VERIFICAÇÃO FORMAL DE ESTRATÉGIA DE CONTROLE DE INTERSEÇÃO COM VEÍCULOS AUTOMATIZADOS

Joana Alves dos Santos
Fábio Luis Baldissera
Rodrigo Castelan Carlson
Universidade Federal de Santa Catarina
Departamento de Automação e Sistemas

RESUMO
Com o rápido desenvolvimento de veículos automatizados, há uma necessidade crescente de coordenar a interação deles no tráfego. A segurança é um fator chave na coordenação de veículos automatizados, portanto, garantí-la por meio de restrições rígidas é de extrema importância. Utilizou-se de verificação formal para confirmar a segurança de um modelo simplificado de uma estratégia de controle de tráfego que coordena veículos automatizados em interseções. Por um erro de projeto, a estratégia não garante a segurança em algumas condições específicas, resultando em colisões traseiras. Uma solução proposta para o problema descrito também é verificada e mostra-se que o resolve corretamente. A modelagem é realizada com a linguagem Fiacre e a verificação é realizada com o verificador de modelos selt.

ABSTRACT
With the fast paced development of automated vehicles, there is an increased need of coordinating how they will behave in traffic. Safety is a key factor in the coordination of automated vehicles, therefore guaranteeing it through hard constraints is of utmost importance. In this paper the use of formal verification is demonstrated for a simplified model of a traffic control strategy that coordinates automated vehicles at intersections. As a result of a design error, the strategy fails to guarantee safety at some specific conditions, resulting in rear end collisions. A solution proposed for the reported problem is also verified and shown to correctly solve it. The modelling is performed with the Fiacre language and the verification is performed with the selt model-checker.

1. INTRODUÇÃO

O desenvolvimento de veículos automatizados vem experimentando intenso crescimento. Porém, limitações de ordem legal, operacional e tecnológica ainda impedem a substituição efetiva em larga escala de veículos operados manualmente por veículos automatizados. Do ponto de vista operacional, a existência de zonas de conflito, como interseções urbanas e entroncamentos rodoviários, requer o desenvolvimento de estratégias de controle de tráfego nessas regiões para coordenação dos veículos. Um grande número de trabalhos de coordenação do tráfego automatizado vêm sendo proposto com múltiplas abordagens (Chen e Englund, 2016; Rios-Torres e Malikopoulos, 2017).

Preocupa, entretanto o pouco rigor na verificação de muitos dos sistemas elaborados até agora, ainda que a verificação seja uma das bases do ciclo de vida de desenvolvimento de sistemas. Shladover (2018) destaca que não há tecnologia que suporte o projeto, o desenvolvimento, a verificação e a validação de software para sistemas complexos como veículos automatizados. Muito embora, o foco dele esteja no veículo e no seu software, a mesma ideia se aplica às estratégias de controle de tráfego hierarquicamente superiores aos veículos, tanto quando trata-se do controle de uma interseção, como quando trata-se de uma rede viária envolvendo várias interseções.

De fato, é pré-requisito crucial para este tipo de sistema que esteja correto, pois as perdas em sistemas envolvendo seres humanos são de valor inestimável. Mesmo sistemas bem projetados

e rigorosamente testados, podem conter erros sutis que causem danos de grandes proporções. Assim, a taxa de erros nesses sistemas deve ser suficientemente baixa para que seja pelo menos tão seguro quanto o caso envolvendo apenas motoristas humanos (Shladover, 2018). Isso passa essencialmente pelo uso de métodos formais de verificação e validação desses sistemas.

Técnicas de verificação formal têm sido muito bem sucedidas em muitos áreas, como eletrônica, informática e aviação (Clarke *et al.*, 1999). Garantir o funcionamento correto de sistemas físicos complexos, como os de tráfego automatizado, é um dos mais desafiadores e importantes problemas em ciência da computação, matemática e engenharia (Platzer, 2010). Apesar disso, aplicações em sistemas de automação do tráfego ainda são incipientes (Stursberg *et al.*, 2004; Ölveczky e Meseguer, 2010; Loos e Platzer, 2011).

Neste trabalho, é demonstrada a aplicação de verificação formal, pela técnica de verificação de modelos, a uma estratégia de coordenação de veículos automatizados em uma interseção proposta por Zhang *et al.* (2017). Essa estratégia propõe uma heurística que aloca tempos para os veículos cruzarem a interseção combinada com uma lei de controle ótimo que define perfis de aceleração para os veículos nas aproximações da interseção para garantir que não haja colisões. Entretanto, Furtado (2017) mostrou que uma restrição de colisão traseira é violada em alguns casos e propôs uma solução, mas sem demonstrar formalmente que sempre funciona. É interessante notar que o erro foi encontrado ao acaso em simulação microscópica, quando demandas muito altas de tráfego foram usadas, e ainda assim ocorriam raramente. Esse exemplo reforça que testes e simulações não são capazes de explorar todos os estados possíveis.

Uma modelagem simplificada da heurística para apenas uma aproximação é realizada com a linguagem Fiacre (Berthomieu *et al.*, 2007). O modelo resultante é verificado com o verificador de modelos selt (Topcased, 2012). O resultado mostra que mesmo por meio de um modelo simples, o erro poderia ter sido encontrado com verificação formal e mostra que a solução proposta garante efetivamente que não há colisão traseira para todos os estados possíveis.

