VERIFICAÇÃO DE AUTÔMATOS TEMPORIZADOS PARA VALIDAÇÃO E ANÁLISE DE DESEMPENHO DE LÓGICA DE CONTROLE SINTETIZADA PELA TEORIA DE CONTROLE SUPERVISÓRIO

JOÃO AURÉLIO V. RODRIGUEZ E ANTONIO E. C. DA CUNHA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA (PGEE)
INSTITUTO MILITAR DE ENGENHARIA (IME)
PRAÇA GENERAL TIBÚRCIO, 80, PRAIA VERMELHA, 22.290-270, RIO DE JANEIRO, RJ
E-MAILS: JOAO.RODRIGUEZ@OUTLOOK.COM, CARRILHO@IME.EB.BR

Abstract — In this work a form of validation and performance analysis of a control logic synthesized by the Supervisory Control Theory (SCT) through formal verification of timed automata is proposed. A case study of the automation of an industrial system for bonding glass in truck cabins is presented.

Keywords — Discrete Event Systems, Supervisory Control, Verification, Validation, Performance Analysis.

Resumo — Neste trabalho demonstra-se uma forma de validação e avaliação de desempenho de uma lógica de controle sintetizada pela Teoria de Controle Supervisório (TCS) por intermédio da verificação formal de autômatos temporizados. Emprega-se como ilustração o estudo de caso da automação de um sistema industrial para colagem de vidros em cabines de caminhões.

Palavras-chave — Sistemas a Eventos Discretos, Controle Supervisório, Verificação, Validação, Análise de Desempenho.

1 Introdução

A Teoria de Controle Supervisório (TCS), iniciada por Ramadge e Wonham em meados da década de 80, trata da síntese de lógicas de controle para Sistemas a Eventos Discretos (SED) (Cassandras e Lafortune, 2008).

A TCS fornece um método automático para geração de lógica de controle, denominada supervisor, dados o modelo do SED a controlar, denominado planta, e as especificações para o comportamento em malha fechada. Os modelos e as especificações são expressos por linguagens e autômatos (Cassandras e Lafortune, 2008).

A lógica de controle sintetizada pela TCS possui garantia de ser *legal*, no sentido de atender às especificações de projeto, *ótima*, no sentido de ser minimamente restritiva ao comportamento do sistema, e *não bloqueante*, no sentido de garantir ausência de *deadlocks* e *livelocks* no sistema em malha fechada (Cassandras e Lafortune, 2008).

Entretanto, muitas abstrações são feitas para a elaboração dos modelos, como no caso deste trabalho, a abstração da passagem do tempo. Não são fornecidas explicitamente formas de validação do sistema em malha fechada antes da implementação propriamente dita, nem são definidas formas para avaliar o desempenho do sistema em malha fechada, que permitam quantificar, por exemplo, os ganhos pela lógica de controle ser minimamente restritiva.

Este trabalho propõe uma forma do emprego da verificação de autômatos temporizados (Alur e Dill, 1994) para fornecer, primeiro, uma forma de validação da lógica de controle gerada pela TCS contra um modelo mais detalhado do sistema e, segundo, uma forma de avaliar o desempenho da lógica de controle.

Para a síntese da lógica de controle no contexto da TCS foram empregados os métodos associados ao Controle Modular Local (CML) e à redução de Supervisores, conforme proposto por (Queiroz e Cury, 2002a). Foi adotada uma abordagem sem o emprego de uma temporização explícita dos eventos, pelo uso de eventos fictícios correspondentes ao término da contagem de certos tempos relevantes. Está em andamento uma modelagem que emprega o Controle Supervisório de SED Temporizados nas abordagens de (Brandin e Wonham, 1993) e de (Cassez et al., 2005).

Para a verificação de autômatos temporizados empregam-se as ferramentas disponíveis no *toolkit* UPPAAL (Behrmann et al., 2006).

Emprega-se como ilustração o estudo de caso da automação de um sistema industrial para colagem de vidros em cabines de caminhões da fábrica *MAN Truck and Bus* de Munique, Alemanha. O que motivou a escolha deste sistema foi a possibilidade de otimizar o processo de colagem, pois o sistema foi originalmente desenvolvido somente com foco na sua funcionalidade na linha de produção. Por outro lado, o sistema apresenta aspectos de temporização interessantes de serem investigados.

A apresentação deste artigo é a seguinte. A Seção 2 apresenta o sistema de colagem de vidros e o problema de controle. A Seção 3 traz a aplicação do CML ao problema de controle do sistema de colagem de vidros. A Seção 4 registra a aplicação da verificação de autômatos temporizados para validação e análise de desempenho da lógica de controle. Por fim, a Seção 5 apresenta alguns comentários conclusivos e as perspectivas de trabalho futuro.

2 Descrição do Sistema

O sistema para colagem de vidros em cabines de caminhões da fábrica *MAN Truck and Bus* de Munique, Alemanha foi concebido para a aplicação de cola nos vidros laterais da parte traseira da cabine e no vidro para brisas bem como para a montagem do vidro para brisas na cabine. O sistema foi motivado

pela precisão dos robôs na aplicação da cola, não somente em relação às coordenadas, mas também em relação ao volume de cola aplicado. Outro fator determinante foi à questão da dimensão (2.242,6mm x 879,5mm) e do peso (29,3kg) do vidro para brisas, que inviabilizava a sua colocação na cabine por um único montador.

