

ESCOLHA ENTRE EVENTOS CONTROLÁVEIS EM IMPLEMENTAÇÃO DE CONTROLE SUPERVISÓRIO EM CLPS

RENAN SEBEM*, ANDRÉ BITTENCOURT LEAL*

**Grupo de Automação de Sistemas e Robótica - GASR*
Rua Paulo Malschitzki, s/n - Campus Universitário Prof. Avelino Marcante
Universidade do Estado de Santa Catarina - UDESC
Joinville, Santa Catarina, Brasil

Emails: renansebem@gmail.com, leal@joinville.udesc.br

Abstract— The supervisory control theory consists in a formal approach to solve discrete event systems control problems which has a non-blocking solution guarantee and minimally restrictive. However, the implementation of supervisory control structure in programmable logic controllers a control action may be blocking or restrictive. Thus, this work presents a proposal to solve the choice problem, regards the randomness period from manufacturing processes and the periodicity of the scan cycle from the programmable logic controllers, ensuring the implemented control logic results a minimally restrictive and non-blocking action.

Keywords— Supervisory Control Theory, Programmable Logic Controllers, Implementation Problems, Choice Problem, Discret Event Systems.

Resumo— A teoria de controle supervisório consiste em uma abordagem formal para a solução de problemas de controle de sistemas a eventos discretos na qual se tem a garantia de uma solução não bloqueante e minimamente restritiva. Entretanto, na implementação da estrutura de controle supervisório em controladores lógicos programáveis pode-se obter uma ação de controle bloqueante ou restritiva. Assim, neste trabalho apresenta-se uma proposta de solução para o problema da escolha, com base na aleatoriedade do período de processos de manufatura e na periodicidade do ciclo de varredura do controlador lógico programável, de modo que a lógica de controle implementada resulte em uma ação minimamente restritiva e não bloqueante.

Palavras-chave— Teoria de Controle Supervisório, Controlador Lógico Programável, Problemas de Implementação, Problema da Escolha, Sistemas a Eventos Discretos.

1 Introdução

O desenvolvimento do controle das plantas de automação da manufatura é uma tarefa complexa e se não bem feito oferece risco à segurança de operadores e equipamentos. Por estes motivos este desenvolvimento demanda um controle seguro contra falhas.

Utilizando a Teoria de Controle Supervisório (TCS) obtém-se a lógica de controle ótima, minimamente restritiva e não bloqueante que garante as especificações do projetista.

A TCS possui um formalismo matemático para cálculo de supervisores, e pode ser representado por linguagens ou autômatos (RAMADGE and WONHAM, 1989). Um supervisor obtido por intermédio da TCS é utilizado para controlar Sistemas a Eventos Discretos (SEDs), usualmente aplicado na academia e literatura para controle de plantas de manufatura (LEAL et al., 2012)(FABIAN and HELLGREN, 1998).

Na TCS os eventos são classificados em controláveis e não controláveis, e são espontaneamente gerados pela planta. Assim o supervisor deve apenas desabilitar ações de controle que podem levar o sistema a apresentar comportamentos indesejados.

Ainda, existem duas principais abordagens para cálculo de supervisores, a abordagem monolítica (RAMADGE and WONHAM, 1989) e a modular local (QUEIROZ, 2000).

Apresentada por Ramadge e Wonham (1989) há mais de duas décadas, a TCS ainda é pouco utilizada na indústria (FABIAN and HELLGREN, 1998). Uma explicação para isto é a falta de ferramentas que facilitem a implementação da lógica de controle obtida na TCS, pois na implementação da lógica de controle surgem problemas que não existem na teoria (FABIAN and HELLGREN, 1998).

Os principais problemas de implementação de controle supervisório na prática são devidos ao uso de Controladores Lógicos Programáveis (CLPs) para controlar as plantas de manufatura. Estes problemas são o problema da escolha, da causalidade, da sincronização inexata, da simultaneidade de eventos e do efeito avalanche (FABIAN and HELLGREN, 1998).