Na Seção 2 é feita uma breve apresentação sobre verificação formal, Fiacre e selt. A estratégia de Zhang *et al.* (2017) e a correção proposta por Furtado (2017) são apresentadas na Seção 3 e modeladas e verificadas na Seção 4. As conclusões do trabalho são apresentadas na Seção 5.

2. VERIFICAÇÃO FORMAL

Verificação formal consiste em verificar se um modelo matemático satisfaz propriedades especificadas formalmente (por meio de sentenças lógicas, por exemplo). Ela é empregada para avaliar a correção de um determinado sistema, representado por um modelo matemático, em relação às especificações de projeto (Baier e Katoen, 2008).

Técnicas de verificação formal apresentam vantagens em relação a outras formas de verificação de correção de sistemas, como revisão por pares ou simulação, por exemplo. Enquanto a simulação permite analisar apenas uma parcela das possíveis trajetórias de um dado sistema, não podendo, portanto, oferecer ao projetista garantias de correção, as técnicas de verificação formal são exaustivas. Assim, se um algoritmo de verificação formal retorna que um modelo satisfaz uma propriedade, então pode-se garantir que não há nenhum comportamento desse modelo que não satisfaça a propriedade em questão.

Embora haja vantagens no uso de técnicas de verificação formal, tais técnicas não substituem os testes com sistemas reais, já que a verificação formal dá garantias somente de que *o modelo* do sistema (e não o sistema em si) satisfaz determinadas propriedades.

2.1. Abordagens de verificação formal

Duas abordagens comumente empregadas para verificação formal são a verificação de modelos (*model-checking*) e a prova automática de teoremas (*automated theorem proving*).

Em verificação de modelos (Clarke *et al.*, 1999; Baier e Katoen, 2008), o modelo é dado na forma de um sistema de transição (basicamente, estados discretos e transições entre eles) e a propriedade a ser verificada é expressa por meio de uma lógica temporal (por exemplo, Lógica Temporal Linear). A verificação de uma propriedade é feita automaticamente e exaustivamente a partir da exploração dos estados alcançáveis do sistema. Quando uma propriedade não é verificada, a ferramenta de verificação fornece um contraexemplo. A apresentação de um contraexemplo auxilia o projetista a identificar o erro de projeto e a corrigí-lo.

A prova automática de teorema (Clarke e Wing, 1996; Platzer, 2010) é uma técnica que também parte de representações formais do sistema e da propriedade a ser verificada. No entanto, por meia desta técnica verifica-se se um sistema satisfaz uma dada propriedade não pela exploração dos estados alcançáveis (como o faz verificação de modelos), mas pela obtenção de uma prova matemática de que a propriedade a ser verificada é, para o modelo do sistema, uma consequência lógica de um conjunto de axiomas e hipóteses.

2.2. Verificação de modelos com Fiacre e Selt

Fiacre é um acrônimo para *Format Intermédiaire pour les Architectures de Composants Répartis Embarqués* (Formato Intermediário para as Arquiteturas de Componentes Distribuídos Embarcados). É uma linguagem formal intermediária para representar os aspectos comportamentais e temporais de sistemas, particularmente de sistemas concorrentes. A linguagem Fiacre foi criada pelo projeto TOPCASED (Farail *et al.*, 2006) para servir como um formato intermediário entre linguagens de alto nível de descrição e ferramentas de verificação. O uso de uma linguagem formal de modelagem intermediária tem dois benefícios (Berthomieu *et al.*, 2007): (i) ajuda a reduzir a diferença semântica entre modelos de alto nível e o formato de entrada das ferramentas de verificação; (ii) possibilita definir com precisão a semântica da linguagem de entrada e compartilhar este trabalho entre diferentes ferramentas de verificação. O selt (Topcased, 2012) é uma ferramenta de verificação de modelos que pode ser empregada com o Fiacre.

2.3. Verificação de estratégias de controle de tráfego

A verificação de segurança para sistemas de controle de tráfego com veículos automatizados é essencial porém é, ainda, pouco aplicada. A estratégia proposta por Stursberg *et al.* (2004) aplica verificação guiada por contraexemplos, baseada em verificação de modelos, a um sistema de controle de cruzeiro com dois carros. Em Ölveczky e Meseguer (2010), foi feito um extenso estudo de caso real baseado na indústria envolvendo sistemas embarcados distribuídos com uma interseção de tráfego com quatro aproximações. Os veículos se comunicam por mensagens assíncronas sem um controlador central. Os requisitos de segurança e vivacidade do sistema real foram formalmente verificados usando a ferramenta *Real-Time Maude* e seus recursos de verificação de modelos. Loos e Platzer (2011) relataram uma abordagem lógica para verificação

de sistemas híbridos que captura as decisões do controle discreto e a dinâmica contínua dos veículos. O modelo apresentado foi de uma interseção de duas vias com semáforo e dois carros. O sistema foi formalmente modelado com lógica diferencial dinâmica e verificado com a ferramenta *KeYmaera* como um sistema livre de colisões e os veículos nunca desobedecem o sinal vermelho.