A aplicação de cola nos vidros passa por três etapas que são: a aplicação do ativador, a aplicação do *primer* e, por fim, a aplicação da cola propriamente dita. Entre as aplicações ativador/*primer* e *primer*/cola devem ser respeitados tempos mínimos de secagem e, após a aplicação da cola, há tempos máximos para a colocação dos vidros na cabine até a secagem.

Como os vidros laterais são aplicados manualmente, são produzidos mais vidros laterais do que para brisas e os mesmos podem ser aplicados à cabine de dois postos de trabalho antes até um posto de trabalho após a aplicação do vidro para brisas. Devido a esta flexibilidade o tempo de secagem, ou vulcanização, da cola aplicada ao vidro lateral é maior do que a do vidro para brisas.

O sistema de colagem de vidros, ilustrado na figura 1, é composto por dois robôs, o *Robot Side Glass* (RSG) e o *Robot Wind Shield* (RWS), dois *buffers* para depósito intermediário dos vidros, B1 e B2, três esteiras: uma para entrada de vidros laterais (EVL), uma para entrada de vidros para brisas (EVPB) e uma para saída de vidros laterais (SVL), um sistema ótico de medição de coordenadas da cabine, *Perceptron* (PER), e pela entrada de cabines (EC) da linha principal de produção. A seguir os principais componentes do sistema são descritos.

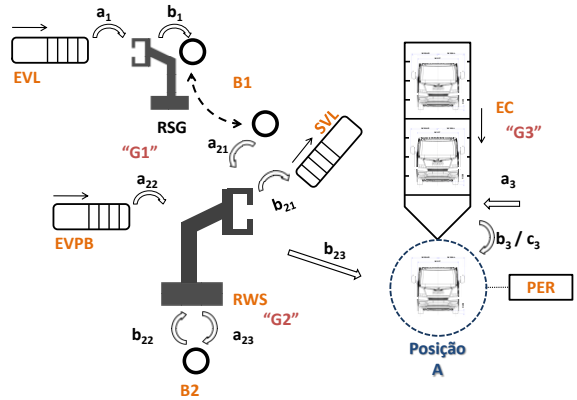

Figura 1. Sistema de colagem de vidros em caminhões.

O robô RWS é utilizado para aplicação do *primer* e do ativador nos vidros laterais. Partindo do princípio que EVL esteja abastecida pelo operador de produção, o RSG pega, por intermédio de ventosas, um par de vidros laterais, posiciona-os para a aplicação do ativador e realiza a aplicação. Após a aplicação do ativador, o robô aguarda 20s e, nesse tempo, faz o processo de autolimpeza e troca o bico para a aplicação do *primer*. O robô então aplica o *primer* e deposita os vidros no *buffer* B1 para aguardar a secagem do produto por 45s.

O robô RWS realiza três processos, descritos a seguir. O primeiro processo consiste na retirada dos vidros laterais do *buffer* B1 por um sistema de ventosas, que então são posicionados para a aplicação da cola. O robô então aplica a cola e deposita os vidros na esteira SVL para que o operador possa retirá-los e aplicá-los à cabine. O segundo trata-se da retirada do vidro para brisas da esteira EVPB para a aplicação do *primer*. Este vidro está previamente disponibilizado por um operador de produção na EVPB já com o ativador aplicado. O robô então aplica o *primer* e deposita o vidro no *buffer* B2 para aguardar o tempo mínimo de 45s para a secagem. O terceiro consiste na retirada do vidro para brisas do *buffer* B2, o posicionamento do mesmo para a aplicação da cola, a aplicação da cola propriamente dita e a montagem do vidro na cabine. Esta operação somente pode acontecer após o robô RWS receber as coordenadas da cabine resultante da medição da mesma pelo *Perceptron*.

O *Perceptron* destina-se a medir as coordenadas principais da cabine e enviá-las a RWS. O processo de medição ocorre após a cabine chegar ao posto de medição, posição A na figura 1, e uma plataforma de alinhamento ser acoplada a mesma.

A linha de produção possui um tempo máximo para tratamento de uma cabine pelo sistema de colagem de 150s, que quando violado, força o acionamento de uma parada de emergência.

O objetivo do controle é coordenar o funcionamento dos robôs e da entrada de cabines para aplicação de ativador, *primer* e cola nos vidros laterais, aplicação de *primer* e cola nos vidros para brisas e colocação do vidro para brisas na cabine, todos respeitando as restrições do tempo de operação do sistema.

3 Aplicação da TCS

Neste trabalho, para a síntese da lógica de controle no contexto da TCS foram empregados os métodos associados ao Controle Modular Local (CML) conforme proposto por (Queiroz e Cury, 2002a).

O primeiro passo é a modelagem da planta, ou o sistema a controlar. Algumas hipóteses simplificadoras foram consideradas, como exposto a seguir.

