

USO DE RESTRIÇÕES SOBRE CORES DECOMPOSTAS COMO ALGORITMO DE CONTROLE NUMA CÉLULA PRODUTIVA ROBOTIZADA

DYONATHA R. COSTA¹, FRANCISCO A. A. MENEZES¹, GIOVANNI C. BARROSO²

1. *Departamento de Engenharia de Teleinformática, Universidade Federal do Ceará*
Av. Benjamin Brasil 1100, bloco 01, apto. 306, Maraponga, Fortaleza -CE
E-mails: drcosta@live.com, alencarmenezes@gmail.com

2. *Departamento de Física, Universidade Federal do Ceará*
Campus do Pici S/N, bloco 928, sala 39
E-mails: gcb@fisica.ufc.br

Abstract— This paper proposes conversion for flexible manufacturing system (FMS) in Colored Petri Nets (CPN) for Ladder Diagram (LD). For this we use the methodology of supervisory control for CPN nominated Control Restrictions on decomposed colors (RCCD). It emphasizes the transformation CPN controlled to LD and ease of inclusion of control. The results described here demonstrate that the methodology facilitating the work of a PLC programmer minimizing the possible errors during the programming phase. To exemplify the methodology we use a vertical articulated robot controlled via PLC, which operates compared with and without supervision.

Keywords— Supervisory control, Coloured Petri nets, discrete event systems.

Resumo — Este trabalho apresenta uma proposta de conversão de modelos de células de manufatura (FMS) em Redes de Petri Coloridas (RPC) para *Ladder Diagram* (LD). Para tanto utiliza-se a metodologia de controle supervísório para RPC denominada Restrições de Controle sobre Cores Decompostas (RCCD). Enfatiza-se a transformação de RPC não controlada para LD e a facilidade de inclusão do controle. Os resultados aqui descritos demonstram que a metodologia facilita o trabalho de um programador de CLP minimizando os possíveis erros durante a fase de programação. Para exemplificar a metodologia é usado um robô articulado vertical controlado via CLP, que opera comparativamente com e sem supervisão.

Palavras-chave— Controle supervísório, redes de Petri coloridas, sistema a eventos discretos.

1 Introdução

O uso de robôs como elementos-chave para maximizar o processo produtivo industrial apresenta-se como solução quando se fala em segurança, qualidade, repetibilidade, acurácia e disponibilidade. Entretanto, na perspectiva de eficiência, a ausência de técnicas de controle pode sugerir a aquisição desnecessária de tais máquinas. Em essência, deseja-se que o custo industrial de tal máquina seja absorvido pela multifuncionalidade desta sem bloqueios, conflitos de operação e com reduzida complexidade operacional.

Nesse sentido, tal objetivo poderá ser alcançado pelo uso da teoria do controle supervísório (TCS) (Ramadge & Wonham, 1987) e (Ramadge & Wonham, 1989), a qual surge da necessidade de uma ferramenta sistemática (Hopcroft, Motwani, & Ullman, 2006) para especificação e implementação do controle dos sistemas a eventos discretos (SED). Nessa teoria, um SED é modelado por um autômato cujo comportamento é descrito por linguagens formais. Sob esta perspectiva, a questão chave do controle é encontrar o supervisor, tal que, associado ao sistema, gere a linguagem especificada. Três fundamentos compõem a TCS: controlabilidade de uma linguagem, existência do supervisor e existência de máxima linguagem controlável. Entretanto, alguns inconvenientes são evidenciados ao se usar autômatos como ferramenta para modelagem de SED, tais como o crescimento exponencial do número de esta-

dos, o que limita seu uso a sistemas de menor tamanho. Diante disso, propõe-se uma abordagem da TCS utilizando como ferramenta as redes de Petri coloridas (RPC), apropriadas à modelagem de sistemas mais robustos, sem prejuízo de visualização gráfica e, com o mesmo poder de análise das Redes de Petri (RP).