2 Problema da Escolha

Segundo a TCS os eventos controláveis e não controláveis são gerados pela planta. Na prática os eventos controláveis não serão gerados pela planta, e deve-se implementar uma lógica que gerará os eventos controláveis. Este problema é conhecido como o problema da causalidade (FABIAN and HELLGREN, 1998).

Devido ao problema da causalidade, é necessário que os eventos controláveis sejam gerados pelo controlador do sistema. Assim, toda vez que o supervisor alcança um estado onde exista um evento

controlável habilitado, e nenhum evento não controlável foi gerado pela planta, o controlador gerará o evento controlável. O problema acontece quando existem dois ou mais eventos controláveis habilitados em um mesmo estado (RAMADGE and WONHAM, 1989). Neste caso, apenas um evento deve ser escolhido para dar evolução aos estados do supervisor, e dependendo da maneira como a escolha é feita a ação de controle pode ser bloqueante ou restritiva (FABIAN and HELLGREN, 1998), causando o problema da escolha.

Toma-se como exemplo uma parte de uma planta de manufatura, composta por um robô e uma máquina. O robô transporta dois tipos diferentes de peças para a máquina, que executa um processo diferente para cada tipo de peça. O autômato da planta do robô é mostrado na Figura 1, onde os eventos ini_1 e fim_1 representam, respectivamente, o início e fim de transporte da peça do tipo 1 para a máquina, e ini_2 e fim_2 da peça do tipo 2.

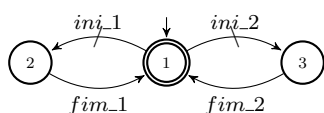


Figura 1: Representação da planta do robô.

A planta da máquina é representada pelo autômato da Figura 2. Onde as transições ini_{M_1} e fim_{M_1} representam o início e fim de operação da máquina para a peça do tipo 1, e ini_{M_2} e fim_{M_2} para a peça do tipo 2.

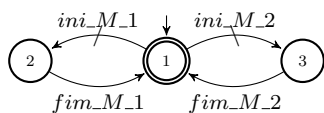


Figura 2: Representação da planta das máquinas.

A especificação de controle mostrada na Figura 3, proíbe o início de operação da máquina antes do fim de transporte de qualquer peça para máquina. Além disso, só permite que a máquina execute o processo referente à cada peça.

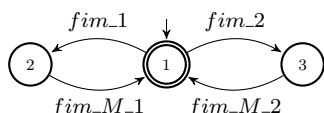


Figura 3: Especificação de controle.

O supervisor monolítico obtido a partir das plantas e especificações mostradas acima é composto de 7 estados e 8 transições. Os primeiros estados deste supervisor são mostrados na Figura 4.

As transições tracejadas representam a continuidade do supervisor, que não é mostrada na

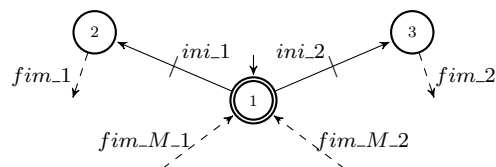


Figura 4: Parte inicial do supervisor.

Figura 4.

Observando a Figura 4, na teoria, o sistema deverá permanecer no estado 1 até que um dos eventos, ini_1 ou ini_2 , seja espontaneamente gerado pela planta. Na prática o sistema nunca sairá do estado 1 se a lógica implementada não escolher um dos eventos para gerar. Ou seja, o controlador deverá escolher entre gerar ini_1 ou ini_2 .

Além disso, segundo Fabian e Hellgren (1998), em uma implementação em linguagem de diagrama de escada (IEC, 2003) o evento gerado pela lógica implementada sempre será o que estiver descrito anteriormente nas linhas do código. Dessa forma a escolha é feita no momento em que o programa é escrito. Este mesmo evento sempre será gerado pela lógica implementada quando o supervisor alcançasse determinado estado (FABIAN and HELLGREN, 1998).

A Figura 5 mostra o código de CLP em diagrama de escada (FABIAN and HELLGREN, 1998) referente à implementação da lógica de controle representada pelo autômato da Figura 4, considerando uma abordagem monolítica e sem um algoritmo para a resolução do problema da escolha, onde o evento ini_1 sempre será gerado pela lógica implementada, a cada vez que o supervisor alcançar o estado 1.