3. ESTRATÉGIA DE CONTROLE DE INTERSEÇÕES AUTOMATIZADAS

Nesta seção é descrita a estratégia de controle proposta por Zhang *et al.* (2017). A estratégia consiste de um modelo, uma heurística que aloca tempos para os veículos cruzarem a interseção e um problema de controle ótimo que determina o perfil de aceleração do veículo na aproximação à interseção. Para uma discussão sobre formulações alternativas, ver Furtado (2017).

3.1. Modelo

Considere uma interseção como a da Figura 1 com uma zona de conflito (ZN), definida por um quadrado de lado S, e uma zona de controle (ZC), que se estende por uma distância L a montante da ZN em cada aproximação. A interseção possui um coordenador que pode se comunicar com os veículos dentro da ZC. Troca de faixa e movimentos de conversão não são permitidos.

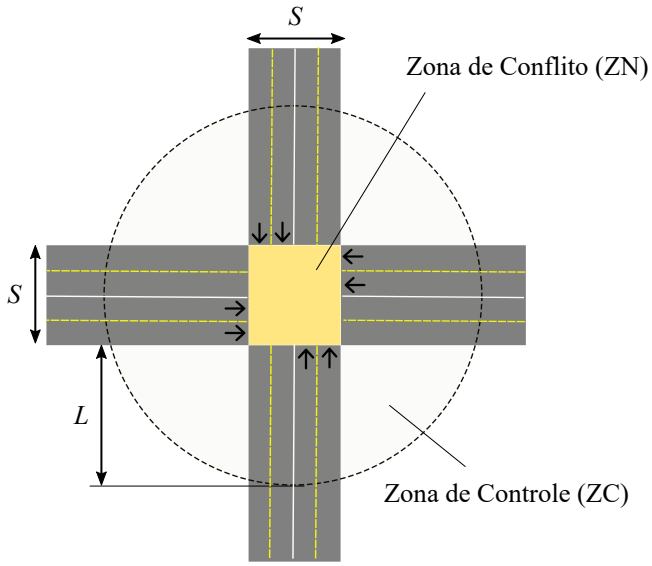

Figura 1: Representação de uma interseção modelada como descrito em Zhang *et al.* (2017).

Os veículos dentro da ZC recebem um número de identificação sequencial i por ordem de chegada. Cada veículo i move-se longitudinalmente em uma faixa especificada e tem sua dinâmica descrita por

$$\dot{p}_i = v_i(t), \quad p_i(t_i^0) = p_i^0, \quad \dot{v}_i = u_i(t), \quad v_i(t_i^0) = v_i^0 \tag{1}$$

em que $p_i(t)$, $v_i(t)$, e $u_i(t)$ são a posição, a velocidade e a aceleração do veículo i dentro da ZC no tempo t, respectivamente, e t_i^0 é o instante de entrada do veículo i na ZC. As condições iniciais p_i^0 e v_i^0 são conhecidas. A entrada de controle $u_i(t)$ e os valores de velocidade são restritos a:

$$u_{\min} \leq u_i(t) \leq u_{\max}, \quad 0 \leq v_{\min} \leq v_i(t) \leq v_{\max}, \quad \forall t \in [t_i^0, t_i^{\mathrm{m}}] \tag{2}$$

em que $u_{\min}$ e $u_{\max}$, e $v_{\min}$ e $v_{\max}$ são os valores mínimos e máximos de aceleração e velocidade, respectivamente. O instante de entrada do veículo i na ZN é t_i^{m}.

A restrição de segurança para que não ocorra colisão traseira é dada por:

$$s_i(t) = p_k(t) - p_i(t) \geq \delta, \quad \forall t \in [t_i^0, t_i^{\mathrm{m}}] \tag{3}$$

em que $s_i(t)$ é a distância entre o veículo i e o veículo k que está imediatamente à frente do veículo i na mesma faixa, e a distância mínima δ é maior do que o comprimento de um veículo.

Assume-se ainda que todos os veículos são instrumentados e as medições ocorrem sem erros ou atrasos; que para o veículo i, as restrições (1) e (2) estão inativas em t_i^0; e que a velocidade dos veículos dentro da ZN é constante, isto é, $v_i(t) = v_i(t_i^{\mathrm{m}}) = v_i(t_i^{\mathrm{f}}), \forall t \in [t_i^{\mathrm{m}}, t_i^{\mathrm{f}}]$, com t_i^{f} o instante de tempo de saída da ZN. Isto implica que:

$$t_i^{\mathrm{f}} = t_i^{\mathrm{m}} + \frac{S}{v_i(t_i^{\mathrm{m}})}. \tag{4}$$

3.2. Heurística

São definidos para cada veículo i quatro conjuntos de veículos da ZC: (i) $\mathcal{R}_i(t)$ contém todos os veículos que trafegam na mesma aproximação que o veículo i, mas em faixas diferentes; (ii) $\mathcal{L}_i(t)$ contém todos os veículos que trafegam na mesma aproximação e faixa que o veículo i; (iii) $C_i(t)$ contém todos os veículos que trafegam em aproximações diferentes do veículo i e que têm destinos que podem causar uma colisão com o veículo i na ZN; e (iv) $O_i(t)$ contém todos os veículos trafegando na aproximação oposta e sentido contrário ao do veículo i, tal que não há possibilidade de colisão com o mesmo na ZN.

