

IMPLEMENTAÇÃO DE CONTROLE SUPERVISÓRIO EM LINGUAGEM LADDER PARA UMA CÉLULA FLEXÍVEL DE MANUFATURA DIDÁTICA

JEFERSON L. CURZEL⁽¹⁾, ANDRÉ B. LEAL⁽²⁾

⁽¹⁾ *Sociedade Educacional de Santa Catarina - Instituto Superior Tupy
Rua Albano Schmidt, 3333, Joinville, 89227-700, SC*

⁽²⁾ *Departamento de Engenharia Elétrica, Universidade do Estado de Santa Catarina
Campus Universitário Prof. Avelino Marcante, Joinville, 89223-100, SC
E-mails: jeferson.curzel@sociesc.com.br, leal@joinville.udesc.br*

Abstract— In this paper we perform the integration of the equipments of a didactic flexible manufacturing cell to be used in the research of supervisory control techniques. The theory of supervisory control is used to find a minimally restrictive supervisor, who satisfies the imposed specifications of control to the system. The implementation of the supervisor obtained is made in a programmable logical controller (CLP), through language LADDER. With this work, we intend to contribute towards the dissemination of the supervisory control theory in the industries.

Keywords— Flexible Manufacturing Cell, Supervisory Control, Discrete Event Systems.

Resumo— Neste trabalho faz-se a integração dos equipamentos de uma célula flexível de manufatura didática a ser utilizada para pesquisa de técnicas de controle supervísório. É utilizada a teoria de controle supervísório para encontrar um supervisor minimamente restritivo, que satisfaça as especificações de controle impostas ao sistema. A implementação do supervisor obtido é feita em controlador lógico programável (CLP), através da linguagem LADDER. Pretende-se, desta forma, contribuir para disseminar o uso da teoria de controle supervísório junto às indústrias.

Palavras-chave— Célula Flexível de Manufatura, Controle Supervísório, Sistemas a Eventos Discretos.

1 Introdução

A Teoria de Controle Supervísório (Ramadge e Wonham, 1989) possibilita que problemas de controle sejam solucionados de forma sistemática, garantindo assim que o supervisor obtido atenderá as especificações impostas pelo projetista. Além disto, o seu uso facilita alterações no programa do controlador lógico programável (CLP), alterações estas que são necessárias sempre que é feita alguma modificação (inclusão/exclusão de equipamentos ou alteração no *layout*) na célula flexível de manufatura. Entretanto, a teoria supracitada não é conhecida e dominada no âmbito industrial, de forma que, em geral, a resolução de problemas de controle supervísório nas indústrias é feita sem a utilização de um procedimento formal.

Neste artigo utilizamos uma célula flexível de manufatura didática e aplicamos a Teoria de Controle Supervísório de forma a encontrar um supervisor minimamente restritivo, que satisfaça as especificações de controle impostas ao sistema. O supervisor obtido é implementado em CLP utilizando-se a linguagem LADDER, por ser esta uma linguagem amplamente difundida, de fácil visualização e entendimento, possibilitando que alterações e localização de falhas sejam feitas rapidamente. Na implementação do supervisor monolítico, propõe-se uma metodologia para reduzir o tamanho do programa implementado em CLP, através da eliminação de linhas de programa.

Na seqüência do artigo, na Seção 2 é apresentada a célula de manufatura e é feita uma breve descrição do seu funcionamento. Na Seção 3 são apresentados os autômatos que modelam os subsistemas que compõem a célula. Na Seção 4 é feito o projeto do supervisor, onde são modeladas as especificações de controle, e é apresentada a síntese do supervisor minimamente restritivo. Na Seção 5 é apresentado o programa implementado em LADDER no CLP e na Seção 6 têm-se as conclusões.

2 A Célula Flexível de Manufatura

A célula de manufatura considerada (Figura 1) é composta por dois robôs, uma mesa giratória, uma estação de teste, uma esteira transportadora e uma mesa para peças rejeitadas.

O funcionamento da célula prevê que as peças que estão na esteira são transportadas pelo robô 1 até a mesa giratória, são manufaturadas e testadas, e se estiverem com problema, são transportadas para a mesa de rejeito pelo robô 2, caso contrário, são retiradas da célula pelo robô 2. Como elemento para teste da peça, utiliza-se um sensor fotoelétrico, que detecta diferença de coloração da peça (branca ou preta).