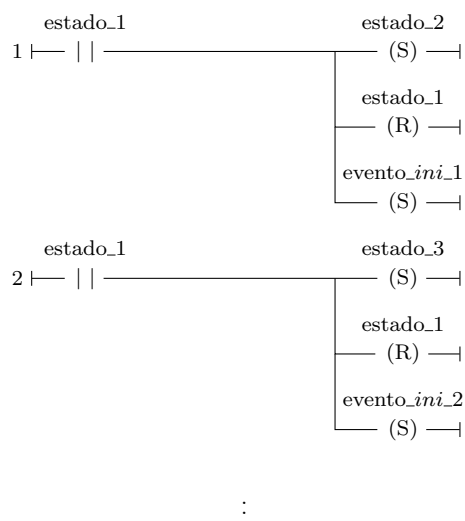


Figura 5: Parte inicial do supervisor representado em diagrama de escadas, sem resolução do problema da escolha.

Retornando ao autômato da Figura 4, se o evento ini_1 estiver descrito no código em uma

linha anterior ao evento *ini_2*, então o estado 3 estará inacessível pela lógica implementada, fazendo com que peças do tipo 2 não sejam processadas pela máquina (MALIK, 2002) (DIETRICH et al., 2001). Assim, a lógica implementada passa a executar apenas metade do supervisor.

Uma solução para o problema da escolha foi proposto por Pinotti et al. (2010). Esta solução consiste em um contador que incrementa seu valor a cada ciclo de varredura. O limite da contagem é o número de eventos envolvidos no problema da escolha em cada estado, de forma que é necessário um contador para cada quantidade de eventos envolvidos, ou seja, se houver problemas da escolha com dois, três e quatro eventos, serão necessários três contadores diferentes.

Esta solução necessita que o projetista conheça todos os supervisores a serem implementados e saiba em quais estados existirão o problema da escolha e além disso, quantos eventos estarão envolvidos. Para o exemplo usado em (PINOTTI et al., 2010) cujo supervisor possui poucos estados (3 estados), esta solução é viável, mas para supervisores com centenas de estados, por exemplo, esta solução se torna difícil de ser implementada sem o auxílio de um ferramenta computacional.

Cruz (2011) propõe uma solução que consiste em alternar uma variável booleana entre valores falso e verdadeiro a cada ciclo de varredura do Controlador Lógico Programável. Esta variável permite ora geração de um evento, ora outro, fazendo com que a probabilidade de escolha de cada evento seja igual e aleatória. Desta forma, o estado 3 da Figura 4 deixaria de ser inacessível (CRUZ, 2011) (LEAL et al., 2012).

Porém, quando existem mais que dois eventos controláveis habilitados em um mesmo estado, Cruz (2011) propõe duas soluções. Uma destas é aumentar o número de variáveis booleanas, sendo necessário um bit para cada dois eventos, e aumentando significativamente a complexidade da solução. A outra consiste em fazer a escolha entre um evento e um par de eventos, considerando que este par seja insensível ao entrelaçamento (FABIAN and HELLGREN, 1998).

Assim como o trabalho de Pinotti et al. (2010), esta solução também necessita que o projetista identifique e analise os estados que sofrem o problema da escolha.

Outras propostas para a estrutura de implementação são interessantes para ajudar na resolução de problemas como a utilização de um *dispatcher* (BASILE and CHIACCHIO, 2007), porém detalhes da resolução não são mostrados pelos autores.

3 Resolução do Problema da Escolha

Em uma solução ideal para o problema da escolha, a escolha de um evento deve ser aleatória e a

probabilidade de escolha de cada evento deve ser igual para todos os eventos. Também, não deve excluir nenhuma parte do supervisor, ou seja, não tornar nenhuma parte inacessível. Outro ponto é que o algoritmo deve reconhecer automaticamente se o problema existe, descartando a necessidade de análise prévia do projetista. Também é importante que a solução exija o mínimo possível dos recursos do controlador.

Além disso, o problema da escolha não se apresenta em todos os estados de modo que o problema deve ser tratado apenas quando for necessário.