Esses conjuntos são a base para a determinação de tempos de chegada dos veículos para que não ocorram colisões. As seguintes regras heurísticas são empregadas:

$$t_i^{\mathrm{f}} = \begin{cases} t_1^{\mathrm{f}}, \text{ se } i = 1, & \text{(5a)} \\ \max\{t_{i-1}^{\mathrm{f}}, t_i^{\mathrm{c}}\}, \text{ se } i-1 \in \mathcal{R}_i(t) \cap O_i(t), & \text{(5b)} \\ \max\left\{t_{i-1}^{\mathrm{f}} + \dfrac{\delta}{v_i(t_{i-1}^{\mathrm{f}})}, t_i^{\mathrm{c}}\right\}, \text{ se } i-1 \in \mathcal{L}_i(t), & \text{(5c)} \\ \max\left\{t_{i-1}^{\mathrm{f}} + \dfrac{S}{v_i(t_{i-1}^{\mathrm{f}})}, t_i^{\mathrm{c}}\right\}, \text{ se } i-1 \in C_i(t). & \text{(5d)} \end{cases}$$

O primeiro veículo que entra na ZC não sofre restrições e t_1^{f} pode ser obtido definindo-se $v_1(t) = v_1(t_1^0)$ para todo $t \in [t_1^0, t_1^{\mathrm{f}}]$ ($u_1(t) = 0$). O limite inferior fisicamente possível para cada veículo i é definido por t_i^{c}, e depende de o veículo poder alcançar $v_{\max}$ antes de t_{i-1}^{m} ou não:

- Se o veículo i entra a ZC em t_i^0, acelera com $u_{\max}$ até $v_{\max}$ e então trafega nessa velocidade até deixar a ZN no instante de tempo t_i^1, então

$$t_i^1 = t_i^0 + \frac{L+S}{v_{\max}} + \frac{(v_{\max} - v_i^0)^2}{2u_{\max}v_{\max}}. \tag{6}$$

- Se o veículo i acelera com $u_{\max}$, mas alcança a ZN no instante t_i^{m} com velocidade $v_i(t_i^{\mathrm{m}}) < v_{\max}$, então

$$\begin{aligned} t_i^2 &= t_i^0 + \frac{v_i(t_i^{\mathrm{m}}) - v_i^0}{u_{\max}} + \frac{S}{v_i(t_i^{\mathrm{m}})}, \\ v_i(t_i^{\mathrm{m}}) &= \sqrt{2Lu_{\max} + (v_i^0)^2}. \end{aligned} \tag{7}$$

Assim, $t_i^c = \max\{t_i^1, t_i^2\}$. Uma vez que t_i^f esteja disponível por meio da Equação 5, t_i^m pode ser calculado pela Equação 4. O par $[t_i^m, t_i^f]$ fornece uma janela de tempo para o veículo cruzar a ZN e é calculado no instante de tempo em que o veículo entra na ZC. Para que isso seja possível, uma estrutura de comunicação é estabelecida:

$$Y_i(t) \triangleq \left\{p_i(t), v_i(t), \mathcal{Q}_i, s_i(t), t_i^m\right\}, \forall t \in [t_i^0, t_i^m] \tag{8}$$

em que $\mathcal{Q}_i \in \{\mathcal{R}_i, \mathcal{L}_i, \mathcal{O}_i, \mathcal{C}_i\}$ é um subconjunto atribuído ao veículo i pelo coordenador. O veículo k também é atribuído pelo coordenador e, baseado nisso, $s_i(t)$ é calculado, enquanto t_i^m é avaliado baseado na informação recebida do veículo $i-1$. Toda informação em $Y_i(t)$, $Y_{i-1}(t)$ e $Y_k(t)$ é considerada disponível assim que o veículo i entre na ZC, e $p_k(t)$, $p_i(t)$ e $v_i(t)$ são obtidos dos sensores dos veículos.

3.3. O problema de controle ótimo

Com a janela de tempo $[t_i^m, t_i^f]$ disponível, o perfil de aceleração do veículo i para que alcance a ZN no tempo t_i^m pode ser calculado por meio de um problema de controle ótimo:

$$\begin{gathered}\min_{u_i} \frac{1}{2}\int_{t_i^0}^{t_i^m} u_i^2(t)dt, \\ \text{sujeito a (1), (2), (4), (5), } p_i(t_i^0) = 0, p_i(t_i^m) = L, \text{ dados } t_i^0, v_i(t_i^0).\end{gathered} \tag{9}$$

3.4. Discussão e contraexemplo

A restrição para evitar colisão traseira dada pela Equação 3 não é incluída na formulação pois entende-se que ela é assegurada ao longo de $[t_i^0, t_i^m]$ pela solução do Problema 9 (Zhang *et al.*, 2017). Porém, esta formulação viola sim a Equação 3 em alguns casos.