Primeiro, considera-se que o suprimento de vidros é bem mais rápido que o tempo de processamento do sistema, e que há sempre disponível um vidro lateral e um vidro para brisas. Com isso, omite-se o detalhamento do funcionamento de EVL e EVPB.

Segundo, considera-se que o vidro lateral depositado em SVL é imediatamente encaminhado, consequentemente, omite-se a modelagem de SVL.

Terceiro, considera-se que o *Perceptron* é acionado na chegada da cabine na posição A e o evento que marca que a cabine está pronta para receber o vidro para brisas corresponde ao envio das coordenadas ao robô RWS. Assim, omite-se o detalhamento do funcionamento do *Perceptron*.

A consequência destas hipóteses simplificadoras é que os componentes de interesse a serem modela-

dos do sistema são RSG, RWS e EC, que são modelados, respectivamente, pelos autômatos G1, G2 e G3 na figura 2. Supõe-se que o leitor está familiarizado com a notação gráfica padrão dos autômatos (Cassandras e Lafortune, 2008).

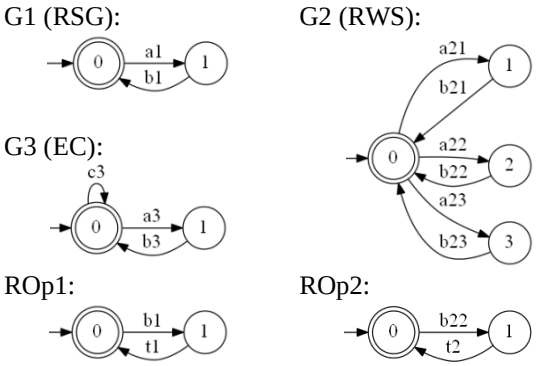

Figura 2 – Modelo da Planta.

Para a modelagem sem a consideração da temporização dos eventos, são necessárias duas restrições operacionais. As restrições ROp1 e ROp2, ilustradas na figura 2, são abstrações de temporizadores para contagem do tempo mínimo de secagem do primer que devem ocorrer, respectivamente, para o vidro lateral após depósito no *buffer* B1 e para o vidro para brisas após o depósito no *buffer* B2.

Na TCS, os eventos são particionados em controláveis e não controláveis. Os eventos controláveis podem ter sua ocorrência inibida, ou desabilitada, enquanto os eventos não controláveis estão sempre habilitados (Cassandras e Lafortune, 2008). A Tabela 1 mostra a descrição dos eventos, o seu atributo de controlabilidade (C para controlável e NC para não controlável) e uma indicação do tempo de atraso para ocorrência do evento, que será empregado mais adiante na seção 4.

O passo seguinte no método é a modelagem de cada especificação de forma isolada, considerando apenas os eventos relevantes (Queiroz e Cury, 2002a). As especificações para a coordenação e correto funcionamento do sistema são as seguintes:

- E1: Evitar o *overflow* (colocar mais que um vidro) e o *underflow* (tentar pegar vidro no vazio) no *buffer* B1;
- E2: Evitar o *overflow* e o *underflow* no *buffer* B2;
- E3: Atender ao tempo mínimo de secagem do *primer* em B1;
- E4: Atender ao tempo mínimo de secagem do *primer* em B2;
- E5: Evitar que a cabine saia sem ter o vidro para brisas colocado;
- E6: Evitar colocar um vidro para brisas no vazio; e
- E7: Evitar o *timeout* da linha de produção.

Os autômatos SP1, SP2 e SP3 na figura 3 foram construídos no intuito de atender às especificações.

Tabela 1. Tabela resumo da descrição dos eventos.

Evento	Descrição	Control.	Atraso [s]
a1	Comando para RSG pegar dois vidros laterais na esteira.	C	-
b1	RSG pega dois vidros laterais na esteira; aplica ativador e espera 20s; troca o feltro de aplicação do ativador; posiciona o bico do *primer*; aplica o *primer*; e deposita os dois vidros laterais em B1. B1 gira 270° e volta-se para RWS	NC	50
a21	Comando para RWS pegar dois vidros laterais em B1.	C	-
b21	RWS pega dois vidros laterais em B1; aplica cola; e deposita vidros laterais na esteira de saída. B1 gira 270° e volta-se para RSG.	NC	20
a22	Comando para RWS pegar vidros para brisas na esteira de entrada.	C	-
b22	RWS pega vidros para brisas na esteira de entrada; aplica *primer*; e deposita vidros para brisas em B2.	NC	30
a23	Comando para RWS pegar vidros para brisas em B2.	C	-
b23	RWS pega vidro para brisas em B2; posiciona o bico aplicador de cola; aplica cola; e posiciona vidro para brisas na cabine.	NC	40
a3	Comando para liberar cabine da posição A e entrar nova cabine.	C	-
b3	Desce o elevador e libera cabine da posição A; apresenta-se uma nova cabine e esta é elevada na posição A; e o *Perceptron* mede as coordenadas da cabine e envia ao controlador de RWS.	NC	50 a 100
c3	*Timeout* da chegada da cabine na linha de produção.	NC	150
t1	*Timeout* do contador de 45s após o depósito do vidro lateral em B1.	NC	45
t2	*Timeout* do contador de 45s após o depósito do vidro para brisas em B2.	NC	45

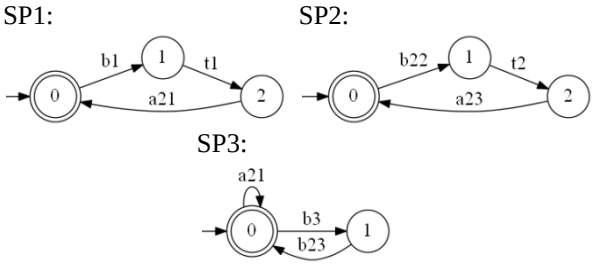

Figura 3 – Especificações.

