MODELAGEM DE PROCESSOS QUÍMICOS E SISTEMAS DE CONTROLE COMO MÁQUINAS DE ESTADOS FINITOS

F.A.COELHO[1], R.J.ZEMP[1] e A.C.V.MELO[2]

[1] Universidade Estadual de Campinas – Faculdade de Engenharia Química
[2] Universidade de São Paulo – Instituto de Matemática e Estatística
E-mail para contato: filipea_c@hotmail.br

RESUMO – *As áreas da engenharia e ciência da computação dispõem de técnicas matemáticas sofisticadas para avaliar se um projeto de hardware e/ou software está correto em relação aos requisitos especificados. Essas técnicas, denominadas de verificação formal, englobam uma diversidade de métodos para provar matematicamente se um sistema viola (ou não) os requisitos (propriedades lógicas) especificados pelo projetista. Apesar de poder prever a violação de requisitos desejados de forma automática, a aplicação das técnicas de verificação formal em projetos de engenharia, e em particular na engenharia química, torna-se pouco prático porque grande parte dos verificadores formais requerem que os sistemas sejam modelados como máquinas de estados finitos (MEFs), enquanto os engenheiros não têm treinamento adequado para esse tipo de modelagem. Por outro lado, os engenheiros são treinados a modelar (especificar) os sistemas através de equações diferenciais. Este trabalho tem por objetivo demonstrar como processos químicos e sistemas de controle (contínuos e/ou lógicos) podem ser modelados como MEFs a partir de equações diferenciais que envolvem tempo. As MEFs foram implementadas no software Matlab e NuSMV.*

1. INTRODUÇÃO

A operação adequada de processos químicos depende de diversos fatores, tais como: o funcionamento correto dos equipamentos, eficácia da estrutura de controle e intervenção de profissionais devidamente treinados. Entretanto, a interação entre essas diferentes partes que constituem o sistema químico pode ser consideravelmente complexa para ser prevista na etapa de projeto Para tanto, precisaríamos especificar todas as condições operacionais de cada uma das partes, para então combiná-las e analisá-las. Essa abordagem levaria a uma explosão combinatorial dos casos a serem avaliados, o que torna a análise impraticável para ser feita manualmente (ou até mesmo computacionalmente).

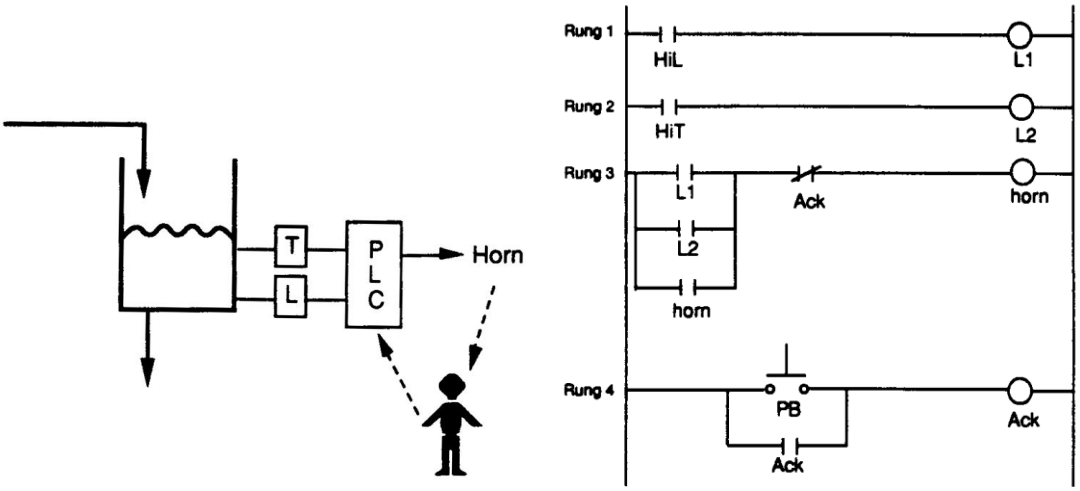
Com problemas semelhantes, a engenharia (e a ciência) da computação desenvolveram as técnicas de verificação formal (BAIER e KATOEN, 2008) para avaliar se um projeto de *hardware* e/ou *software* atende a determinadas propriedades desejadas. Para viabilizar a aplicação dessas técnicas, baseadas em teorias matemáticas, foram criadas ferramentas que automatizam a verificação

se propriedades (requisitos) são preservadas (ou não) no dado modelo do sistema. Essa verificação de requisitos teria grande valor se aplicada à engenharia química, pois permitiria avaliar se um projeto de sistemas químicos não viola propriedades especificadas pelo engenheiro (por exemplo, "um tanque nunca deve vazar" ou "uma resistência nunca deve ser ligada com o reator vazio") e orientaria o projeto na direção de um sistema "livre" de falhas.

1.1. Problema exemplo

Uma das aplicações da verificação formal em engenharia é na verificação de diagramas ladder (SILVA, 2008) os quais são utilizados para programação de CLPs (Controladores Lógico Programáveis). A Figura 1(a) ilustra um sistema onde os alarmes de nível alto e temperatura alta do tanque de armazenamento são ativados por um CLP e a sirene (*horn*) é silenciada pelo operador. O diagrama ladder para esse processo está apresentado na Figura 1(b).