A estratégia descrita tem como uma de suas alegadas vantagens a necessidade de depender somente das informações do último veículo que entrou na ZC, $Y_{i-1}(t)$, embora $Y_k(t)$ seja usada para o cálculo de $s_i(t)$. Análise e simulações da estratégia revelam que sequências específicas de veículos não conflitantes entrando na ZC podem levar à alocação da mesma janela de tempo para dois veículos seguidos na mesma faixa, porque o veículo k imediatamente à frente do veículo i quando $k \neq i-1$ não é considerado. Nesse cenário, como a alocação de tempo pela Equação 5 não é dependente das variáveis de decisão do Problema 9, a solução desse último resultaria em uma sequência de controle $u_i(t)$ que causa colisão. Há também outras formulações para o Problema 9 que de alguma maneira incluem a restrição de colisão traseira. Nesses casos, o problema se tornaria infactível, ainda que pudesse ser factível com a alocação apropriada das janelas de tempo. Veja Furtado (2017) para uma discussão detalhada.

Um contraexemplo é apresentado a seguir para demonstrar a violação da Equação 3. Foi considerada uma interseção e veículos com os parâmetros na Tabela 1, e uma sequência de chegadas com quatro veículos que causa conflito. Furtado (2017) apresenta um exemplo com mais veículos. Um esquemático da interseção em um determinado instante de tempo é apresentado na Figura 2. A sequência é a seguinte (ver também a Tabela 2):

- O veículo #1 é o primeiro a chegar, entra na ZC pela faixa 1 em $t_1^0 = 11$ s, e tem sua velocidade mantida. De acordo com as Equações 5a e 4, $t_1^m = 27{,}16$ s e $t_1^f = 28{,}24$ s.
- O veículo #2 entra depois do veículo #1 em $t_2^0 = 12{,}50$ s, pela faixa 0. Resulta das Equações 5d e 4 que $t_2^m = 29{,}32$ s a $t_2^f = 30{,}40$ s.
- O veículo #3 chega em $t_3^0 = 12{,}55$ s pouco depois do veículo #2 e na mesma faixa que o

veículo #1. Da Equação 5b resulta que $t_3^{\mathrm{f}} = 30{,}40$ s e da Equação 4 que $t_3^{\mathrm{m}} = 29{,}32$ s.

- Finalmente, o veículo #4 chega em $t_4^0 = 14$ s depois do veículo #3 e na mesma faixa que o veículo #2. Da Equação 5b resulta $t_4^{\mathrm{f}} = 30{,}40$ s e da Equação 4, $t_4^{\mathrm{m}} = 29{,}32$ s, os mesmos valores dos veículos #2 e #3.

Tabela 1: Parâmetros da interseção e dos veículos.

Interseção			Veículo			
L (m)	S (m)	δ (m)	$v_{\min}$ (m/s)	$v_{\max}$ (m/s)	$u_{\min}$ (m/s^2)	$u_{\max}$ (m/s^2)
250	18	10	2,78	16,67	−4	4

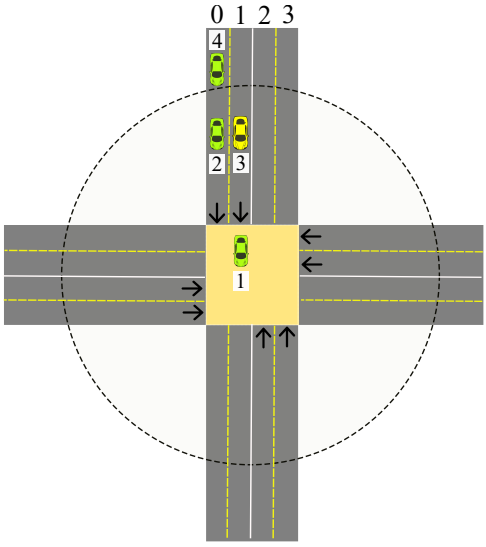

Figura 2: Esquemático da sequência usada para o contraexemplo.

Tabela 2: Sequência de chegadas do contraexemplo.

#	t_i^0 (s)	$v_i(t_i^0)$ (m/s)	k	t_i^{f} (s)	$t_i^{\mathrm{f,feas}}$ (s)	$v_i(t_i^{\mathrm{f}})$ (m/s)	faixa	$i-1 \in Q_i$
1	11,00	16,67	-	28,24	28,24	16,67	1	–
2	12,50	16,67	-	30,40	30,40	16,67	0	C_5
3	12,55	16,67	3	30,40	30,40	16,67	1	$\mathcal{R}_6$
4	14,00	16,67	2	30,40	31,00	16,67	0	$\mathcal{R}_7$

A sequência apresentada leva à alocação das mesmas janelas de tempo para três veículos. Assim, os veículos #2 e #4 na faixa 0 atingiriam o mesmo espaço físico no mesmo instante de tempo, isto é, colidiriam. Isso acontece pois t_k^{f} não foi considerado na Equação 5.

Se três ou mais veículos em trajetórias não conflitantes entram na ZC sequencialmente, e dois desses veículos estão na mesma faixa mas não são adjacentes na ordem de chegada, o esquema de alocação da Equação 5 falha em garantir um ambiente livre de colisão. O mesmo problema aconteceria se o veículo #3 estivesse trafegando na aproximação oposta, nas faixas 2 ou 3.