Na figura 3, o autômato SP1 expressa que, após o depósito dos vidros laterais por RSG em B1 (evento b1), deve-se esperar por, pelo menos, 45s (t1) para que RWS possa retirá-lo (a21). Assim, SP1 atende às especificações E1 e E3. O autômato SP2 é construído de forma análoga a SP1 para atender E2 e E4. O autômato SP3 expressa que RWS só faz a colocação do vidro para brisas (b23) após o recebimento de coordenadas da cabine (b3), que outra colocação só se dará quando chegar outra cabine e que a retirada

de vidros laterais de B1 por RWS fica impedida após a chegada da cabine (*selfloop* de a21 no estado 0). Assim, SP3 procura atender às especificações E5 e E6 de forma direta, e a E7 de forma indireta, ao dar prioridade à colocação do vidro para brisas sobre a passagem de cola nos vidros laterais. Isso foi necessário, pois a modelagem da especificação E7 de uma forma direta levaria a um comportamento não controlável, pois não é possível impedir o evento c3 no estado inicial de G3.

Em seguida, obtém-se as plantas locais para cada especificação, por composição dos componentes da planta que possuam eventos em comum com a especificação. A composição é realizada pela operação de produto síncrono, *sync()*, em que os componentes evoluem em paralelo, sendo que as transições com eventos comuns devem ser sincronizadas (Cassandras e Lafortune, 2008). Os componentes que fazem parte das plantas locais Gloci para as respectivas especificações SPi e o número de estados das plantas locais Gloci são mostrados na Tabela 2.

Tabela 2. Resumo da síntese modular local.

i	SPi	Gloci	Ri	Zi	Zri
1	3	32 G1 G2 ROp1 ROp2	48	32	3
2	3	8 G2 ROp2	8	8	3
3	2	16 G2 G3 ROp2	32	20	2

O próximo passo é o cálculo do autômato que representa o comportamento de cada planta local que satisfaça a respectiva especificação. Para a especificação SPi e a planta local Gloci tal autômato é obtido por Ri = *trim(sync(SPi,Gloci))*, em que *trim()* denota o componente *trim* (Cassandras e Lafortune, 2008). A coluna Ri da Tabela 2 mostra o número de estados dos respectivos autômatos Ri.

Em seguida são calculados os autômatos Zi que representam os comportamentos contidos em Ri que sejam controláveis, no sentido de que a ação de controle respeite a restrição da existência de eventos controláveis e não controláveis em Gloci, e minimanente restritivos ao comportamento das plantas locais (Cassandras e Lafortune, 2008). A coluna "Zi" da Tabela 2 mostra os números de estados dos autômatos Zi.

Para que a supervisão modular local possa ser aplicada, os supervisores Zi, i=1...3, devem ser não conflitantes (Queiroz e Cury, 2002a). Para isso, faz-se o teste da *modularidade local*, que pode ser realizado calculando-se a composição síncrona Zc = $sync(Z1, Z2, Z3)$ e verificando se o autômato resultante Zc é *trim* (Queiroz e Cury, 2002a). Para o conjunto de supervisores calculado foi obtido como resultado um autômato Zc *trim* com 88 estados e 280 transições.

Por fim, calculam-se supervisores reduzidos empregando os métodos como em (Sivolella et al., 2006). A coluna "Zri" indica o número de estados de cada supervisor reduzido calculado. Mostram-se os supervisores reduzidos na Figura 4, com os respectivos mapas de desabilitação de eventos, que são os eventos a serem desabilitados na planta para cada estado do supervisor (Cassandras e Lafortune, 2008).

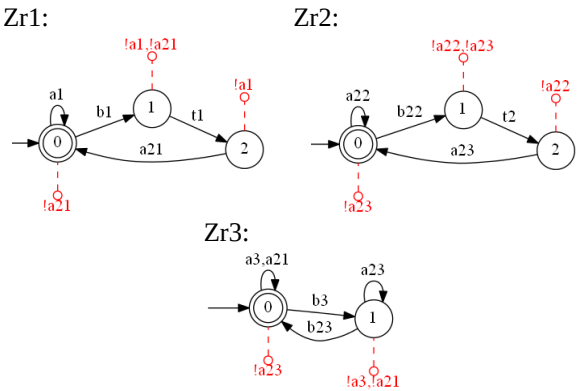

Figura 4 – Supervisores.