(Gomaa, 2011) apresenta uma ferramenta em Java para a conversão de um modelo de SED em RP para a linguagem *Ladder Diagram* (LD). Esta automatização representa uma vantagem para a programação, embora restrita às RP lugar-transição. (Barghash, Abuzeid, Al-Rabadi, & Jaradat, 2011) estudam o problema dos SED semiautomáticos em 3 níveis de funcionalidades a partir das RP e, realizando implementações destes em CLP via LD. O trabalho não explora, entretanto, as estratégias de conversão de RP em LD, fator de elevada importância para o projetista. (Morais & Castrucci, 2007) apresentam as técnicas *top-down e bottom-up* para modelagem de um SED em RP e posterior conversão para LD apresentando uma tabela de correlação entre esses. Tais técnicas, embora mantenham uma programação bem estruturada, não permitem ao programador aplicar controle sem grandes alterações na RP e no programa.

O objetivo deste trabalho é a utilização do método de controle de SED denominado Restrições de Controle sobre Cores Decompostas (RCCD) para a implementação de um algoritmo de controle numa célula de produção robotizada. Neste faz-se uso do

método Fusão de Controladores e das RPC (Prata, Barroso, & Arruda, 2008).

A estrutura deste trabalho compreende uma pequena introdução da TCS e das RP na Seção 2; a apresentação do método RCCD e do método de fusão de controladores, na Seção 3; a metodologia proposta para transformação de um modelo RPC em LD, com e sem controle, bem como um exemplo de aplicação, são apresentados na Seção 4; as discussões sobre a operação do robô via CLP, na Seção 5; e as conclusões, na Seção 6.

2 A modelagem de sistemas

2.1 Conceitos de redes de Petri

As RP são uma ferramenta matemática e gráfica para modelagem de SED. Estas consistem de um grafo direcionado com dois tipos de nós: lugares e transições, e arcos direcionados e ponderados conectando lugares a transições e transições a lugares, bem como fichas (inteiros positivos) associadas aos lugares, conforme (Murata, 1989).

As redes de Petri coloridas (RPC) (Jensen & Kristensen, 2009) são uma extensão às RP que combinam a estrutura de uma RP com uma linguagem de programação. Assim, pode haver diferenciação de fichas e associação de variáveis e funções aos arcos da rede em substituição aos pesos dos arcos das RP. Sendo assim, as RPC são capazes de modelar sistemas mais complexos de forma mais compacta e de fácil visualização.

2.2 Controle supervisorio

O desafio de encontrar o controle supervisorio numa rede é, de um modo geral, resolvido pela TCS, que separa o sistema a ser controlado (*open loop dynamics*) do controlador (*feedback control*) e envolve a modelagem, especificação do comportamento e síntese do supervisor. No processo, a síntese de um supervisor é realizada para um dado modelo com o objetivo de satisfazer uma especificação de comportamento desejada. Dessa forma, o supervisor é um agente externo que possui habilidade de observar os eventos gerados pelo sistema e influenciar no seu comportamento através de entradas de controle. Em malha fechada, a ação de controle do supervisor garante que o funcionamento do sistema esteja de acordo com uma especificação dada.

Na TCS, segundo (Barkaoui, Chaoui, & Zouari, 1997) e (Cury, 2001), um SED é modelado por um par de linguagens, L e L_m , bem definidas, mais o conjunto de eventos Σ . L representa o conjunto de todas as sequências de eventos que o sistema pode gerar a partir de seu estado inicial e L_m é a linguagem marcada que representa as tarefas completadas do sistema. Seja:

$$G = (Q, \Sigma^*, \delta, \Sigma, q_0, Q_m) \quad (1)$$

o gerador associado a estas duas linguagens. Por definição, Σ^* representa o conjunto de todas as se-

quências finitas em Σ , incluindo a sequência nula ϵ ; Q é o conjunto de estados de G ; $\delta: \Sigma \times Q \rightarrow Q$ é a função de transição de estados, tal que, $\delta(\epsilon, q) = q$ e $\delta(\sigma, q) = q'$ onde $q, q' \in Q$ e $\sigma \in \Sigma$; $q_0 \in Q$ é o estado inicial de G e $Q_m \subseteq Q$ é o conjunto de estados marcados. $\Sigma: Q \rightarrow 2^\Sigma$ denota o conjunto ativo de eventos num estado, tal que, $\delta(q, s)$ é definida.