Esta solução está alinhada com a estratégia de implementação de controle supervisorio em CLP proposta por (SEBEM, 2012) e estratégia *memory safe* proposta por (LOPES et al., 2011; LOPES et al., 2012).

3.1 Proposta de Solução

Basicamente, a proposta consiste na utilização de uma variável *Random*, do tipo inteira, que será incrementada a cada ciclo de varredura do CLP de 0 a um valor máximo ($lim_{máx}$).

Quando o problema da escolha existir em um determinado estado, o controlador deverá escolher entre n eventos habilitados neste estado, que serão indexados de 1 a n .

O resto da divisão inteira de *Random* por n , também conhecida como operação de módulo de *Random* por n , retornará um valor entre 1 a n . Assim o resultado da escolha será o resto da divisão inteira de *Random* por n .

Destacando que ao incrementar a variável *Random*, eventualmente esta atingirá seu valor máximo, sendo necessário que seja reinicializada para 0. Em alguns CLPs, quando a variável do tipo inteira é incrementada em seu valor máximo, esta passa a valer 0 (SCHNEIDER, 2011).

Na Figura 6 é mostrado o fluxograma do algoritmo proposto para resolução do problema.

3.2 Probabilidade

Se o valor máximo, $lim_{máx}$, for proporcional ao número de eventos envolvidos na escolha, a probabilidade será distribuída igualmente entre todos os eventos. Porém desta forma será necessário um valor máximo diferente para diferentes números de eventos envolvidos. Desta forma, quanto maior for esse valor máximo, melhor distribuída será a probabilidade, seja qual for a quantidade de eventos envolvidos.

As probabilidades máxima e mínima da ocorrência de qualquer evento envolvido na escolha serão:

$$P_{máx} = \frac{1}{n} + \frac{n - \text{mod}((lim_{máx} + 1), n)}{n * (lim_{máx} + 1)} \quad (1)$$

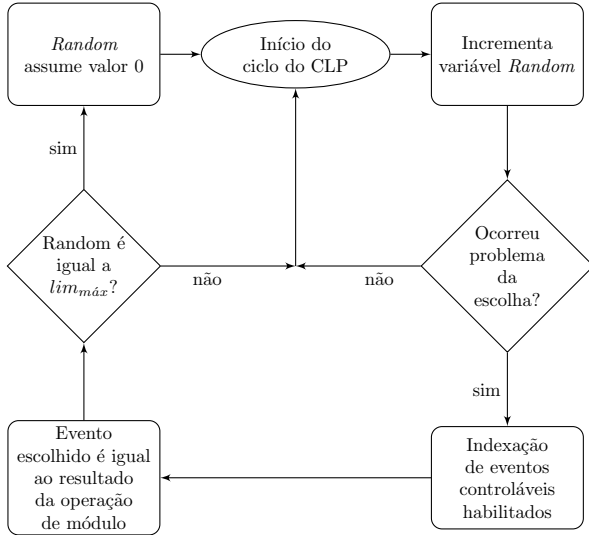


Figura 6: Fluxograma do algoritmo de resolução do problema.

$$P_{min} = \frac{1}{n} - \frac{\text{mod}((\text{lim}_{máx} + 1), n)}{n * (\text{lim}_{máx} + 1)} \quad (2)$$

Onde:

- n é o número de eventos envolvidos;
- $\text{lim}_{máx}$ é o valor máximo que a variável *Random* pode representar;
- $\text{mod}((\text{lim}_{máx} + 1), n)$ é igual ao resto da divisão inteira de $\text{lim}_{máx} + 1$ por n .

E o erro máximo entre a probabilidade máxima e mínima é dado por:

$$e = \frac{1}{\text{lim}_{máx} + 1} \quad (3)$$

3.3 Aleatoriedade

A variável utilizada como base para a operação módulo que resulta em uma escolha de evento é incrementada a cada ciclo de varredura do CLP. Por si só, isto não oferece alguma aleatoriedade à escolha. Porém, como dito anteriormente, o problema da escolha não se manifesta em todos os estados, e nos estados que se manifesta, manifesta-se diferentemente.