4 Validação e Avaliação de Desempenho

Nesta seção apresenta-se uma proposta de validação e avaliação de desempenho da lógica de controle sintetizada na seção 3 empregando-se a verificação de autômatos temporizados.

4.1. Proposta Geral

A figura 5 ilustra esquema geral proposto para validação e avaliação de desempenho, composto pela planta, que contém os modelos para os componentes da planta e as restrições operacionais, o supervisor, que contém os supervisores modulares locais sintetizados, e o monitor, que contém um conjunto de componentes empregados para gerar condições para a validação e a avaliação de desempenho.

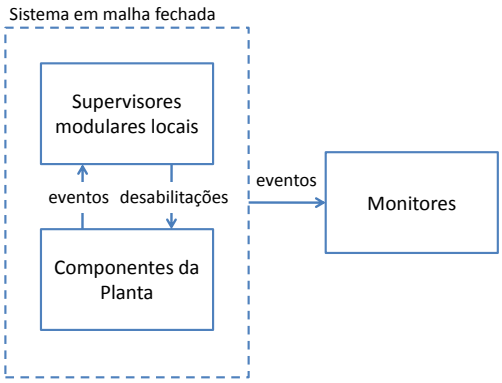

Figura 5 – Diagrama em blocos da planta, supervisor e monitor.

Utilizam-se os recursos da ferramenta computacional UPPAAL (Behrmann et al., 2006). O UPPAAL é um ambiente para modelagem, simulação e verificação de sistemas representados por uma rede de autômatos temporizados.

4.2. Componentes da Planta

A figura 6 ilustra os componentes da planta implementados na forma de uma rede de autômatos temporizados. Os componentes são construídos a partir do definido na Figura 2, com a inclusão no UPPAAL da temporização de eventos da Tabela 1.

Existem algumas diferenças entre os AT do UPPAAL e os autômatos ordinários, remete-se o leitor

para (Behrmann et al., 2006) para maiores detalhes. Além dos estados e das transições entre os estados, os AT possuem variáveis de relógio para contagem da passagem do tempo, por exemplo, a variável "q" em G1 (figura 6). Aos estados do AT são associados invariantes, que são condições lógicas para que estejam ativos, por exemplo, "q<=50" em s1 de G1. Às transições do AT são associados etiquetas, guardas e reinicializações. As etiquetas indicam a ocorrência dos eventos associados às transições, como "a1!" para a transição de s0 para s1 em G1. Os guardas são condições que habilitam a ocorrência das transições, como "q==50" para a transição de s1 para s0 em G1. As reinicializações são funções que são executadas com as transições do AT, e funcionam para atualizar o valor de variáveis com a ocorrência de transições, como "q:=0" na transição de s0 para s1 em G1. No AT o estado inicial é indicado por um círculo duplo e não são definidos estados marcados.

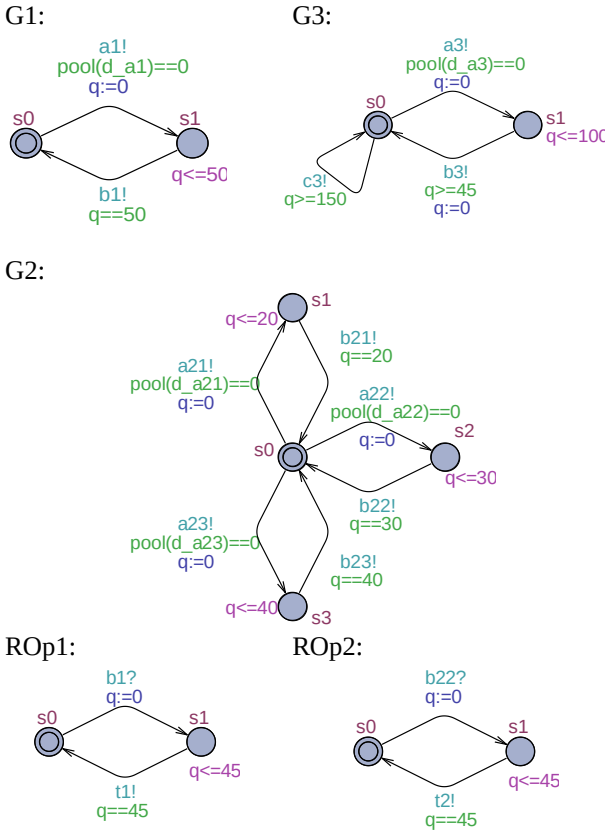

Figura 6 – Componentes da planta.

A sincronização de eventos, implementada no UPPAAL na forma de um canal de *broadcast*, indica que as transições com etiquetas e? são sincronizadas com a ocorrência de uma transição com etiqueta e! (Behrmann et al., 2006). Para implementar um esquema de sincronização na forma prevista na composição síncrona da seção 3, as etiquetas do tipo e! são definidas apenas para os componentes da planta, entendidos como geradores dos eventos, enquanto os eventos dos supervisores e monitores possuem a etiqueta e?, no sentido de seguirem os eventos gerados pelos componentes da planta. Se houver componentes da planta, ou restrições operacionais, com eventos sincronizados, como é o caso de G1 e ROp1, escolheu-se o componente da planta como gerador do evento.