Segue que:

$$L = \mathcal{L}(G) = \{s \in \Sigma^*: \delta(q_0, s) \text{ é definida}\}, \quad (2)$$

$$L_m = \mathcal{L}_m(G) = \{s \in \mathcal{L}(G): \delta(q_0, s) \in Q_m\}, \quad (3)$$

G é dito não-bloqueante (trim) quando

$$\mathcal{L}(G) = \overline{\mathcal{L}_m(G)}, \quad (4)$$

caso contrário é dito bloqueante.

Quando o autômato G tem um “comportamento proibido” (gera tarefa não especificada), o objetivo do controle é então restringir o comportamento do sistema não controlado, descrito por G , dentro dos limites do “comportamento não-proibido”, dado como um subconjunto de $\mathcal{L}(G)$; esse controle pode ser alcançado conectando o sistema em malha com um supervisor S (um controlador externo), sob a restrição que S nunca desabilite um evento não controlável. O sistema resultante em malha fechada denotado por S/G , a linguagem gerada $\mathcal{L}(S/G) \subseteq \mathcal{L}(G)$ e sua linguagem marcada consiste exatamente das sequências marcadas de G que estão sob o controle de S . O “comportamento não-proibido” é dado como um subconjunto de $\mathcal{L}_m(G)$. Assim, uma propriedade importante que S deve satisfazer é ser um supervisor próprio, dessa forma, o sistema sob supervisão S/G é não-bloqueante, ou seja,

$$\mathcal{L}(S/G) = \overline{\mathcal{L}_m(S/G)}, \quad (5)$$

3 O método de Restrições de Controle Sobre Cores Decompostas (RCCD)

3.1 RCCD

Aproveitando-se das características das RPC, o método restrições de controle sobre cores decompostas (RCCD), para a síntese de supervisores, desenvolve-se a partir da análise da matriz de incidência da RPC, com a decomposição dessa matriz em outras matrizes de coeficientes de uma mesma variável (Menezes, Barroso, & Prata, 2012).

Dessa forma é proposta uma solução para síntese de supervisores em RPC, utilizando a ferramenta CPN Tools (Jensen & Kristensen, 2009).

Observe que um lugar em uma RPC pode conter fichas de diferentes cores, as quais podem ou não sofrer restrições (controle), independentemente. Nesse caso, definem-se dois conjuntos de fichas:

- fichas de restrições, que são as fichas que estão submetidas a alguma restrição (controle).

- fichas complementares de fluxo, que são as fichas do conjunto que não sofrem restrições.

Dado um SED não controlado, uma especificação de controle, o modelo em RPC e a marcação inicial da rede, pode-se obter um supervisor seguindo os passos apresentados por (Menezes, Barroso, & Prata, 2012).

3.2 A fusão de controladores

No caso em que o RCCD é aplicado a uma rede que apresente diferentes conjuntos de cores associados a diferentes lugares, e expressões de arcos compostas por variáveis distintas, podem-se definir as especificações como um só conjunto, sem prejuízo para o sistema modelado, o que leva à Fusão de Controladores (FC). Esse controlador resultante agrega toda ação dos lugares de controle (*Controladores*).

Dada uma RPC com conjuntos de cores representados por variáveis distintas, sejam B_i e M_0 , respectivamente, a matriz das restrições com as relativas fichas complementares de fluxo e a marcação inicial da RPC controlada, então a marcação inicial e a matriz de incidência de fusão do supervisor são, respectivamente,

$$M_0 = \sum_{i=1}^r B_i + \mathcal{L}_i M_0, \quad (6)$$

$$\mathcal{H} = -(\sum_{i=1}^r \mathcal{L}_i)C. \quad (7)$$

A prova matemática deste teorema pode ser vista em (Menezes, Barroso, & Prata, 2012).

4 Metodologia para a conversão de uma RPC para LD

4.1 Contexto industrial

A programação de um robô numa célula de manufatura pode ser simplificada pela inserção de um CLP, o qual tratando as informações de uma rede, envia ao robô somente os pulsos de ordem de operação. Desta forma, ganha-se na redução da complexidade da programação do robô, uma vez que este será, para o sistema, um simples atuador (*slave*) de manufatura flexível (FMS).

