), a planta virtual implementada (Fig. (1)) representa uma autoclave e poderia ser utilizada, por exemplo, na esterilização de instrumental médico – odontológico em hospitais e consultórios.

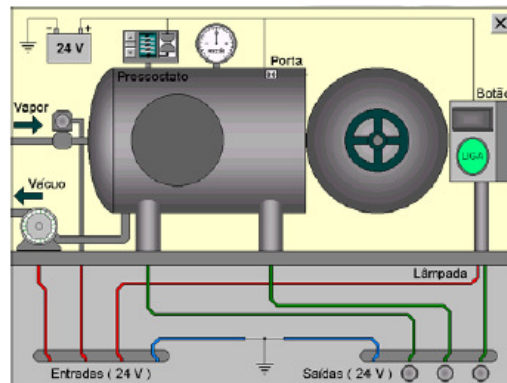


Figura 1. Planta Virtual na tela do computador

Fonte: Galvez et al. (2006)

Os recursos operacionais mínimos necessários para a utilização da planta virtual são: uma fonte de 24V para estimular as entradas E1, E2 e E3 do módulo externo de interface e algum dispositivo de leitura ou sinalização para as saídas S1, S2 e S3 (Fig. (2)). Esses recursos são bem mais acessíveis se comparado com o recursos que seriam necessários para acionar e operar uma planta real.



Figura 2. Terminais da Interface Externa

Fonte: Galvez et al. (2006)

Graças ao módulo de interface, a planta virtual responde a entradas reais e fornece sinais de saída reais, através do painel de terminais, dentro dos padrões industriais reconhecidos pelos dispositivos de controle comerciais. Assim, o controle da planta pode ser realizado através de dispositivos de controle externos, proporcionando ao aluno a experimentação das etapas de um projeto de controle como numa planta real. O sistema permite a implementação de funções de controle básicas e avançadas tais como: aquisição de dados, supervisão de variáveis, temporização de eventos, lógica de relés, controladores PID etc.

A principal característica do laboratório virtual proposto por Galvez et al. (2006) é que ele permite que toda a experimentação em automação e controle possa ser realizada em um ambiente real, apesar de a planta ser virtual.

2.2. Arquiteturas de Controle

Existem basicamente dois tipos de arquitetura de controle. A centralizada e a distribuída. A arquitetura centralizada é composta de apenas um elemento controlador que faz todas as ações de controle da planta. A principal desvantagem desta arquitetura é a dificuldade em se realizar modificações, limitada pela capacidade de processamento da unidade de controle (Matsusaki, 2004), além de problemas de controle afetarem a planta de forma mais dura, já que este controla toda a planta. Este tipo de arquitetura não é utilizada em sistemas manufatura de flexível.

Arquiteturas distribuídas são bem adequadas para FMS, já que não apresentam os problemas maiores da arquitetura centralizada, além de a FMS possuir diversos elementos que devem ter algumas de suas ações controladas localmente. Os dois tipos principais de arquitetura de controle distribuída são Arquitetura de Controle Hierárquico e Arquitetura de Controle Heterárquico.

2.2.1. Controle Hierárquico

Uma arquitetura de controle tipicamente hierárquica apresenta níveis de controle e contém vários módulos de controle dispostos em níveis, numa organização piramidal como ilustra a Fig. (3). Todas as atividades do nível inferior são determinadas pelo nível superior, caracterizando uma relação mestre-escravo (Matsusaki, 2004).

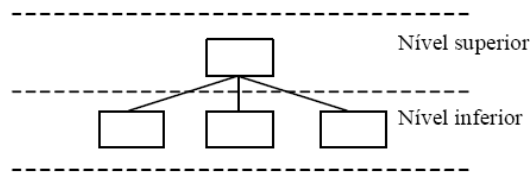


Figura 3. Arquitetura de Controle Hierárquica
Fonte: Matsusaki (2004)

Uma outra forma de Arquitetura Hierárquica é a hierárquica modificada Fig. (4), também conhecida como híbrida. A diferença desta está no nível de autonomia dos controladores subordinados. A conexão dos controladores dispostos em uma mesma hierarquia permite a resolução de problemas simples que dispensem a intervenção do controlador principal (Teixeira, 2006).