Além disso, a evolução de estados de um supervisor no tempo depende dos eventos não controláveis do sistema, que por sua vez são gerados pela planta de manufatura. As plantas de manufatura são de natureza mecânica, onde os processos destas possuem tempos de execução maiores

e menos precisos que o tempo do ciclo de varredura do CLP. Desta forma a própria ocorrência do problema da escolha torna-se aleatória no tempo.

Testes foram realizados no Laboratório de Automação da Manufatura (LAMAN) na Universidade do Estado de Santa Catarina (UDESC), para a comprovação da aleatoriedade do período de processos de manufatura. Os equipamentos utilizados para testes foram uma esteira, um *pallet*, um sensor óptico e um CLP.

Para isto, um teste simples é considerado: a esteira permanece ligada durante todo o teste com o *pallet* em cima desta e o CLP registra o tempo para cada volta completa do *pallet* na esteira, através do sinal enviado pelo sensor óptico ao CLP.

Os resultados são mostrados na Tabela 1, e indicam que existe aleatoriedade do período de uma volta completa de um *pallet* em uma esteira.

Tabela 1: Períodos de uma volta completa de um *pallet* em uma esteira de manufatura.

Volta	Duração (ms)
1	25894
2	25547
3	25666
4	25652
5	25434
6	25737
7	25580
8	25736
9	25676

Analisando os dados da Tabela 1 tem-se como duração média 25658 *ms* e desvio padrão de 128,39 *ms*. Isso significa que o desvio padrão é muito maior que o período de ciclo de varredura do CLP utilizado no teste, que varia entre 1 e 2 *ms* (SCHNEIDER, 2011).

3.4 Exemplo

Considerando o autômato da Figura 7 para demonstração, e supondo todos os eventos estão habilitados e abordagem monolítica, o problema da escolha existe no estado 1 do supervisor. Utilizando uma variável *Random* do tipo inteira positiva de 16 bits, o número total de eventos controláveis do supervisor, N , é 4 e o número de eventos envolvidos no problema da escolha no estado 1, n , é 3. Assim, para uma variável de n bits podemos representar números de 0 a $2^n - 1$, desta forma para 16 bits temos $\text{lim}_{máx} = 2^{16} - 1 = 65535$.

Na Tabela 2 mostram-se os resultados obtidos com o algoritmo de resolução do problema, sendo que na segunda coluna é mostrado os índices dos eventos envolvidos no problema, e a terceira coluna mostra a probabilidade de geração de cada evento pelo controlador.

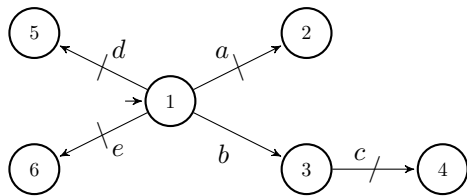


Figura 7: Autômato utilizado para demonstração.

Tabela 2: Resultados obtidos com algoritmo de resolução do problema da escolha.

Evento	Índice do Evento	Probabilidade (%)
a	0	33,334
d	1	33,333
e	2	33,333

As probabilidades máxima e mínima podem ser calculadas usando a Equação 1 e a Equação 2, respectivamente. O erro encontrado através da Equação 3 é igual a 0,001 %.

4 Ferramenta Nadzoru

Esta proposta foi implementada em uma ferramenta chamada Nadzoru (LOPES, 2012). A ferramenta consiste em ambiente para criação e edição de autômatos, realização de cálculos para autômatos (como a obtenção de supervisores), geração automática de código para microcontrolador (linguagem C) e para CLP (linguagem Texto Estruturado), simulador de supervisor e integração de níveis MES e SCADA a partir de autômatos de estados finitos (LOPES, 2012).

É interessante destacar que a proposta deste trabalho pôde ser facilmente utilizada para a geração automática de código.

5 Conclusões

Este trabalho resolve de maneira simples o problema da escolha, utilizando o mínimo de recursos possíveis do controlador, e de uma forma concisa, baseada em equações.

A solução proposta oferece algumas vantagens em relação a métodos anteriores.