Para cada evento controlável e define-se um vetor booleano d_e. Cada posição de d_e indica se um dado supervisor está habilitando ou desabilitando o evento e. O vetor d_e é indexado de 0 até NS-1, em que NS é o número de supervisores. Se o valor de d_e[i-1] é zero, com i = 1...NS, isso indica que o supervisor Zri está habilitando o evento e, caso d_e[i-1] seja um, Zri está desabilitando o evento e.

Um evento e está então habilitado quando o guarda da transição e! na planta, definido pela condição *pool(d_e)==0*, é verdadeiro. A função *pool(d_e)* faz o "OU" lógico de todos os elementos do vetor d_e. Assim, um evento e está habilitado na planta apenas se todos os supervisores "concordarem" que o mesmo deva estar habilitado num dado momento.

Além da temporização dos eventos, os componentes da planta poderiam incluir aspectos não levados em conta na modelagem para a síntese dos supervisores, como o comportamento das esteiras de entrada e saída ou, até mesmo, aspectos da implementação do sistema de controle, como as sequências operacionais previstas em (Queiroz e Cury, 2002b).

4.3. Supervisores Modulares Locais

A Figura 7 mostra os supervisores modulares locais da Figura 4 implementados na forma de uma rede de autômatos temporizados.

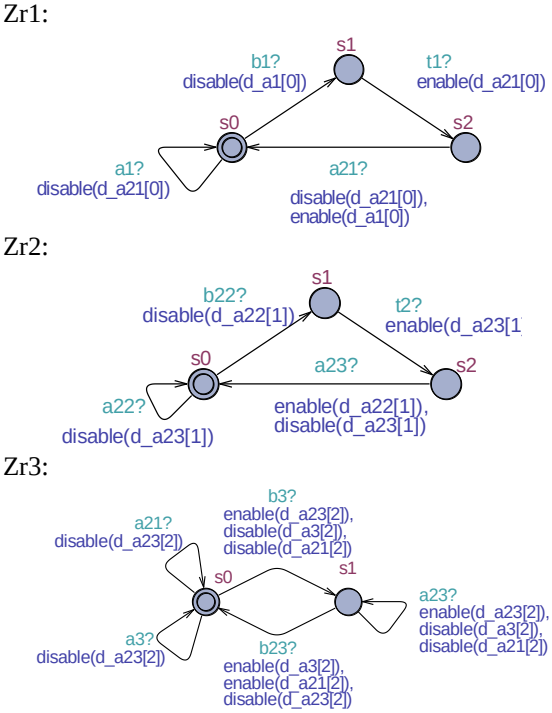

Figura 7 –Supervisores modulares locais.

No lugar dos mapas de desabilitação de eventos da Figura 4, a habilitação e a desabilitação dos eventos é feita pelas funções *enable()* e *disable()*, respectivamente, definidas nas reinicializações das transições dos supervisores, conforme na Figura 7. Para o supervisor Zri e o evento e, numa transição com *enable(d_e[i-1])*, o elemento d_e[i-1] torna-se 0 e,

numa transição com *disable(d_e[i-1])*, o elemento d_e[i-1] torna-se 1.

No lugar de serem implementados os supervisores modulares locais, poder-se-ia implementar o supervisor monolítico definido pelo autômato Zc mencionado na Seção 3, que representa o comportamento em malha fechada esperado do sistema. Entretanto os supervisores modulares locais permitem fácil substituição e inserção de novos componentes na arquitetura, além de serem de mais fácil implementação, pois possuem potencialmente menos estados.

4.4. Monitores e Especificações de Validação

Foi definido um conjunto de monitores associados a especificações em CTL (*Computation Tree Logic*), a linguagem de escrita das especificações para verificação no UPPAAL (Behrmann et al., 2006), que permitem o emprego da verificação formal para a validação do sistema composto pelos componentes da planta associados aos supervisores modulares locais.

Os monitores de validação são construídos com base nas especificações desejadas para o sistema, autômatos SPi definidos na Figura 3. A Figura 8 ilustra os monitores de desempenho M1 a M3, construídos com base nas especificações SP1 a SP3, respectivamente. Associadas aos monitores M1 a M3 são escritas as especificações CTL C1 a C3 da Figura 9, respectivamente. Remete-se o leitor a (Behrmann et al., 2006) para detalhamento da linguagem CTL da ferramenta UPPAAL.

Considere então o monitor M1 na Figura 8. O ciclo formado pelos estados s0, s1 e s2 de M1 correspondem ao comportamento expresso por SP1 na Figura 3. Os estados BAD1, BAD2 e BAD3 expressam condições indesejáveis, ou seja: que não se retire um vidro lateral sem que ele esteja lá (BAD2), que não se retire o lateral em B1 sem ter passado o tempo necessário para 'descanso' do *primer* (BAD1) e não se coloquem dois vidros laterais um em cima do outro (BAD3). A especificação CTL C1 na Figura 9, expressa se a condição indesejável definida pelos estados BAD1, BAD2 e BAD3 não ocorre para todos os traços de eventos que o sistema pode gerar. A verificação de C3 retornou que a mesma é verdadeira, confirmando o correto funcionamento do supervisor Zr1.