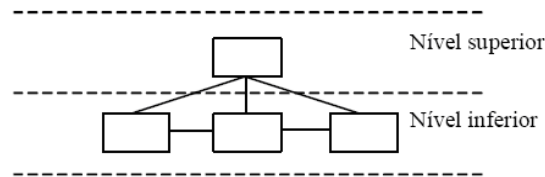


Figura 4. Arquitetura de Controle Hierárquica Modificada
Fonte: Matsusaki (2004)

O problema deste tipo de hierarquia é que uma modificação ou adição de um novo controle subordinado requer modificações nos níveis superiores, ou seja, quase sempre isto causa impacto em todo o sistema de controle. Pode-se também considerar que qualquer problema nos níveis superiores representa problemas nos níveis inferiores, havendo, neste sentido, herança dos problemas da arquitetura centralizada (Matsusaki, 2004).

2.2.2. Controle Heterárquico

O controle heterárquico é esquematizado na Fig. (5). Observa-se que não existem níveis de controle superiores e a cooperação entre as entidades é realizada através de negociação. A autonomia local implica em ter bases de informação locais, reduzindo ou eliminando a informação global e centralizada (Matsusaki, 2004). Com a tomada de decisão local e autônoma, o desempenho é otimizado, mas por outro lado a descentralização dificulta a análise sobre o contexto local, já que não existe uma relação hierárquica entre os elementos de controle (Teixeira, 2006).

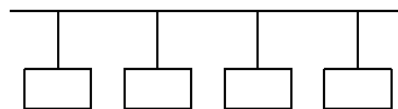


Figura 5. Arquitetura de Controle Heterárquica
Fonte: Matsusaki (2004)

3. A FMS REAL

O ambiente de simulação desenvolvido nesse trabalho foi baseado em uma planta real. Assim, os testes com o controlador feitos para a planta simulada poderão ser futuramente implementados também na planta real, validando tanto o estudo relativo à simulação de FMS quanto ao estudo sobre a metodologia de controle desenvolvida.

A planta estudada é do SENAI-CIMATEC da Bahia. Ela foi montada com o intuito de propiciar um ensino inovador de mecatrônica no estado. Apesar de ser uma planta com objetivos didáticos seu custo foi de aproximadamente US\$700.000,00, sendo, portanto, uma planta completa, apresentando todas as características de uma FMS industrial, além de contar com equipamentos de porte equivalente ao utilizado na indústria.

Esta planta é composta por um total de 7 subsistemas, além da esteira de transporte e o CLP principal, sendo que 2 deles foram excluídos da simulação considerando que a planta se mantém perfeitamente funcional sem estes subsistemas.

Os processos simulados são os seguintes:

- armazenamento,
- usinagem,
- teste,
- pintura,
- montagem.

A planta tem como entrada peças brutas (cilindros e chapas), todos de metal. Como saída tem-se uma peça composta por quatro peças, duas oriundas de cilindros brutos e duas oriundas de chapas brutas, sendo que estas peças montadas podem ter duas cores diferentes, azul ou verde. A Fig. (6) ilustra a produção da planta.

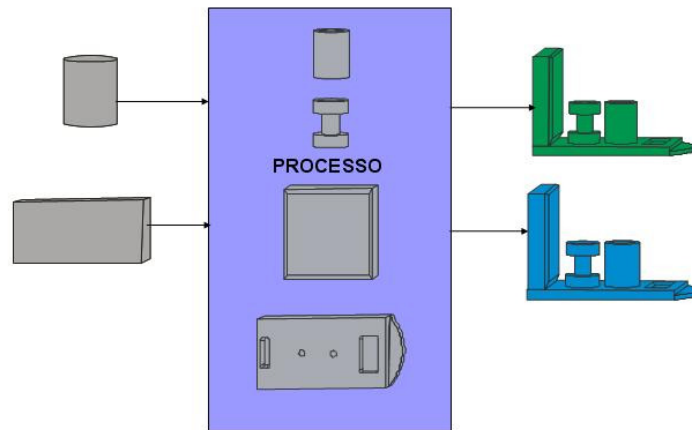


Figura 6. Funcionamento geral da FMS

Como exemplos, serão descritas e mostradas as simulações da estação FMS-1 e da esteira.

3.1. Esteira

A esteira transportadora é responsável pelo transporte de materiais entre todos os subsistemas da FMS. Para isto, ela possui até 32 vagões circulantes, que através de *pallets* levam peças em diferentes estados, desde brutas até montadas. Estes vagões são parados em cada estação (ou subsistema) através de *stoppers* locais. Estes são dispositivos pneumáticos que quando acionados impedem a passagem do vagão. Próximo a cada *stopper* existe um sensor responsável por identificar a presença de vagão parado e outros 5 sensores responsáveis por identificar o número do vagão. A Fig. (7) mostra em detalhe a esteira (1) um *stopper* (2), 5 sensores (3) para identificação do transportador, um sensor (4) para indicar presença do vagão e (5) a vista inferior de um vagão transportador.