A Figura 2 mostra o CLP S7 300, utilizado para controlar a célula, e a interface para sinais de entrada e saída.

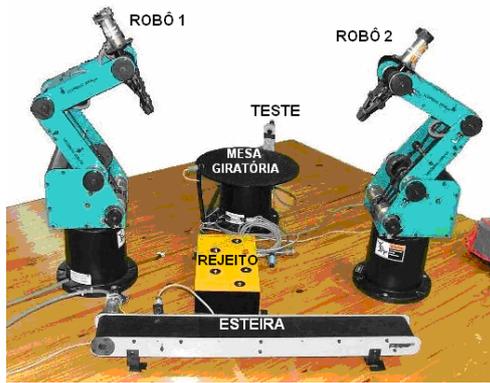


Figura 1. A Célula Flexível de Manufatura Didática.

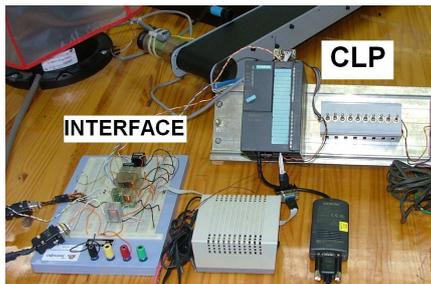


Figura 2. O CLP utilizado e a interface de sinais.

Os sinais dos sensores e atuadores dos equipamentos da célula são interligados, respectivamente, ao módulo de entradas e saídas digitais do CLP, por meio do qual é feito o controle da seqüência de operação de cada subsistema físico. As rotinas de execução das tarefas dos robôs são executadas em controladores individuais, dedicados para cada robô, que estão interligados ao CLP através da interface de sinais de entradas e saídas.

3 Modelagem da Célula

A modelagem da célula e a resolução de problemas de controle são feitos seguindo a Teoria de Controle Supervisório de Sistemas a Eventos Discretos (Ramadge e Wonham, 1989). Nesta abordagem o controle é feito por um autômato denominado de supervisor, o qual restringe o comportamento do sistema físico de forma a satisfazer um conjunto de especificações. O alfabeto de eventos que compõe os modelos individuais de cada subsistema físico é dividido em dois tipos de eventos: os eventos controláveis, cuja ocorrência pode ser desabilitada pela ação de controle, como por exemplo, o início de uma atividade ou a parada de uma esteira, e os eventos não-controláveis, cuja ocorrência não pode ser desabilitada pela ação de controle, como por exemplo, a ativação de um sensor (Cury, 2001).

Como simplificação para a resolução do problema proposto neste trabalho, adota-se apenas duas posições para a mesa giratória, onde na primeira posição tem-se a entrada da peça, e na segunda posição tem-se o teste e a retirada da peça da mesa giratória, conforme a Figura 3.

A tabela 1 indica o evento associado a cada subsistema físico da célula e a descrição do evento.

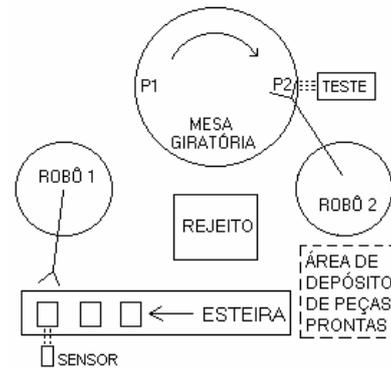


Figura 3. Planta baixa da célula.

Tabela 1. Eventos dos subsistemas físicos da célula.

SUBSISTEMA	EVENTO	DESCRIÇÃO
Esteira	E_liga	Liga a esteira.
	E_desl	Desliga a esteira.
Sensor da esteira	S_liga	Chegada de peça na esteira.
	S_desl	Saída de peça da esteira.
Mesa giratória	I_giro	Liga a mesa giratória.
	F_giro	Fim de giro da mesa.
Robô 1	T_M	Inicia a operação do robô 1.
	F_rb1	Fim de operação do robô 1.
Estação de teste	I_teste	Início do teste de peça.
	T_OK	Resultado do teste => peça boa.
	T_NOK	Resultado do teste => peça ruim.
Robô 2	T_R	Início de operação do robô 2 para retirada de peça ruim.
	T_S	Início de operação do robô 2 para retirada de peça boa.
	F_rb2	Fim de operação do robô 2.