Apresenta uma boa aleatoriedade e probabilidade bem distribuída entre os eventos a serem escolhidos, sabendo que o período de execução de plantas de manufatura não são constantes.

Não há necessidade de identificação e análise prévia pelo projetista, em quais estados o problema existirá, quantos eventos estarão envolvidos, se os eventos envolvidos serão insensíveis ao entrelaçamento ou não.

A implementação pode ser realizada independentemente do tipo e fabricante do CLP e da linguagem de programação.

Este método de resolução pode ser utilizado em microcontroladores, uma vez que o microcontrolador possua Real Time Interrupt (RTI) e o processo controlado possua variações no período de tempo maiores que o período da RTI. Além disso, pode ser utilizada para a geração automática de código para microcontrolador.

Agradecimentos

O primeiro autor agradece à Universidade do Estado de Santa Catarina pelo suporte financeiro concedido por intermédio do Programa Institucional de Bolsas de Iniciação Científica (PROBIC).

Referências

- BASILE, F. and CHIACCHIO, P. (2007). On the implementation of supervised control of discrete event systems, *IEEE Transactions on Control Systems Technology* **15**(4): 725–739.
- CRUZ, D. L. L. (2011). *Metodologia para Implementação de Controle Supervisório Modular Local em Controladores Lógicos Programáveis*, Master's thesis, Universidade do Estado de Santa Catarina, Joinville, SC.
- DIETRICH, P., MALIK, R., WONHAM, W. M. and BRANDIN, B. A. (2001). Implementation considerations in supervisory control, *Proceedings of SCODES'01*, Paris, France.
- FABIAN, M. and HELLGREN, A. (1998). PLC-based implementation of supervisory control for discrete event systems, *37th IEEE Conference on Decision and Control*, Vol. 3, pp. 3305 – 3310.
- IEC (2003). International standard iec 61131-3: Programmable logic controllers - part 3: Programming languages.
- LEAL, A. B., CRUZ, D. L. L. and HOUNSELL, M. S. (2012). Plc-based implementation of local modular supervisory control for manufacturing systems, *Manufacturing System, Dr. Faieza Abdul Aziz (Ed.), InTech*.
- LOPES, Y. K. (2012). *Integração dos níveis MES, SCADA e controle da planta de manufatura com base na teoria de linguagens e autômatos*, Mestrado em engenharia elétrica, Universidade do Estado de Santa Catarina, Joinville, SC.
- LOPES, Y. K., HARBS, E., LEAL, A. B. and ROSSO Jr., R. S. U. (2011). Proposta de implementação de controle supervisório em microcontroladores, *X Simpósio Brasileiro de Automação Inteligente - SBAI 2011*.

- LOPES, Y. K., Leal, A. B., Jr., R. S. U. R. and Harbs, E. (2012). Local modular supervisory implementation in microcontroller, *9th International Conference of Modeling, Optimization and Simulation - MOSIM'12* .
- MALIK, P. (2002). Generating controllers from discrete-event models, *F. Cassez, C. Jard, F. Laroussinie, and M. D. Ryan, editors, Proc. of MOVEP 2002*.
- PINOTTI, A. J., LEAL, A. B. and OLIVEIRA, D. S. (2010). Uma proposta para implementação da estrutura de controle supervísório em controladores lógicos programáveis, *XVIII Congresso Brasileiro de Automática - CBA*, Bonito - MS., Brasil, pp. 2830–2837.
- QUEIROZ, M. H. d. (2000). *Controle supervísório modular de sistemas de grande porte*, Master's thesis, Universidade Federal do Estado de Santa Catarina.
- RAMADGE, P. J. and WONHAM, W. M. (1989). The control of discrete events system, *Proceedings of the IEEE*, Vol. 77, pp. 81–98.
- SCHNEIDER (2011). Unity pro: Program languages and structure reference manual, www.schneider-electric.com/. Acessado em 04/2013.
- SEBEM, R. (2012). *Estudo e Implementação de Método para Geração Automática de Código para CLP*, Trabalho de conclusão de curso (Bacharelado em Engenharia Elétrica), Universidade do Estado de Santa Catarina. Joinville, SC.