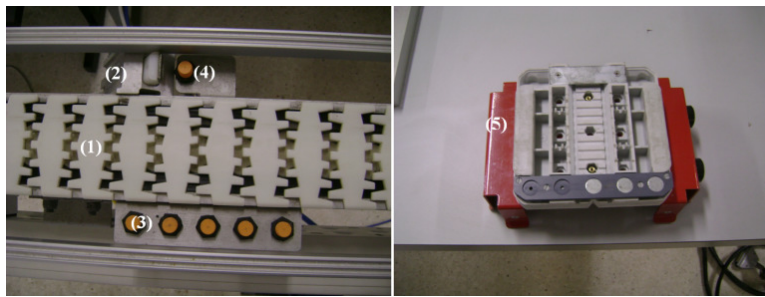


Figura 7. Esteira com *stoppers* e sensores e vagão transportador (vista inferior).

3.2. Estação FMS-1

O subsistema FMS-1 (a sigla neste caso significa *Flexible Machine System*, ou sistema flexível de usinagem) é responsável por receber as peças brutas e usiná-las, obtendo-se assim 4 peças diferentes. Esta estação é composta por um robô Mitsubishi RV-4A responsável pelo transporte interno das peças, por um torno e uma fresa ROMI, responsáveis por usinar as peças e por dois *buffers*, capazes de armazenar uma peça cada um (Fig. (8)).

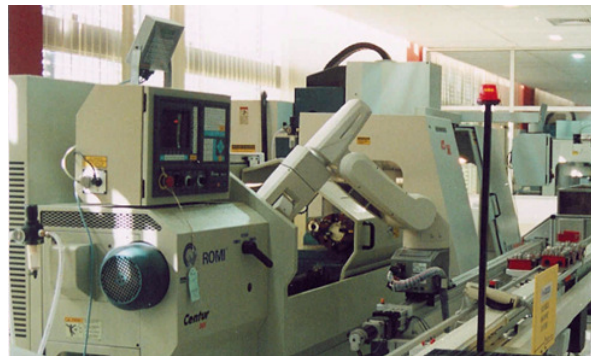


Figura 8. Estação FMS-1.

As peças que podem ser admitidas nesta estação são o cilindro bruto e chapa bruta. Se a peça for admitida, o CLP principal comandará um programa específico para o robô executar. Este pegará a peça com o *pallet* e colocará num *buffer*, sendo o *buffer* 1 para cilindro e 2 para chapa. Depois pegará a peça e fará a ação de acordo com o número do programa enviado. Esta ação pode ser colocar a peça no torno, colocar a peça na fresa, virar a peça no torno, virar a peça na fresa, colocar a peça de volta no *buffer* designado ou colocar o *pallet* com a peça na esteira num vagão parado. Sempre quando o robô está executando um programa ele informa seu estado ao CLP por um sinal discreto. Durante a execução do programa o robô se move para as posições necessárias, agarra ou solta as peças e sempre retorna para a sua posição inicial.

O torno e a fresa também executam programas e fecham suas portas de acordo com as mensagens do CLP principal. Os estados do torno e da fresa (executando ou não programa) também são enviados ao CLP. O cilindro bruto, após ter os seus dois lados usinados, se torna cilindro usinado do tipo 1 ou tipo 2 e a chapa bruta após usinagem se torna chapa usinada tipo 1 ou tipo 2. O tipo é determinado de acordo com a usinagem efetuada.

O CLP pode ainda enviar mensagem de liberação do vagão (vazio ou ocupado), provocando a desativação do *stopper*.

4. METODOLOGIA

A proposta do trabalho é realizar a simulação da planta descrita de forma que ela tenha um comportamento real, evidenciando os problemas do controle a serem realizados, como o seqüenciamento de tarefas. Foi definido que a simulação precisa evidenciar apenas os eventos importantes para o controle da planta. Isso significa que processos e eventos que não são importantes para verificar o controle que está sendo feito não necessitam ser simulados. Um exemplo seria mostrar o processo de inspeção de qualidade com detalhes enquanto o importante é apenas indicar que a inspeção está ocorrendo.

Primeiramente é necessário definir a arquitetura de controle do FMS em questão. De acordo com as descrições anteriores, observa-se que se tem uma Arquitetura de Controle Hierárquica, na medida em que existe um CLP principal

que envia os comandos de controle para as estações que possuem os seus próprios controladores. Na Fig. (9a) está representada a arquitetura da planta real.