3.1 Modelos dos subsistemas físicos da planta

O funcionamento dos subsistemas físicos pode ser modelado pelos autômatos G1 a G6, mostrados na Figura 4. Nos modelos dos autômatos apresentados, os círculos representam os estados dos subsistemas e os círculos duplos representam estados onde o subsistema completa uma tarefa. Os arcos entre os estados indicam as transições ou eventos. Arcos seccionados por uma pequena linha transversal indicam que o evento é controlável (Cury, 2001). Os eventos envolvidos nesta modelagem são descritos na Tabela 1.

Para o autômato G1, que modela o funcionamento da esteira, o estado inicial "0" representa a esteira desligada e o estado "1" representa a esteira ligada. Nota-se aqui que os eventos E_liga e E_desl são eventos controláveis e que, portanto, podem ser desabilitados pelo supervisor.

Os autômatos G2 a G6 representam o comportamento do sensor da esteira, do robô 1, da mesa giratória, da estação de teste e do robô 2, respectivamente.

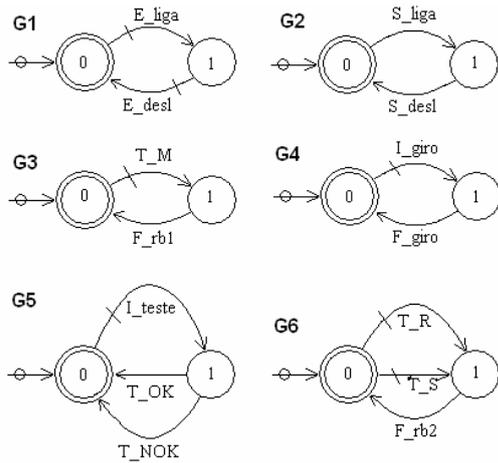


Figura 4. Autômatos dos subsistemas G1 a G6.

3.2 Modelo das restrições físicas da planta

Conforme proposto por Moraes e Leal (2006), foram modeladas algumas restrições físicas, devido aos modelos individuais não possuírem informações importantes do sistema global. Por exemplo, se a esteira não estiver acionada, não ocorrerá o acionamento do sensor posicionado ao final desta, não havendo dessa forma a geração do evento correspondente. Na Figura 5 são mostradas as restrições físicas R1 a R4.

A restrição física R1 indica que o sensor da esteira só pode ser ativado quando a esteira está ligada. A restrição física R2 informa que uma vez ativado o sensor da esteira, é possível (e desejado) que a esteira seja desligada antes do sensor ser desativado.

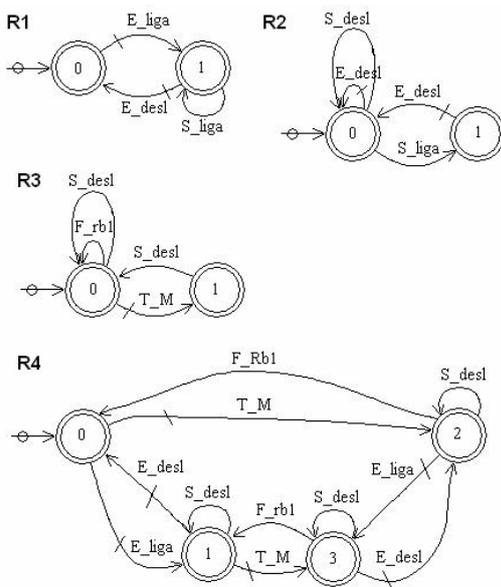


Figura 5. Autômatos para as restrições R1 a R4.

A restrição R3 considera o fato de que a desativação do sensor da esteira depende da ação do robô 1. Desta forma, a desativação de tal sensor se dará após o início e antes do final de operação do robô 1 (na reti-

rada de peça pelo robô 1). A restrição física R4 informa que o sensor da esteira não pode ser desativado se a esteira estiver desligada e o robô 1 estiver parado.