Portanto, a metodologia consiste em:

- Modelar o FMS em RPC sem controle;
- Implementar um algoritmo de trabalho no FMS via CLP, baseado no modelo RPC, usando-se a metodologia de conversão;
- Baseado nas restrições, aplicar o RCCD para encontrar o(s) supervisor(es), conforme 3.1;
- Realizar a fusão dos supervisores, conforme 3.2;
- Incluir o(s) lugar(es) de controle no modelo RPC do FMS com a fusão de supervisores conforme passo anterior;
- Incluir o controle no algoritmo de trabalho no FMS via CLP usando a metodologia de conversão;

- Programar o robô como *slave* do CLP.

A ação acima pode ser usada no contexto industrial onde há algoritmos de controle já implementados nos CLP's de FMS, ou ainda pode ser usada para o projeto de FMS. Desta forma, explora-se o quanto baixo é o esforço de implementação e/ou alteração na programação do CLP para implementar o controle supervisorio por RCCD.

4.2 Descrição de célula de manufatura flexível (FMS)

Esta seção apresenta a aplicação de RCCD na implantação do controle supervisorio em uma célula de manufatura robotizada, onde foram usados: a) 1 CLP TWDLCAE40DRF; b) 1 Robô Mitsubishi modelo RV-A2J; c) 1 conjunto de 8 botoeiras sem identificação; d) 1 conjunto de peças de duas cores distintas.

Seja a célula de manufatura apresentada na Figura 1.

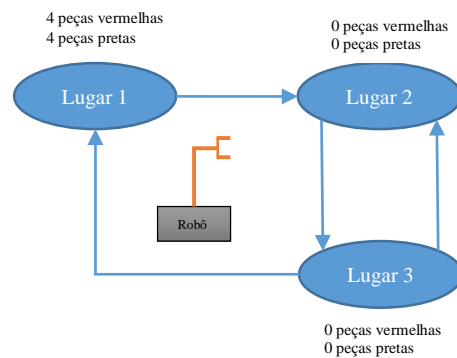


Figura 1 - Célula de manufatura robotizada

Nela, um robô precisa posicionar peças de diferentes cores entre 3 lugares e com um *loop* de reprocesso em caso de falha de controle de qualidade, representado pela seta que liga o lugar 3 ao lugar 2.

Da forma em que se encontra o sistema, o robô pode pegar peças **aleatoriamente**, e sem controle da quantidade de peças, entre quaisquer lugares. Ocorre que a capacidade real de processamento e de reprocesso nos lugares 2 e 3 é limitada a uma única peça de cada cor.

Esta situação caracteriza a ausência de um controle e, portanto, torna oportuna a aplicação do RCCD com as restrições de número de peças nos lugares supracitados.

4.3 Modelagem do FMS sem controle

Na rede da Figura 2 é apresentado o modelo do comportamento não controlado do robô. Observe que enquanto houver fichas em $P1$ (peças disponíveis para a execução de operações fabris), a transição $t1$ estará habilitada e, assim, pode-se colocar até 8 fichas no lugar $P2$. Mas, se $P2$ for um lugar limitado, é necessário exercer controle para limitar o número de fichas (peças) nesse lugar. O mesmo ocorre com $P3$.

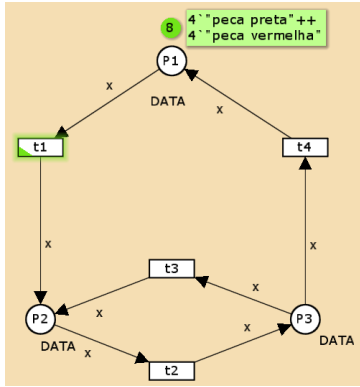


Figura 2 - Modelagem do FMS em RPC

4.4 Conversão do modelo sem controle no programa diagrama Ladder para CLP

A primeira ação é converter o modelo sem controle em um programa de CLP. Nesta ação, como $t1$ possui somente um lugar de entrada e um de saída, então o disparo de $t1$, para cada cor, é convertido no trecho de programa apresentado nas figuras 3 e 4.

Se no lugar $P1$ houvesse mais fichas de outras cores, bastaria acrescentar uma linha de programa com uma entrada externa para cada cor.