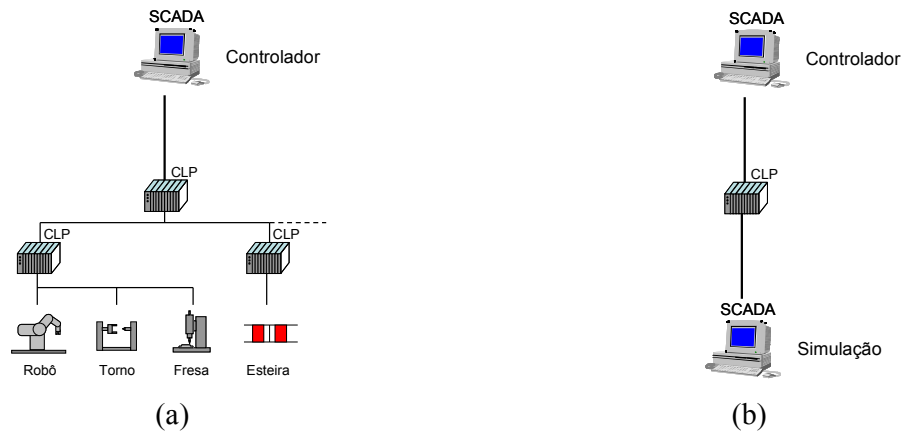


Figura 9. Arquitetura da planta real (esquerda) e da planta virtual (direita)

Na simulação, toda a planta física, bem como os controladores locais de cada estação serão simulados pelo computador (Fig. (9b)). Para o CLP principal, não importa se o sistema simulado é real ou não, já que a simulação responderá aos mesmos comandos e enviará as mesmas respostas que os controladores locais e os sensores e atuadores da planta real.

4.1. Mapa de Memória do CLP

O primeiro passo para a concepção da simulação é elaborar o mapa de memória do CLP a ser utilizado no controle da FMS virtual. Em vez de ler e acionar as saídas e entradas do CLP como o sistema real, a simulação irá fazê-lo através de uma área de memória. Dessa forma, não é necessário um CLP com um número grande de pontos de entradas e saídas, reduzindo o custo do sistema.

Conhecendo todas as entradas e saídas de cada subsistema, basta dedicar bits para cada dado de saída e entrada do CLP relativo à planta. Por exemplo, deve ser endereçado o bit de cada sensor de presença de vagão de cada estação da esteira. Um exemplo de dado de saída do CLP para a planta seriam os bits destinados a comandarem os programas dos robôs. Estes programas são codificados por um grupo de bits, devendo este grupo ser endereçado no CLP. Todos os dados de entrada e saída são digitais.

4.2. Modelagem das Estações da FMS

Outro passo importante na concepção de uma simulação é modelar a planta a ser simulada de forma a facilitar a programação da simulação, além de proporcionar ao programa resultante um código mais limpo e conciso.

O SFC foi escolhido como ferramenta para esta modelagem, por ser um diagrama de fácil entendimento e bastante conhecido no meio da automação. Além disso, a FMS descrita é completamente orientada a eventos, ou seja, uma condição tem de ser satisfeita para que os estados e ações dos componentes da planta evoluam, independente destas condições virem de comandos do CLP ou serem condições internas da planta ou ainda serem a mescla destes dois. Um exemplo de condição interna seria quando o torno, na estação FMS-1, acaba de usinar uma peça (evento) a peça se torna usinada (ação), já que tanto o evento quanto a ação não são controlados pelo CLP, e sim são resultantes de um processo que o CLP dispara, que no caso é a usinagem da peça.

Foi utilizado o editor de SFC do software ISaGRAF da SIXNET para a montagem dos diagramas SFC. O editor é simples e intuitivo de ser utilizado e facilita o trabalho de desenhar estes diagramas, além deste software atender aos padrões definidos pela norma IEC 61131-3.

A Fig. (10) representa o SFC do robô na FMS-1. Como pode ser observado, os estados 1 a 15 representam a execução dos programas, incluindo o programa de espera. As transições 1 a 14 são responsáveis por identificar que o robô recebeu um programa codificado em 4 bits, entrando assim em um estado de execução de programa. As transições 15 a 28 indicam que o programa foi finalizado, ativando o estado 16 que é de finalização e disparando consequentemente a transição 29, retornando assim para o estado de espera.