4 Projeto do Supervisor

A Teoria proposta por Ramadge e Wonham (1989) prevê que a partir de um conjunto de especificações de controle possamos obter um supervisor que garanta que o sistema se comporte conforme desejado. A seguir são apresentadas as especificações de controle.

4.1 Modelo das especificações

Na Figura 6 são mostradas as especificações de controle E1 a E9. A especificação E1 indica que a esteira pode ser ligada somente quando o sensor está desativado e deve ser desligada apenas quando o sensor está ativado. A especificação E2 indica que o robô 2 executa o transporte da peça para o rejeito ou para a saída em função do resultado da estação de teste. A especificação E3 impõe que a mesa giratória inicie seu giro somente após o fim de operação do robô 1. A especificação E4 indica que o robô 2 deve iniciar sua operação somente após o fim de giro da mesa. Além disso, a mesa só pode girar novamente após o fim de operação do robô 2. A especificação E5 indica o robô 1 só pode finalizar sua operação com a mesa em repouso.

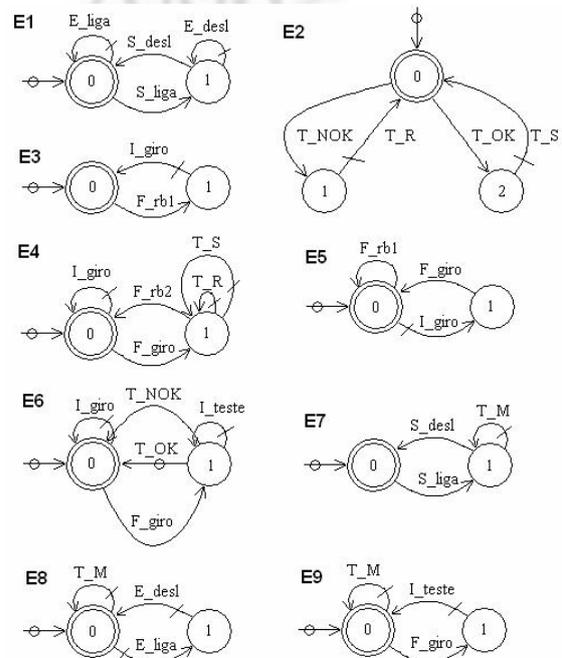


Figura 6. Autômatos para as especificações E1 a E9.

A especificação E6 indica que a estação de teste deve iniciar sua operação somente após o fim de giro da mesa. Além disso, um novo giro na mesa não deve

ser iniciado enquanto o teste não for finalizado. A especificação E7 indica que a operação do robô 1 deve ser iniciada somente se houver peça ativando o sensor. A especificação E8 indica que o robô 1 só deve iniciar sua operação se a esteira estiver desligada. A especificação E9 prioriza o início de teste em detrimento ao início de transporte do robô 1.

4.2 Síntese do supervisor

Neste trabalho utiliza-se um único supervisor para controlar a célula de manufatura. Assim, no processo de síntese do supervisor deve-se obter um modelo para a planta (sistema a ser controlado) e um modelo global que reúna todas as especificações de controle. O modelo para a planta é obtido através da composição síncrona (Ramadge e Wonham, 1989) entre os autômatos dos subsistemas físicos e as restrições físicas do sistema, ou seja, $G = G_1 \parallel G_2 \parallel \dots \parallel G_6 \parallel R_1 \parallel \dots \parallel R_4$. Esta composição síncrona resulta em um autômato com 128 estados e 728 transições. A especificação global E é obtida a partir das especificações individuais E_i , isto é, $E = E_1 \parallel E_2 \parallel \dots \parallel E_9$. A linguagem alvo K é obtida a partir da sincronização de G e E , fazendo-se $K = G \parallel E$. Por fim, calcula-se a máxima linguagem controlável contida em K , denotada $\text{SupC}(G, K)$. O autômato que reconhece $\text{SupC}(G, K)$ possui 73 estados e 158 transições e consiste no supervisor não bloqueante e minimamente restritivo, que será implementado no CLP de forma a garantir o cumprimento das especificações de controle. Todo o processo de síntese do supervisor, descrito acima, foi realizado com o auxílio da ferramenta Grail (Raymond e Derick, 1995).