De forma análoga o monitor M2(M3) e a especificação CTL C2(C3) se referem à especificação SP2(SP3), e a verificação de C2(C3) retornou como verdadeira, confirmando o correto funcionamento do supervisor Zr2(Zr3).

O monitor M4 e a especificação CTL C4, Figuras 8 e 9, respectivamente, dizem respeito à especificação E7 da seção 3, isto é, o não *timeout* da linha de produção. O sistema de controle proposto não consegue tratar isso, pois embora os supervisores expressem todos os intertravamentos necessários, uma solução é não fazer nada com a planta no estado inicial. Dessa forma atingem-se os estados indesejáveis e a verificação retorna C4 como falsa.

Outras especificações de validação podem ser escritas, e as apresentadas a seguir não chegam a exaurir as possibilidades.

A especificação CTL C5 na Figura 9 serve para verificar se não existe um estado de *deadlock* no sistema, isto é, estado em que não haja possibilidade de evolução do modelo. Por outro lado, a especificação CTL C6 na Figura 9 é interpretada como a seguir: "é possível que a partir de qualquer estado possa-se chegar à configuração inicial dos modelos da planta e dos supervisores?" Como estes estados são marcados nos modelos da TCS, então a partir de qualquer estado chega-se ao estado marcado em malha fechada. Daí pode-se concluir que o sistema em malha fechada é não bloqueante. O resultado da verificação retornou que as especificações C5 e C6 são verdadeiras, o que leva a concluir que o sistema composto pelos componentes da planta associados aos supervisores não possui estados de *deadlock* ou estados bloqueantes.

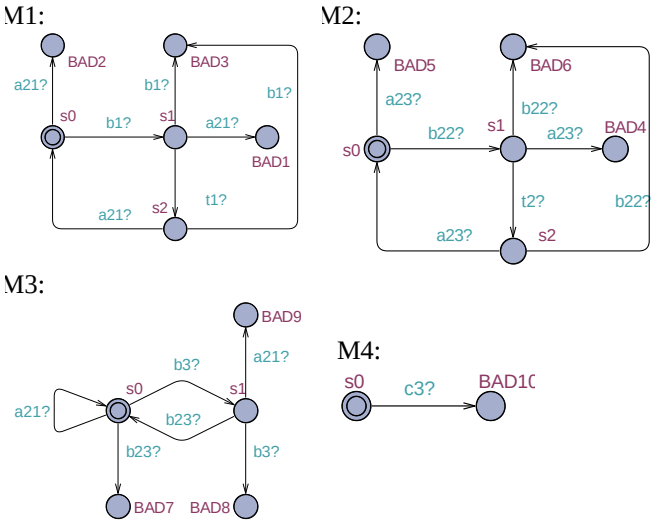

Figura 8 – Monitores de validação.

```
C1: A[] not (M1.BAD1 or M1.BAD2 or M1.BAD3)
C2: A[] not (M2.BAD4 or M2.BAD5 or M2.BAD6)
C3: A[] not (M3.BAD7 or M3.BAD8 or M3.BAD9)
C4: A[] not (M4.BAD10)
C5: A[] not deadlock
C6: A<> (G1.s0 and G2.s0 and G3.s0 and
ROp1.s0 and ROp2.s0 and Zr1.s0 and Zr2.s0
and Zr3.s0)
```

Figura 9 – Especificações CTL de validação.

Os procedimentos para validação do comportamento conjunto dos supervisores modulares com os componentes da planta empregando a verificação formal baseada nos monitores e nas especificações CTL são, de certa forma, gerais, pois baseiam-se nos comportamentos desejáveis levantados quando da síntese dos supervisores.

4.5. Monitores e Especificações de Desempenho

A verificação formal também pode ser empregada para a avaliação do desempenho do sistema em malha fechada em termos da conclusão das tarefas desejadas.

Como ilustração, considere o monitor MD1 da Figura 10. O monitor MD1 trata, essencialmente do tratamento de uma cabine, desde a autorização para a chegada da mesma (a3), passando pela sua chegada na posição A (b3), até a colocação do vidro para brisas (b23), em que atinge ao estado ONE.

O monitor MD1 é dotado com as seguintes variáveis auxiliares. A variável q é um relógio que conta o tempo entre a chegada da primeira cabine na posição A e a colocação do vidro para brisas pelo robô RWS na mesma cabine. As variáveis n1, n2, n3 e n4 são contadores para o número de vidros laterais tratados nas seguintes situações. São limitadas ao valor NL para garantir a convergência na verificação. As variáveis n1/n3 contam o número de vidros laterais que são processados pelo robô RSG (aplicação de *primer* e depósito em B1) antes/após a chegada da primeira cabine. As variáveis n2/n4 contam o número de vidros laterais que são processados pelo robô RWS (aplicação de cola e liberação) antes/após a chegada da primeira cabine.

As especificações CTL C7 a C10 na Figura 11 dizem respeito à avaliação de propriedades expressas em MD1.