Robo

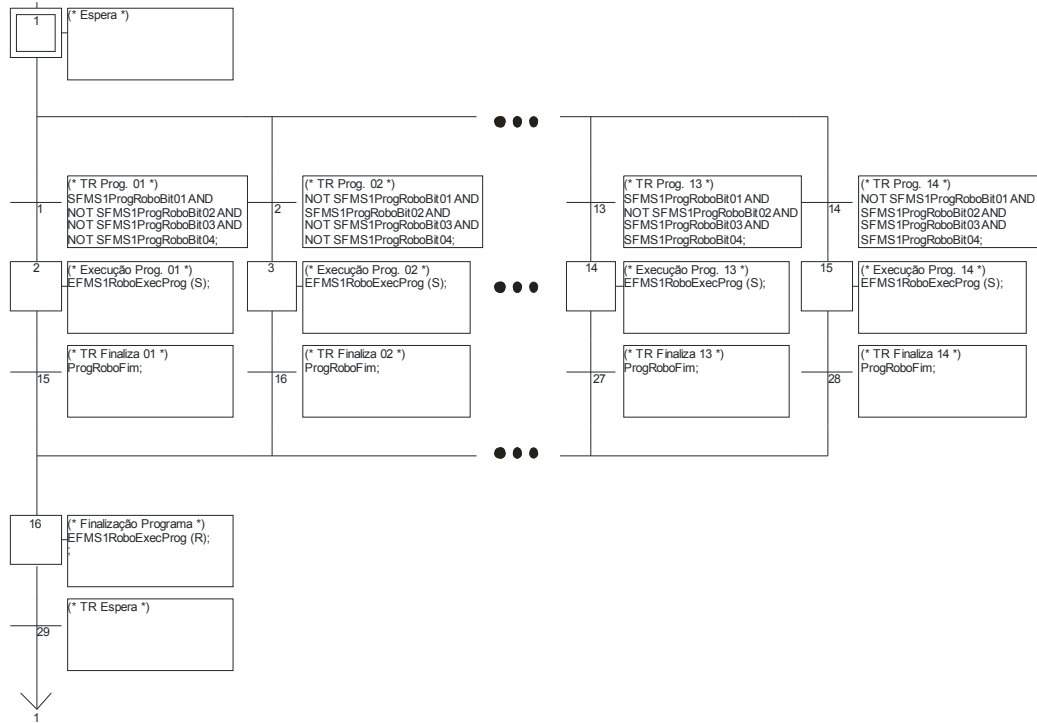


Figura 10. SFC do robô da FMS-1

5. IMPLEMENTAÇÃO

A implementação é dividida em duas partes. A primeira diz respeito à configuração de comunicação entre o SCADA onde a simulação será implementada e o CLP e a segunda diz respeito à programação da simulação em si.

5.1. Configuração da Comunicação entre o Elipse E3 e o CLP

O Elipse E3, por ser um *software* SCADA, portanto feito para interagir com CLP's e outros dispositivos industriais, possui *drivers* para configuração da comunicação com um CLP.

O *driver* utilizado é o Siemens M-Prot e dá suporte para os seguintes protocolos: PPI, MPI, ISOTCP (Ethernet) e MPI encapsulado em Ethernet. O protocolo de comunicação escolhido para ser utilizado é o ISOTCP. Esta escolha considerou apenas a facilidade que se tem de utilizar este protocolo, já que em geral computadores possuem placa de rede Ethernet e o meio físico utilizado é o cabo Ethernet, bastante conhecido e comum.

Após configurado o *driver*, foram criadas *tags* de comunicação. Estas *tags* são responsáveis por receber e enviar dados ao CLP.

5.2. Programação da Simulação

Com exceção da esteira, a implementação da simulação de todos os subsistemas da planta baseou-se nos modelos em SFC. No caso da esteira, o número de estados possíveis levaria a um SFC muito extenso e tornaria sua implementação dispendiosa.

A esteira foi representada por um *buffer* circular no qual os valores 0 representam posições vazias e os número de 1 a 32 representam os vagões. Outro vetor foi criado para armazenar os conteúdos dos vagões, ou seja, tem-se um vetor com 32 posições. Foram criadas duas *tags* básicas tipo vetor. A primeira representa o *buffer* circular da esteira e a segunda representa o vetor de vagões com os seus conteúdos. Na primeira *tag* foi inserido um evento que a cada período de tempo definido dispara um *script* que é responsável por girar a esteira. Neste mesmo *script*, as condições para que os vagões se acumulem na frente das estações quando seus *stoppers* estiverem acionados estão implementadas, bastando considerar em qual posição do *buffer* circular está cada *stopper*. Estas posições são definidas por *tags* do tipo inteiro e servem para armazenar o número da posição na esteira em que o *stopper* se encontra. O vetor de conteúdo dos vagões inicia-se todo em zero, já que os vagões iniciam-se vazios. À medida que os robôs e manipuladores das estações vão colocando ou modificando os conteúdos dos vagões, estes valores vão sendo alterados através do vetor que representa o conteúdo dos vagões.