5 Programação do CLP

O CLP utilizado para a programação foi o Siemens da família Step7_300 com a CPU 321IFM. Esta CPU possui 10 entradas e 06 saídas digitais, as quais são interligadas à planta para a monitoração dos estados dos sensores e acionamento dos subsistemas físicos. A linguagem utilizada na programação do CLP foi a Diagrama de Contatos (LADDER), por ser a mais conhecida no meio industrial e de fácil edição e entendimento.

Na programação do CLP utilizou-se uma metodologia diferente das empregadas por Fabian e Hellgren (1998) e Queiroz *et al* (2001). Os primeiros utilizam a implementação com base na detecção de flancos positivos para os eventos não-controláveis. No trabalho de Queiroz *et al* (2001), a síntese de supervisores é feita utilizando-se a abordagem modular local e na implementação dos supervisores modulares é feita a habilitação ou a desabilitação de um subconjunto de eventos controláveis que faz com que o comportamento da planta evolua de forma a respeitar cada restrição dos supervisores modulares.

Neste trabalho, quando da ocorrência de eventos não-controláveis, variáveis internas do CLP (*flags*) armazenam as informações, até que elas sejam processadas. Para a implementação, foram criados quatro blocos de programa de forma a facilitar a organização:

- bloco de gerenciamento (OB1)
- bloco do supervisor (FC1)
- bloco dos eventos (FC2)
- bloco das saídas (FC3)

5.1 Programação do bloco OB1

O bloco OB1 é necessário no CLP utilizado pois em sua rotina de execução é definida a seqüência em que os blocos de função (FC) serão executados, conforme a figura 7.

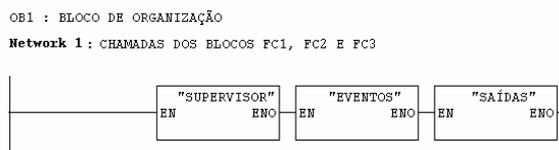


Figura 7. Programação do bloco OB1 em LADDER.

5.2 Programação do bloco FC1 - Supervisor

Na programação do bloco FC1 foi implementado o supervisor monolítico. Dessa forma, cada estado do supervisor corresponde a uma memória interna do CLP, que é ativada ou desativada de acordo com a seqüência do supervisor. O supervisor é mostrado em parte na Figura 8, onde cada estado foi nomeado no bloco de programação com o prefixo “S”, assim, o estado “0” do supervisor passou a ser simbolizado por “S0” na programação do CLP.

Conforme ilustrado na Figura 9, a partir do comando “INÍCIO”, o supervisor ativa o estado “S0” (*Network 1*). Neste estado, o supervisor fica aguardando a ocorrência do evento E_{liga} (*Network 2*), que por ser um evento controlável, deve ser habilitado pelo próprio programa do CLP. A geração deste evento é feita no bloco FC2 (Fig. 11 – *Network 1*), mas para tal é necessário que o motor da esteira tenha sido efetivamente ligado, o que é feito no bloco FC3 (Fig. 10 – *Network 1*).

5.3 Programação do bloco FC3 - Saídas

Neste bloco são definidas as ações do supervisor, de acordo com o estado definido no bloco FC1. Assim, conforme visto, quando o supervisor estiver no estado “S0”, a saída que ativa o motor da esteira será acionada, conforme a Figura 10 (*Network 1*). Este comando envia um sinal para a saída física do CLP.

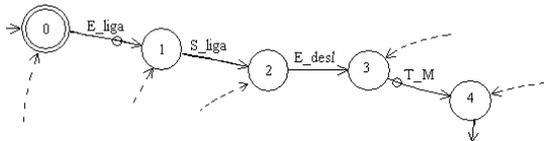


Figura 8. Parte inicial do supervisor obtido.