3.5. Solução

A solução proposta para resolver este problema (Furtado, 2017) consiste em forçar explicitamente que o veículo i mantenha uma distância mínima δ do veículo k imediatamente à frente dele. Isto requer informações adicionais do veículo k que, como visto, já estão disponíveis, apenas foram

negligenciadas. Uma janela de tempo factível e segura pode ser obtida depois do cálculo de t_i^{f} pela Equação 5 e sua substituição por $t_i^{\mathrm{f,feas}}$ como:

$$t_i^{\mathrm{f,feas}} = \max\left\{t_i^{\mathrm{f}}, t_k^{\mathrm{f}} + \frac{\delta}{v_k(t_k^{\mathrm{f}})}\right\}. \tag{10}$$

A aplicação desta solução ao cenário do contraexemplo é apresentada na coluna $t_i^{\mathrm{f,feas}}$ da Tabela 2. A alocação permanece a mesma para os veículos #1, #2, e #3. Já o veículo #4, ao invés de ter alocado $t_4^{\mathrm{f}} = 30{,}40$ s pela Equação 5, recebe $t_4^{\mathrm{f,feas}} = 31{,}00$ s após imposição da nova restrição de limite inferior pela Equação 10. Uma solução semelhante à de Furtado (2017) foi apresentada de forma independente por Malikopoulos *et al.* (2018).

4. VERIFICAÇÃO DA ESTRATÉGIA

Nesta seção, a estratégia da Seção 3 é modelada por meio de uma abstração discreta que elimina variáveis contínuas, sem no entanto alterar a validade das propriedades que se deseja verificar. Por limitações de espaço, apenas a heurística, de onde advém o erro discutido na Seção 3.4 é modelada. O modelo é simplificado e considera apenas uma aproximação à interseção. Além de ser suficiente para demonstrar o problema em discussão, é uma etapa natural no processo de elaboração de um modelo completo (Lygeros *et al.*, 1998). Assim, apenas será verificado se o sistema sem e com correção aloca ou não janelas de tempo que provocariam colisão traseira na entrada da ZN.

Com um modelo mais completo, outras propriedades poderiam ser verificadas, como propriedades discutidas por Zhang *et al.* (2017) que garantem que o Problema 9 é factível. Também poderiam ser verificadas propriedades gerais de sistemas, como ocorrência de bloqueios ou de postergação indefinida ou crescimento ilimitado de filas. Com a abstração e ferramenta adequadas, a verificação do sistema envolvendo o Problema 9 também pode ser realizada, por exemplo, para garantir que dois veículos com janelas de tempo diferentes não colidem durante a aproximação (considere o caso em que as soluções ótimas fazem com que um veículo muito lento demore a acelerar e um muito rápido logo atrás demore a reduzir sua velocidade).

4.1. Modelagem em Fiacre

O modelo elaborado em Fiacre representa a alocação de intervalos de tempo para os veículos cruzarem a ZN. Como a velocidade é constante dentro da ZN e o instante de entrada é determinado pela Equação 4 a partir de t_i^{f}, basta modelar este último. A Listagem 1 apresenta a descrição do modelo em Fiacre. Exceto por alguns comandos específicos da linguagem Fiacre (linhas 5, 6, 12 e 27), o restante é muito semelhante a linguagens de programação de alto nível como Pascal ou C. Comentários na listagem esclarecem detalhadamente a modelagem.

Uma fila é usada para representar uma aproximação com duas faixas e cada veículo que chega é incluído na fila. A posição de um veículo na fila é sua identificação i. Cada veículo é representado por um *array* com dois valores: um valor indica a faixa onde o veículo se encontra e o outro indica o valor escolhido para t_i^{f}. Um novo veículo é gerado em qualquer faixa. Notar que valores no conjunto dos naturais são usados para t_i^{f}, pois para efeitos desta modelagem basta verificar se os valores de t_i^{f} são diferentes e não o valor preciso contínuo calculado. Um variável é usada para monitorar o tamanho da fila e outra é usada para manter registro do último veículo a entrar na ZC.