Assim ' n ' cores corresponderiam a ' n ' linhas de programa.

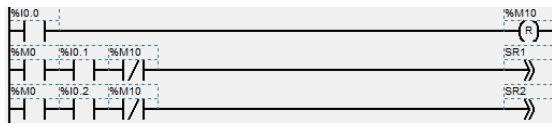


Figura 3 - linha de programa do CLP chamando respectiva sub-rotina

Na Figura 3 são mostradas duas rotinas de varredura, realizadas quando um sensor de proximidade ativa uma das oito entradas do CLP.

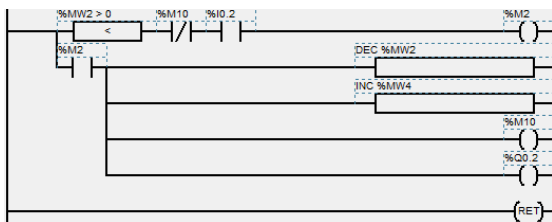


Figura 4 - sub-rotina do CLP condição de saída de $P1$

A Figura 4 apresenta a ação executada quando blocos de memórias (" $\%MW2$ " e " $\%MW4$ ") recebem respectivamente decremento e incremento representando uma ficha que sai do lugar $P1$ ($P4$) para o lugar $P2$ ($P3$), conforme Figura 2.

4.5 Aplicação de RCCD

Seja a célula manufatura apresentada na Figura 1. Seja a Figura 2 a modelagem desta em RPC. Observa-se que esta possui apenas um conjunto de cores (DATA) relativo a todos os lugares da rede e que a todos os arcos é associada a variável x . Nesse caso,

fazendo uso do algoritmo RCCD (Menezes, Barroso, & Prata, 2012) temos que: $D = D_x$ e $C = C_x$.

Impondo as especificações de restrições descritas em 4.2, $M(p2) \leq 1'$ "peça preta" e $M(p3) \leq 1'$ "peça vermelha" tem-se, da RPC não controlada, a matriz de incidência relativa à variável x em (10).

$$D^x = \begin{pmatrix} -1(x) & 0 & 0 & 1(x) \\ 1(x) & 1(x) & -1(x) & 0 \\ 0 & -1(x) & 1(x) & -1(x) \end{pmatrix} \quad (8)$$

o que implica na matriz de referência de incidência, da RPC não controlada, relativa à variável x em (11):

$$C^x = \begin{pmatrix} -1 & 0 & 0 & 1 \\ 1 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 \end{pmatrix} \quad (9)$$

O vetor L indica os lugares que sofrem as restrições:

$$\mathcal{L} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (10)$$

Assim, a matriz C_c , do supervisor:

$$C_c = -\mathcal{L}C = \begin{pmatrix} -1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 1 \end{pmatrix} \quad (11)$$

mostra que o supervisor possui dois lugares de controle, $PC1$ e $PC2$.

A primeira linha da matriz refere-se à restrição $M(p2) \leq 1'$ "peça preta" e diz que o lugar de controle $PC1$ é entrada para as transições $t1$ e $t2$ e saída para $t3$. A segunda linha refere-se à restrição $M(p3) \leq 1'$ "peça vermelha" e diz que o lugar de controle $PC2$ é entrada de $t3$ e saída de $t2$ e $t4$. A seguir, as marcações iniciais dos lugares de controle $PC1$ e $PC2$ são adquiridas em (14), sabendo-se que a matriz B que apresenta a soma das restrições com as respectivas fichas complementares de fluxo é:

$$B = \begin{pmatrix} 1''a_1'' + 2''a_2'' \\ 1''a_2'' + 2''a_1'' \end{pmatrix} \quad (12)$$

e a marcação inicial da RPC não controlada é

$$M_0 = \begin{pmatrix} 2''a_1'' + 2''a_2'' \\ 0 \\ 0 \end{pmatrix} \quad (13)$$

Dessa forma,

$$M_{c_0} = \begin{pmatrix} 1''a_1'' + 2''a_2'' \\ 1''a_2'' + 2''a_1'' \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2''a_1'' + 2''a_2'' \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1''a_1'' + 2''a_2'' \\ 1''a_2'' + 2''a_1'' \end{pmatrix} \quad (14)$$

Onde: " a_1 " é uma peça preta e " a_2 " uma vermelha.