A especificação C7 destina-se a avaliar o tempo mínimo de colocação de um vidro lateral numa cabine. Literalmente expressa a questão: "Existe caminho no qual o estado ONE seja verdadeiro e q seja menor que o valor especificado numericamente em Qmax?" Na verificação, começa-se com valor alto para Qmax, por exemplo, 50, e diminui-se até que a especificação torne-se falsa, o que ocorreu na passagem de Qmax de 40 para 39. Assim 40s é o tempo mínimo para colocação de vidro para brisas numa cabine.

As especificações C8, C9, C10 e C11 tratam do número de vidros laterais processados pelo sistema nas situações caracterizadas pelos contadores n1, n2, n3 e n4, respectivamente. De forma geral, testam se existe no sistema um estado em que ONE seja válido e ni seja igual a um determinado valor Nimax, com i=1...4. A sistemática para verificação destas especificações foi iniciar Nimax (i=1...4) com um valor baixo e elevá-lo até que a especificação respectiva torne-se falsa. Com a verificação, os valores máximos para os quais as especificações são verdadeiras foram N1max=2, N2max=2, N3max=1 e N4max=1. Assim, conclui-se que o robô RSG pode processar até 2 vidros laterais após o comando para entrar uma cabine; o robô RWS pode processar até 2 vidros laterais após o comando para entrar uma cabine; o robô RSG pode processar até 1 vidro lateral após a chegada da primeira cabine; e o robô RWS pode processar até 1 vidro lateral após a chegada da primeira cabine.

Os procedimentos mencionados nesta seção podem ser expandidos, por exemplo, para estimar o tempo mínimo para o processamento dos vidros laterais com a criação de novos monitores e especificações de forma semelhante ao feito.

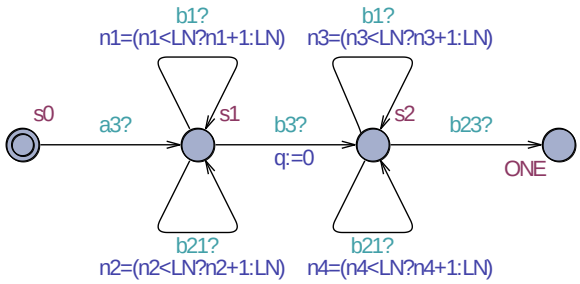

Figura 10 – Monitor de desempenho MD1.

```
C7:  E<> (MD1.ONE and MD1.q<=Qmax)
C8:  E<> (MD1.ONE and MD1.n1==N1max)
C9:  E<> (MD1.ONE and MD1.n2==N2max)
C10: E<> (MD1.ONE and MD1.n3==N3max)
C11: E<> (MD1.ONE and MD1.n4==N4max)
```

Figura 11 – Especificações CTL para o desempenho.

5 Conclusões

Este trabalho coloca-se como uma alternativa para teste e validação da lógica de controle gerada pela TCS antes da implementação propriamente dita no controle do sistema físico, contribuindo assim para a confiabilidade e o desempenho do projeto do sistema de controle.

O detalhamento do sistema empregado na verificação limitou-se à explicitação do tempo de atraso das transições. O método pode ser expandido para um detalhamento que inclua mais componentes, incluindo aspectos de implementação em hardware.

Estão em desenvolvimento supervisores calculados com base nas abordagens de síntese de SEDT de Brandin e Wonham (1993) e de Cassez et al. (2005), para comparação com a lógica de controle sintetizada neste trabalho empregando-se o método proposto.

Referências Bibliográficas

Alur, R. e Dill, D.L. A Theory of Timed Automata. Theoretical Computer Science 126, p. 183-235, 1994.

Behrmann, G., David, A. e Larsen, K. G. A Tutorial on UPPAAL 4.0. Department of Computer Science, Aalborg University, Denmark, 2006.

Brandin, B.A. e Wonham W.M. Modular Supervisory Control of Timed Discrete-Event Systems. Proceedings of the 32nd Conference on Decision and Controle. San Antonio, Texas, December, 1993.

Cassandras, C. G. e Lafortune, S. Introduction to Discrete Event Systems, 2a ed., Kluwer Academic Publishers, Massachussetts, 2008.

Cassez, F., David, A., Fleury, E., Larsen, K. G. e Lime, D. Efficient on-the-fly algorithms for the analysis of timed games. Proceedings of the 16th Conf. on Concurrency Theory (CONCUR'05), volume 3653 of Lecture Notes in Computer Science, pages 66-80. Springer, 2005.

Queiroz, M. H. e Cury, J. E. R. Controle Supervisório Modular de Sistemas de Manufatura. Controle & Automação, v. 13, n.2, p. 115-125, 2002a.

Queiroz, M. H. e Cury, J. E. R. Synthesis and Implementation of Local Modular Supervisory Control for a Manufacturing Cell. In: 6th International Workshop on Discrete Event Systems (WODES), v. 1. p. 377-382, 2002b.

Sivolella, L.F., Cunha, A.E.C. e Ades, R. Redução de Supervisores como Ferramenta para a Implementação de Supervisores em Controladores Discretos. Anais do Congresso Brasileiro de Automática, Salvador, 2006.