Para cada estação foram criadas *tags* que representam as saídas ou entradas digitais do CLP. Estas *tags* foram então associadas a *tags* de comunicação com o CLP. Dessa forma, quando um bit da *tag* de comunicação muda de valor, a *tag* criada correspondente também tem seu valor alterado. A essas *tags* foram também associados nomes intuitivos, como a *tag* de saída do CLP nomeada como “LiberaVagao”, que aciona ou não o *stopper*.

Todas as estações foram programadas utilizando *scripts*, sempre seguindo os diagramas SFC elaborados. Cada diagrama SFC se tornou uma *tag* interna da estação, que representa o número do estado ativo.

Todas as transições dos diagramas SFC foram implementadas seguindo uma mesma estrutura nas *tags* que representam os SFC's. As *tags* possuem eventos que podem ser disparados de acordo com condições booleanas, relacionadas às condições de disparo das transições. Quando estes eventos são disparados, são executados *scripts* que realizam as ações descritas nos SFC's, assim como outras ações relacionadas à animação.

Para cada estação foi feita uma tela de animação gráfica. Uma quantidade grande de desenhos foi criada para representar todos os elementos importantes da planta virtual do ponto de vista do controle. Um exemplo é o robô que possui três posições possíveis, com a garra em cima (Fig. (11a)), com a garra em baixo à frente (Fig. (11b)) e com a garra em baixo atrás (Fig. (11c)). Todos os desenhos inseridos nas telas são tratados como objetos pelo Elipse E3, permitindo que *scripts* sejam executados nestes objetos a partir dos eventos gerados. Estes eventos foram criados a partir da mudança dos valores das *tags* que representam os SFC's, executando os *scripts* que são responsáveis pelas animações. Um exemplo seria a criação de uma figura inicialmente vazia (figura em branco e invisível, indicando que inicialmente a peça não está circulando) que representa o cilindro na estação FMS-1. Esta figura executa um *script* toda vez que o valor da *tag* “CilTipo”, que representa o SFC em que os estados indicam o tipo de cilindro em circulação, tem o seu valor alterado. Este *script* é responsável por carregar uma figura que representa o tipo de cilindro circulando na estação e por tornar a figura visível (cilindro circulando) ou invisível (cilindro ausente). Esta mesma figura também executa um *script* toda vez que a *tag* “LocCil”, que representa o SFC em que os estados indicam a localização do cilindro, tem o seu valor alterado (evento), sendo neste caso alterada a posição (x,y) da figura na tela. Se o valor “LocCil” mudar para um valor que indica que a peça está no *buffer*, os atributos x e y dessa figura serão alterados pelo *script* de forma que a figura fique visualmente sobre o *buffer*. Assim, o comportamento da figura que representa o cilindro na FMS-1 é coordenado por duas *tags* originadas de dois SFC's.

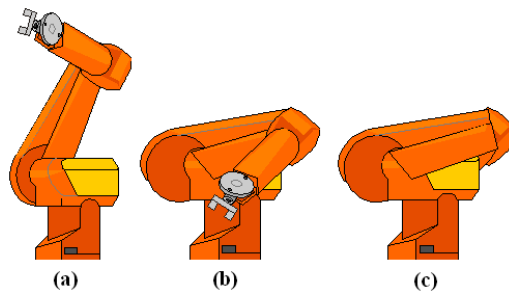


Figura 11. Figuras que representam o robô e suas possíveis posições

As animações foram feitas de forma a responderem às mudanças de valores das *tags* e fornecerem um *feedback* visual importante da planta. Este *feedback* deve ser capaz de proporcionar a identificação do seqüenciamento de rotinas de controle da planta feitas pelo controlador, da mesma forma que seria numa planta real.

6. RESULTADOS

Como resultado será mostrado uma seqüência de telas da simulação desenvolvida para a estação FMS-1. Foram inseridos nas telas *checkboxes* e botões capazes de atribuir valores às *tags* de saída do CLP como um modo manual, para que testes da simulação sem o CLP pudessem ser feitos. Alguns botões abrem telas modais que contém outros botões (ex. botões de execução de programas específicos do robô) e/ou *checkboxes*.