As linhas da matriz M_{c0} são as marcações iniciais dos lugares de controle $PC1$ e $PC2$.

4.6 A fusão dos controladores da RPC

Os controladores $PC1$ e $PC2$ satisfazem às condições de 3.2 e, portanto, a fusão destes produz o lugar de controle $Pcr1$. O qual tem a matriz de incidência de fusão:

$$\mathcal{H} = (-1 \ 0 \ 0 \ 1) \quad (15)$$

e a marcação inicial da fusão

$$M_0 = (1' "a_1" + 1' "a_2") \quad (16)$$

O supervisor resultante da fusão ($Pcr1$), nesse caso, passa a exercer o mesmo controle que os controladores $PC1$ e $PC2$.

4.7 Modelagem do controle supervisorio após a fusão de controladores

Fazendo-se uso da técnica de fusão de controladores, 4.6, obtém-se uma rede controlada com um modelo mais simplificado, o qual, por esta razão, encontra elevada atratividade industrial. A Figura 5 apresenta o resultado desta rede após a fusão dos controladores com a manutenção das restrições impostas no RCCD.

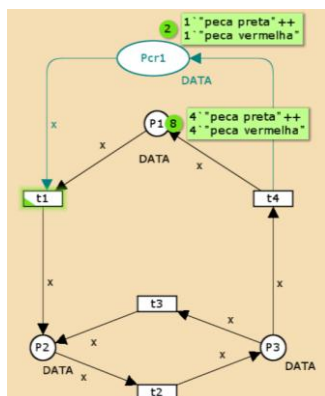


Figura 5 - RPC com fusão de controladores

4.8 Implementação do programa com controle em CLP

Uma vez concluídas as etapas anteriores, ou ainda de posse de um programa de CLP já pronto, torna-se simples a implementação de controle em CLP, pois, acrescentar um lugar de entrada a uma transição equivale a um "and" na programação do CLP. Basta, então, incluir blocos de memórias, na condição "and", nas linhas de programa que equivalem aos dois arcos da Figura 5, os quais conectam as transições "t1" e "t4" ao lugar "Pcr1". As Figuras 6, 7 e 8 mostram os impactos da inclusão do controle na programação do CLP. Percebe-se que poucas alterações no programa permitem a execução do controle.

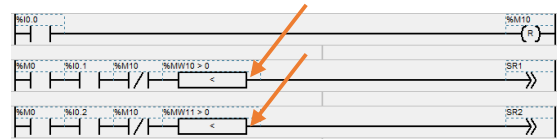


Figura 6 - Programa de CLP com supervisorio

A inclusão de memórias nas linhas de programa na Figura 6, apresenta a primeira alteração no programa original do CLP (Figura 3).

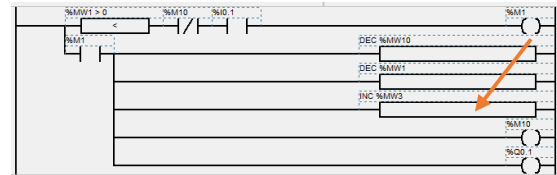


Figura 7 - Sub-rotina do CLP com alteração do controle supervisorio condicao de saida de Pcr1.

A inclusão de memória para registrar o decremento de $Pcr1$, na Figura 7, apresenta a segunda alteração na programação do CLP (Figura 4).

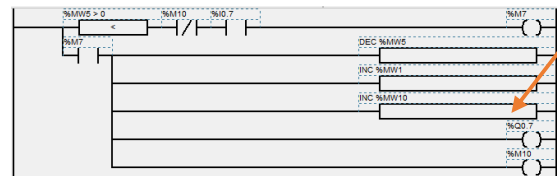


Figura 8 - Sub-rotina do CLP com alteração do controle supervisorio condicao de chegada de Pcr1.

A inclusão de memória para registrar o incremento de $Pcr1$, na Figura 8, apresenta a terceira e alteração na programação do CLP.