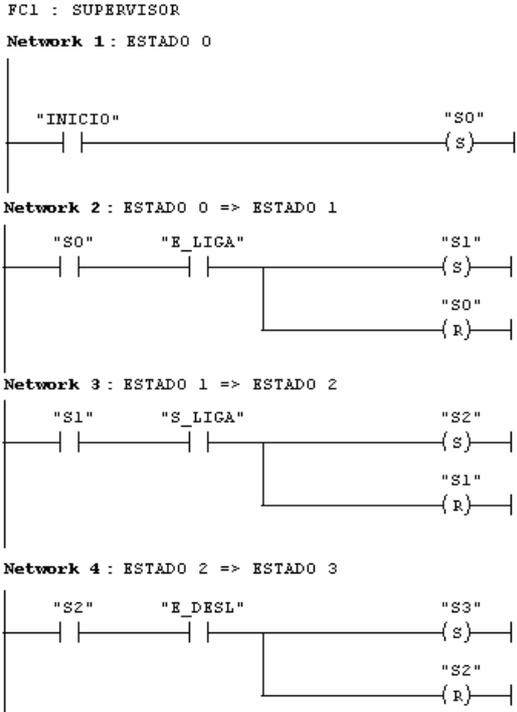


Figura 9. Programação do bloco FC1 em LADDER.

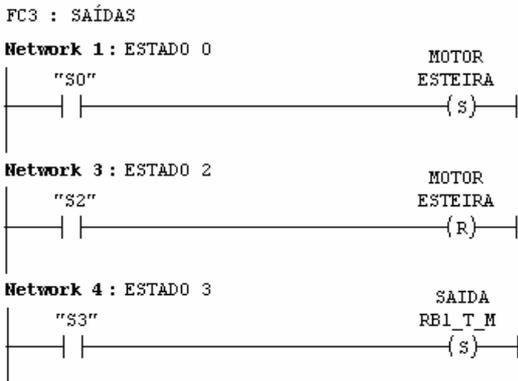


Figura 10. Programação do bloco FC3 em LADDER.

5.4 Programação do bloco FC2 - Eventos

Neste bloco são ativados ou desativados os eventos de acordo com sinais de entrada e saída do CLP. Dessa forma, ao ser ativado o motor da esteira, o evento E_liga será ativado, indicando para o supervisor (bloco FC1) que o motor da esteira foi ativado. Ao mesmo tempo, o evento que informava ao supervisor que o motor da esteira havia sido desligado é desativado, conforme a Figura 11.

Uma particularização na implementação deste problema foi adotada quando da ocorrência de dois

eventos controláveis em um estado qualquer do supervisor. Na Figura 12 representamos uma seqüência iniciada no estado 6, que transita direto para o estado 11, pois os dois eventos (E_liga e I_giro) são disparados. A programação do CLP quando da ocorrência dos dois eventos controláveis a partir do estado 6 é feita conforme vemos nas figuras 13 e 14.

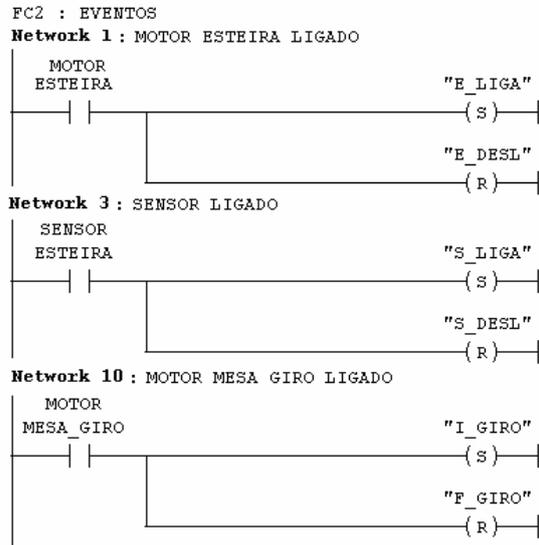


Figura 11. Programação do bloco FC2 em LADDER.

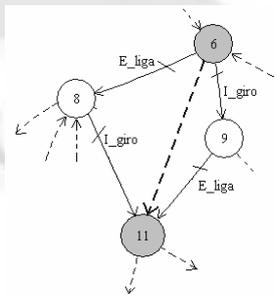


Figura 12. Disparo simultâneo de eventos controláveis.