Listagem 1: Código fonte em FIACRE da estratégia discretizada antes da correção.

```
const NUM_MAX_VEI : nat is 5              /* Número máximo de veículos na ZC.    */
type Veiculo is array 2 of nat            /* Armazena infos de um veículo na ZC.*/
type Via is queue NUM_MAX_VEI of Veiculo  /* Via é uma fila composta por         */
                                          /* veículos na ZC.                     */
process AIC is                            /* Processo e estados do autômato      */
  states operacao, finalizado             /* estendido que modela o sistema.     */
  var via     : Via  := {||},             /* Inicia como uma fila vazia.         */
      tam     : 0..5 := 0,                /* Tamanho da fila.                    */
      faixa   : 0..1 := 0,                /* Faixa 0 ou 1.                       */
      tf      : nat  := 0,                /* Instante de tempo de saída da ZN.   */
      ult_vei : array 2 of nat := [0,0]   /* Último veículo que entrou na ZC.    */
    from operacao select
    on not (full(via));                 /* Se a ZC não tiver NUM_MAX_VEI, um novo  */
      faixa := any;                     /* veículo pode entrar em qualquer faixa.  */
      if empty(via) then                /* Se a via estiver vazia, o primeiro      */
        tf  := 1                        /* veículo recebe instante de tempo  1.    */
      else
        if faixa = ult_vei[0] then      /* Se o novo veículo chegou na mesma faixa */
          tf  := ult_vei[1] + 1         /* que o último, o instante de tempo do    */
        else                            /* novo veículo tem que ser maior que o do */
          tf  := ult_vei[1]             /* último. Caso contrário, o tempo do novo */
        end                             /* veículo poderá ser igual ao do anterior.*/
      end;
      via := enqueue(via,[l, tf]);      /* Adiciona novo veículo à fila.           */
      tam := tam + 1;                   /* Incrementa tamanho da fila.             */
      ult_vei := [l,tf];                /* Atualiza infos sobre o último veículo.  */
      to operacao
      [] on (full(via));                /* Via com MAX_NUM_VEI finaliza o processo.*/
      to finalizado
    end
AIC
```

A Listagem 2 apresenta a modelagem em Fiacre após a correção do erro conforme proposto por Furtado (2017). Apenas as partes do código que diferem em relação à Listagem 1 foram comentadas. Além de condições adicionais para verificar se o último veículo está à frente na mesma faixa, foram necessárias variáveis para manter registro do último veículo a entrar na ZC em cada faixa.

4.2. Especificação, verificação e contraexemplo com selt

A especificação de que dois veículos não podem ter janelas de tempo iguais na mesma faixa para os modelos apresentados nas Listagens 1 e 2 pode ser formalizada em lógica temporal por um conjunto de expressões representadas a seguir:

$$\Box\neg\left((\text{vei}[0][i] = \text{vei}[0][i+j]) \wedge (\text{vei}[1][i] = \text{vei}[1][i+j])\right). \tag{11}$$

para:

$$\text{vei} \in \text{via}, \{i : 0 \leq i \leq \text{NUM\_MAX\_VEI} - 1\}, \{j : 0 < j \leq \text{NUM\_MAX\_VEI} - i - 1\}. \tag{12}$$

Além da notação típica de teoria dos conjuntos, são usados os operadores lógicos de negação ($\neg$), igualdade ($=$), e conjunção ou E lógico ($\wedge$), e o operador de lógica temporal sempre ($\Box$). Notar que optou-se arbitrariamente por verificar apenas os estados com cinco veículos na fila

Listagem 2: Código fonte em FIACRE da estratégia discretizada depois da correção.

```
const NUM_MAX_VEI : nat is 5
type Veiculo is array 2 of nat
type Via is queue NUM_MAX_VEI of Veiculo
process AICC is states operacao, finalizado
  var via      : Via  := {||}, tam : 0..5 := 0, faixa : 0..1 := 0, tf : nat  := 0,
      tf_ult_0 : 0..1 := 0,     /* tf do último veículo na faixa 0              */
      tf_ult_1 : 0..1 := 0,     /* tf do ultimo veículo na faixa 1              */
      ult_vei  : array 2 of nat := [0,0]
  from operacao select
  on not (full(via));
    faixa := any;
    if empty(via) then
      tf := 1
    else
      if faixa = ult_vei[0] then
        tf := ult_vei + 1
      else                        /* Se o novo veículo estiver em faixa dife-    */
        if faixa = 0 then         /* rente do que o último e estiver na faixa 0. */
          if ult_vei[1] >= (tf_ult_0 + 1) then  /*Se o tempo do último veículo */
                                  /* é maior ou igual ao do último veículo na    */
                                  /* faixa 0 com aumento de segurança, os tempos */
            tf := ult_vei[1]      /* podem ser iguais. Se o tempo do último      */
          else                    /* veículo for menor que o do último veículo   */
            tf := tf_ult_0 + 1    /* na mesma faixa o tempo do novo veículo tem  */
          end                     /* que ser maior que do último na mesma faixa  */
        else                      /* Mesmas comparações e ações para faixa 1.    */
          if ult_vei[1] >= (tf_ult_1 + 1) then
            tf := ult_vei[1]
          else
            tf := tf_vei_1 + 1
          end
        end
      end
    end;
    via := enqueue(via,[l, tf]);
    tam := tam + 1;
    ult_vei := [l,tf];
    if faixa = 0 then             /* Atualiza tempo do último que chegou         */
     tf_ult_0 := ult_vei[1]       /* na faixa 0.                                 */
    else                          /* Atualiza tempo do último que chegou         */
     tf_ult_1 := ult_vei[1]       /* na faixa 1.                                 */
    end;
    to operacao
    [] on (full(via));
    to finalizado
  end
AICC
```

(NUM_MAX_VEI), apesar de serem suficientes apenas três veículos.