5 Discussões

5.1 Por que não realizar todo o controle no próprio robô?

Fundamentalmente por conta da elevada complexidade para realizar a implementação e da dificuldade de alteração das rotinas de programação, somado às limitações de memórias e de processamento dos variados modelos de robôs encontrados.

Todavia, a maioria dos robôs articulados verticais podem ser acionados via CLP. Assim, uma operação complexa pode ser simplificada de modo que o robô não precise tratar dos conflitos de chamadas de sub-rotinas, varreduras e demais, representando ganhos de energia elétrica, processamento, memória e aumento de vida útil do mesmo. Desta forma, um CLP trata as entradas da FMS e as envia ao robô como ordem de operação.

5.2 Qual a garantia de replicabilidade?

Desde que a RPC seja modelada corretamente, torna-se fácil a conversão do mesmo no algoritmo implementado no CLP, como demonstrado na Secção 4. Caberá, entretanto, ao programador atentar para a inclusão de memórias no CLP as quais representam as cores das fichas no lugar "Pcr1" e, na sequência,

incluir estas memórias, como condição “and”, na linha de habilitação das transições que estavam livres, ou seja, sem restrição. Obviamente, um algoritmo que use lógica de acionamento “ou” usará a condição “or” em lugar da “and” descrita neste trabalho.

6 Conclusão

Neste trabalho é apresentada uma metodologia de conversão de modelos RPC para LD com e sem supervisão. Para a supervisão foi usado o método RCCD. Mostrou-se a praticidade da conversão, facilitando o trabalho do programador em LD. E, também que com poucas alterações num programa de CLP, pode-se alcançar o controle das operações de uma FMS robotizada sem comprometer as rotinas de programação já existentes no CLP, e, essencialmente garantindo o alcance do controle das operações.

A vantagem desta metodologia sobre as metodologias anteriores citadas são o uso de RPC enquanto as demais usam somente RP lugar-transição. Uma outra vantagem é que podemos converter um modelo não controlado e após incluir o controle com pequenas modificações no LD.

Atualmente estamos trabalhando na automatização desta metodologia.

Referências bibliográficas

- Barghash, M. A., Abuzeid, O. M., Al-Rabadi, A. N., & Jaradat, A. M. (2011). Petri Nets and Ladder Logic for Fully-Automating and Programmable. *American J. of Engineering and Applied Sciences*, 252-264.
- Barkaoui, K., Chaoui, A., & Zouari, B. (1997). Supervisory control of discrete event systems based on structure theory of petri nets. *IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation*.
- Cury, J. (2001). Teoria de controle supervisorio de sistemas a eventos discretos.
- Gomaa, M. M. (2011). Petri net to ladder logic diagram converter and a batch process simulation. *ARN Journal of Engineering and Applied Sciences*.
- Hopcroft, J., Motwani, R., & Ullman, J. (2006). Introduction to Automata Theory Languages and Computation.
- Jensen, K., & Kristensen, L. M. (2009). Coloured Petri Nets. Em *Modelling and Validation of Concurrent Systems* (p. 8). Aarhus: University of Aarhus.
- Menezes, F. A., Barroso, G. C., & Prata, B. d. (2012). Restrições de controle sobre cores decompostas: uma proposta no controle supervisorio de sistemas a eventos discretos utilizando redes de petri coloridas. *Revista Controle & Automação/Vol.23 no.3/Maio e Junho*.
- Moody, J., & Antsaklis, P. (1998). Supervisory Control of Discrete Event Systems Using Petri net,. *Kluwer Academic Publishers*.
- Morais, C. C., & Castrucci, P. d. (2007). Engenharia de automação industrial. Rio de Janeiro: LTC.
- Murata, T. (1989). Petri nets: Properties, analysis and applications,. *Proceedings of the IEEE* 77(4): 541–580.
- Prata, B. A., Barroso, G. C., & Arruda, J. B. (2008). Um novo método para controle supervisorio de sistemas a eventos discretos baseado em redes de petri coloridas. *XV Simpsio Brasileiro de Pesquisa Operacional*.
- Ramadge, P. J., & Wonham, W. M. (1987). Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization* 25(1): 206–230.
- Ramadge, P., & Wonham, W. (1989). The control of discrete event systems . *Proceedings of the IEEE* 77(1): 81–98.