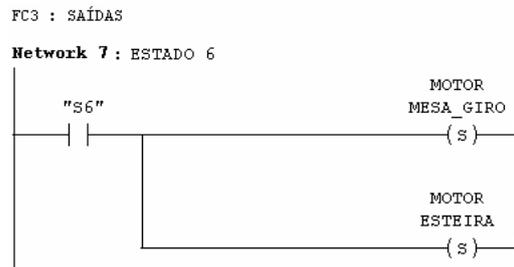


Figura 13. Disparo simultâneo de eventos no bloco FC3.

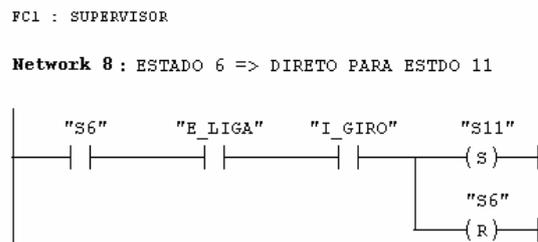


Figura 14. Disparo simultâneo de eventos no bloco FC1.

Generalizando, caso ocorram dois ou mais eventos controláveis simultâneos em um estado qualquer, o autômato transita para o estado no qual tais eventos convergem.

Nos estados do supervisor em que existem eventos controláveis e não-controláveis, dá-se prioridade no tratamento dos não-controláveis, pois estes são espontaneamente gerados em função da dinâmica da planta e devem ser tratados imediatamente após a sua ocorrência. Na programação do CLP, essa prioridade é dada de acordo com a disposição das linhas do programa, ou seja, como a execução do programa é cíclica, os eventos não-controláveis são executados nas linhas anteriores aos eventos controláveis. Por exemplo, no estado 29 (Figura 15), os eventos não-controláveis T_OK e T_NOK têm prioridade de execução.

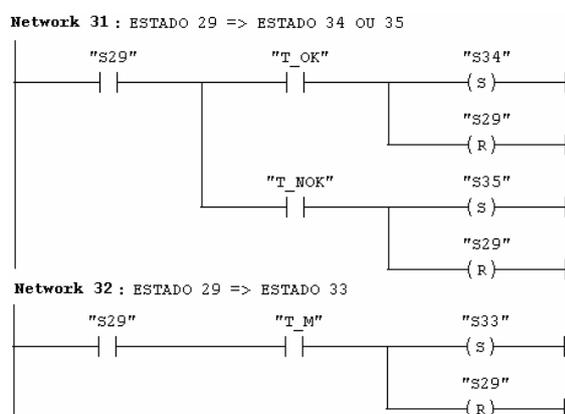


Figura 15. Programação de eventos não-controláveis (bloco FC1).

6 Conclusão

O emprego da Teoria de Controle Supervisório em células flexíveis de manufatura modeladas a partir de autômatos e sua implementação em linguagem de CLP mostrou-se bastante versátil do ponto de vista da reprogramação de rotinas de trabalho e inclusão de novos dispositivos na célula. Com os resultados obtidos neste trabalho, pretende-se disseminar o uso da teoria de controle supervisório em células flexíveis de manufatura industriais.

Referências Bibliográficas

- Cury, J. E. R. (2001). Teoria de Controle Supervisório de Sistemas a Eventos Discretos. *V Simpósio Brasileiro de Automação Inteligente (Minicurso)*.
- Fabian, M. and Hellgren, A. (1998). PLC-based implementation of supervisory control for discrete event systems. *Proceedings of the 37th Conference on Decision and Control*. Tampa, USA.
- Moraes, W.R. e Leal, A.B. (2006). Controle Supervisório do Transportador de Entrada de um

Sistema Flexível de Manufatura. *Proceedings of the VII International Conference on Industrial Applications-Induscon*. Recife, Brasil.

- Queiroz, M.H., Santos, E.A.P. e Cury, J.E.R. (2001). Síntese modular do controle supervisório em diagrama escada para uma célula de manufatura. *Anais do V Simpósio Brasileiro de Automação Inteligente*, Vol. 5, p. 1072.
- Ramadge, P. J. and Wonham, W. M. (1989). The Control of Discrete Event Systems, *Proceedings of IEEE*, Vol. 77, n. 1, pp. 81-98.
- Raymond, D. and Derick, W. (1995). Grail: A C++ library for automata and expressions, in *Journal of Symbolic Computation*, 11, pp. 341-350.