A Fig. (12) mostra uma seqüência de imagens da tela de simulação da FMS-1 durante a execução do programa 1 do robô (pega cilindro no vagão e coloca no *buffer* 1). O primeiro quadro mostra o robô na posição de espera com o *stopper* já acionado pelo CLP. O próximo quadro mostra o robô abaixado, no momento em que a garra é fechada. Também é possível observar o acúmulo de vagões, sendo dois deles vazios e um com chapa bruta. O terceiro quadro mostra o robô carregando o cilindro bruto juntamente com o *pallet*, pronto para ser colocado no *buffer* 1. O cilindro e o seu *pallet* estão posicionados em cima do *buffer* no quarto quadro. No quinto e último quadro, o robô está na posição inicial e em espera (fim do programa), a peça está no *buffer* e o *stopper* liberou os vagões. Em todos os quadros, tanto o torno quanto a fresa estão com suas portas abertas.

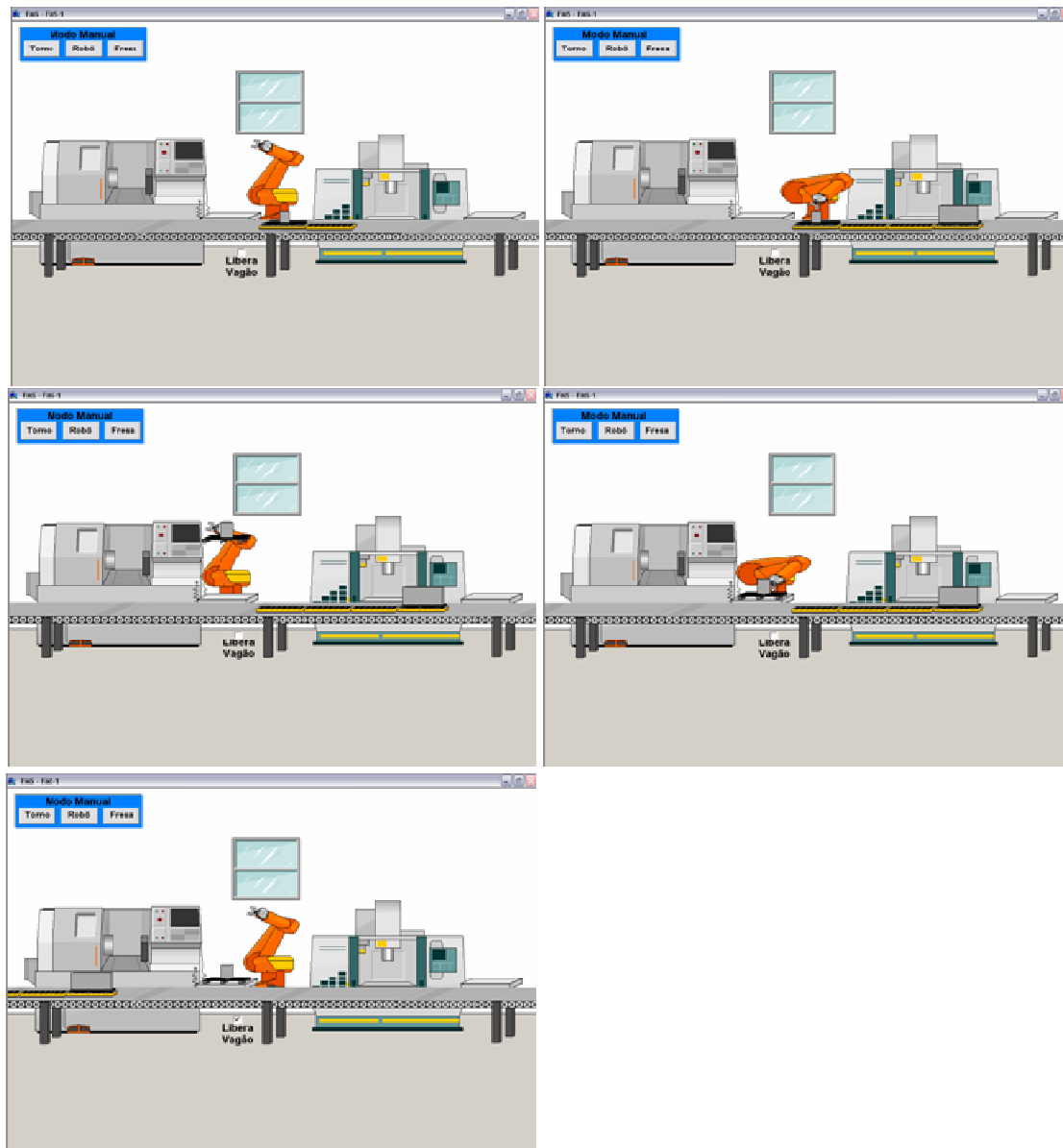


Figura 12. Seqüência de execução do programa 1 do robô na FMS-1

7. CONCLUSÕES

Uma planta FMS é naturalmente dividida em diversas estações (ou células) e cada estação possui diversos componentes, tais como robôs e máquinas CNC, além de um sistema de transporte para movimentar materiais entre as estações. A FMS em questão possui 5 estações e cada uma teve os seus componentes modelados através de SFC's. Somente a esteira (que não é considerada uma célula) foi modelada utilizando-se um *buffer* circular.