A execução do selt para a Especificação 11–12 percorre um grafo contendo todos os estados alcançáveis a partir do estado inicial e devolve verdadeiro se a especificação tem valor verdade

verdadeiro para todos os estados e devolve falso caso contrário. Para o modelo da Listagem 1, como era de se esperar, o selt devolve falso e gera um contraexemplo com transições e estados que levam a um estado para o qual a Especificação 11–12 tem valor verdade falso. O estado final do contraexemplo gerado pelo selt é apresentando na Tabela 3. Notar que os veículos #3 e #4 na faixa 0 têm alocado o mesmo tempo de saída da ZN. Esse é apenas um dos diversos contraexemplos que poderiam ter sido identificados pelo selt, mas foi o primeiro encontrado pelo algoritmo que faz a busca no espaço de estados do modelo.

Tabela 3: Contraexemplo encontrado pelo selt para o modelo da Listagem 1.

#	faixa	t_i^{f}
0	0	1
1	0	2
2	0	3
3	1	3
4	0	3

Para o modelo da Listagem 2, o resultado foi verdadeiro, ou seja a Especificação 11–12 tem valor verdade verdadeiro para todos os estados alcançáveis. Ou seja, a correção proposta por Furtado (2017) garante que não há colisão traseira causada pela alocação de janelas de tempo iguais para dois veículos na mesma faixa.

5. CONCLUSÕES

A engenharia de tráfego está passando por uma transformação sem precedentes com o surgimento dos veículos automatizados. Nesse contexto, novas ferramentas deverão ser incorporadas a essa área da engenharia para garantir a segurança dos usuários dos sistemas de tráfego. A verificação formal de uma estratégia de controle de interseção com veículos automatizados proposta em Zhang *et al.* (2017) foi demonstrada para um modelo simplificado. Essa estratégia não garante a operação livre de colisões em certas condições e uma solução para o problema proposta for Furtado (2017) também foi verificada. A verificação mostrou que a solução proposta evita as colisões.

Como trabalhos futuros, pretende-se modelar e verificar outras estratégias de controle propostas na literatura, bem como incorporar verificação no processo de projeto de novas estratégias.

REFERÊNCIAS

Baier, C. e J.-P. Katoen (2008) *Principles of Model Checking*. MIT Press.

Berthomieu, B.; J.-P. Bodeveix; M. Filali; H. Garavel; F. Lang; F. Peres; R. Saad; J. Stoecker; F. Vernadat; P. Gaufillet *et al.* (2007) The syntax and semantics of Fiacre. *Report LAAS*, n. 07264.

Chen, L. e C. Englund (2016) Cooperative Intersection Management: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, v. 17, n. 2, p. 570–586.

Clarke, E. M.; O. Grumberg e D. Peled (1999) *Model checking*. MIT press.

Clarke, E. M. e J. M. Wing (1996) Formal methods: State of the art and future directions. *ACM Computing Surveys*, v. 28, n. 4, p. 626–643.

Farail, P.; P. Gaufillet; A. Canals; C. Le Camus; D. Sciamma; P. Michel; X. Crégut e M. Pantel (2006) The TOPCASED project: a toolkit in open source for critical aeronautic systems design. *Embedded Real Time Software*, v. 781, p. 54–59.

Furtado, H. S. (2017) «Evaluation of automated intersection control strategies for automated vehicles». Inglês. Dissertação de mestrado. Florianópolis, SC, Brasil: Programa de Pós-graduação em Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina.

Loos, S. M. e A. Platzer (2011) Safe intersections: at the crossing of hybrid systems and verification. *14th International IEEE Conference on Intelligent Transportation Systems*, p. 1181–1186.

Lygeros, J.; D. N. Godbole e S. Sastry (1998) Verified hybrid controllers for automated vehicles. *IEEE Transactions on Automatic Control*, v. 43, n. 4, p. 522–539.

Malikopoulos, A. A.; C. G. Cassandras e Y. J. Zhang (2018) A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. *Automatica*, v. 93, p. 244–256.

Ölveczky, P. C. e J. Meseguer (2010) Specification and verification of distributed embedded systems: a traffic intersection product family. *arXiv preprint arXiv:1009.4265*.

Platzer, A. (2010) *Logical analysis of hybrid systems: proving theorems for complex dynamics*. Springer Science & Business Media.

Rios-Torres, J. e A. A. Malikopoulos (2017) A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, v. 18, n. 5, p. 1066–1077.

Shladover, S. E. (2018) Connected and automated vehicle systems: introduction and overview. *Journal of Intelligent Transportation Systems*, v. 22, n. 3, p. 190–200.

Stursberg, O.; A. Fehnker; Z. Han e B. H. Krogh (2004) Verification of a cruise control system using counterexample-guided search. *Control Engineering Practice*, v. 12, n. 10, p. 1269–1278.

Topcased (2012) *Manual Reference Pages - selt*. Disponível em: <http://projects.laas.fr/tina/manuals/selt.html#2>. Acesso em: 03 junho 2017.

Zhang, Y.; C. G. Cassandras e A. A. Malikopoulos (2017) Optimal control of connected automated vehicles at urban traffic intersections: a feasibility enforcement analysis. *American Control Conference 2017*, p. 3548–3553.

Joana Alves dos Santos (joana97santos@gmail.com)
Fábio Luis Baldissera (fabiobaldissera@gmail.com)
Rodrigo Castelan Carlson (rodrigo.carlson@ufsc.br)
Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina
Florianópolis, SC, 88040-900, Brazil