A implementação, após a etapa de modelagem tornou-se uma atividade muito eficiente já que toda idéia de como seria feita a programação estava representada nos modelos. A comunicação via Ethernet com o CLP foi configurada e a criação da simulação foi realizada baseando-se no modelo. A programação também envolveu a criação de animações que representam os eventos físicos que ocorrem na planta.

Durante a implementação foram feitos testes que comprovaram o satisfatório funcionamento da simulação, respondendo perfeitamente à geração manual dos valores de saída do CLP, bem como à geração pelo próprio programa do CLP através da comunicação Ethernet.

Esta planta foi criada para ser utilizada como ferramenta de estudo de métodos de controle de plantas FMS, que muitas vezes não podem ser feitos devido aos altos custos na montagem de uma planta real. Tendo este objetivo, será necessário integrar esta planta com um CLP e um supervisorío juntamente com um algoritmo de controle.

8. AGRADECIMENTOS

Os autores agradecem à FAPEMIG pelo apoio financeiro ao desenvolvimento do trabalho e ao SENAI-CIMATEC por disponibilizar as informações sobre o sistema estudado.

9. REFERÊNCIAS

- Galvez, J. M, Santos, A. C., Miranda, N., 2006. “Um Laboratório Virtual para Aplicações em Ensino de Automação e Controle”. XVI CBA – Congresso Brasileiro de Automática, Salvador-BA.
- Matsusaki, C. T. M., 2004. “Modelagem de Sistemas de Controle Distribuídos e Colaborativos de Sistemas Produtivos”. São Paulo, 154p. Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.
- Teixeira, E. L. S., 2006. “Desenvolvimento de uma Unidade de Gerenciamento de uma Célula Flexível de Manufatura Integrada a um Sistema CAD/CAPP/CAM”. Brasília-DF, 178p. Dissertação (Mestrado em Sistemas Mecatrônicos) – Departamento de Engenharia Mecânica e Mecatrônica, Universidade de Brasília.

10. DIREITOS AUTORAIS

Os autores são os únicos responsáveis pelo conteúdo do material impresso incluído neste trabalho.

DEVELOPMENT OF A FLEXIBLE MANUFACTURING SYSTEM SIMULATION USING SCADA SOFTWARE

Horácio Medina da Rocha, horaciomedina@gmail.com¹

Lucas Nasciutti Prudente, lukaznp@yahoo.com.br¹

Eduardo José Lima II, eduardo@demec.ufmg.br²

¹Engenharia de Controle e Automação, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627 - Pampulha, CEP 31270-901, Belo Horizonte, Minas Gerais, Brasil

²Departamento de Engenharia Mecânica, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627 - Pampulha, CEP 31270-901, Belo Horizonte, Minas Gerais, Brasil

Abstract: *Real Flexible Manufacturing Systems (FMS) are extremely expensive, structurally complex and demand a large space to be build. A didactic plant may cost hundreds of thousands of dollars, what allied with the lack of space and complexity can make academic researches and practical classes impossible to be done. This work's objective is to simulate a FMS using a SCADA software such that the virtual plant is capable to represent all the actions of a real plant, obtaining a powerful tool for researches on automation of these plants. The SCADA software was chosen to simplify the integration of the virtual plant and a PLC (Programmable Logic Controller), which is responsible for the commands sent to the plant and for the signals received from it. In the project of the simulation, SFC (Sequential Function Chart) diagrams were used considering that the actions of the FMS components can be represented by well defined states and actions as all the output and input signals are discrete. Another factor that makes the use of SFC an ideal choice is that the SCADA software used (Elipse E3) allows the representation of the state transitions through boolean conditions and each component of the plant can be represented by independent objects, using the functionalities of the object-oriented programming. The developed simulation corresponded to the real plant, allowing tests of CLP programs and the verification of control techniques.*

Keywords: *Flexible Manufacturing System, Simulation, SCADA*