No diagrama ladder da Figura 1, os trilhos verticais indicam os trilhos de energia, enquanto as linhas horizontais, representam possíveis fluxos de energia ou sinal. Vários símbolos para botões, contatos e relés podem ser colocados nas linhas do ladder. Se os contatos apropriados são ativados, o relé é energizado e os contatos correspondentes são fechados. Por exemplo, o contato de nível alto (HiL) na linha 1, provoca a ativação do relé L1 (também na linha 1) e causa o fechamento do relé L1 na linha 3.



(a) (b)

Figura 1 – Exemplo de um diagrama ladder para controle de um tanque. Fonte: Moon *et al* (1992).

Diagramas ladder extensos são difíceis de ler e interpretar. Portanto, para saber se a programação feita corresponde ao comportamento desejado, são realizados testes manuais para as entradas e observa-se as respostas do CLP. Com isso, só temos certeza de que o sistema funciona para o conjunto de testes realizados. Com a verificação formal, podemos provar que o sistema atende aos requisitos especificados (propriedades descritas em lógica temporal) e isso é uma verdade do sistema,

independentemente dos dados de teste.

Por exemplo, podemos descrever a propriedade de que a sirene sempre vai soar quando o nível alto é atingido usando lógica temporal (ilustrada na Figura 2) e verificar se o diagrama ladder preserva essa propriedade. Assim, com a propriedade lógica temporal e um modelo do diagrama ladder definidos, o verificador vai estimular o modelo de forma exaustiva e terá dois possíveis resultados: ou o sistema preserva a propriedade; ou o sistema viola e propriedade e um contraexemplo com a sequência de estados (quais contatos foram acionados) que levaram o ladder a não acionar a sirene quando foi detectado nível alto.

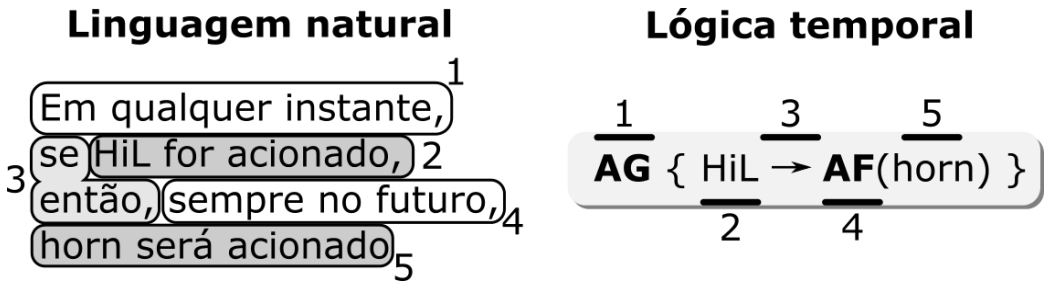


Figura 2 – Exemplo de propriedade lógica temporal e o significado em português.

1.2. Solução proposta

A inconveniência da verificação de modelos que dificulta sua aplicação na engenharia química é que para a maioria dos verificadores, os sistemas devem ser modelados como Máquinas de Estados Finitos (MEFs), que pressupõem a existência de estados discretos do sistema, mesmo que ele seja contínuo. As máquinas de estados finitos (ou autômatos finitos) são modelos matemáticos que representam os estados discretos e finitos de um sistema. A Figura 3 ilustra um diagrama de estados de uma máquina de estados finitos (correspondente ao diagrama ladder da Figura 1): os estados discretos do sistema são representados por elipses, as transições entre os estados são representadas por setas e o estado inicial (por onde a máquina começa sua execução) é representado por uma seta sem origem chegando a esse estado. Os estados possuem proposições que são verdadeiras enquanto a máquina estiver naquele estado, assim, se o autômato da Figura 3 iniciar em s1 e em algum momento HiT ocorrer, ele vai passar por uma transição para o estado s3, onde são verdadeiros os valores de L2, horn e o próprio HiT. Apesar dos autômatos possuírem estados finitos, a execução pode ser infinita, efetuando infinitas transições entre os estados, se assim for permitido.

É importante notar que os autômatos finitos não modelam o tempo de forma explícita. Não é possível, por exemplo, extrair informações sobre o tempo que o sistema da Figura 3 leva para sair do estado s1 para o estado s3. Autômatos finitos focam na descrição dos estados do sistema e dos eventos que provocam as transições, portanto, são modelos essencialmente qualitativos.

Sistemas encontrados na engenharia química podem ser modelados como autômatos finitos, porém raramente o são, visto que os engenheiros químicos estão habituados a descrever estes sistemas por meio de equações diferenciais, equações algébricas, entre outros modelos contínuos. A mudança

de paradigma de modelagem, entretanto, é essencial para se utilizar das técnicas já estabelecidas de verificação. Nesse contexto, este trabalho demonstra uma metodologia para discretização de equações diferenciais ordinárias de 1ª ordem de modo a transformá-las automaticamente em máquinas de estados finitos para fins de verificação formal.

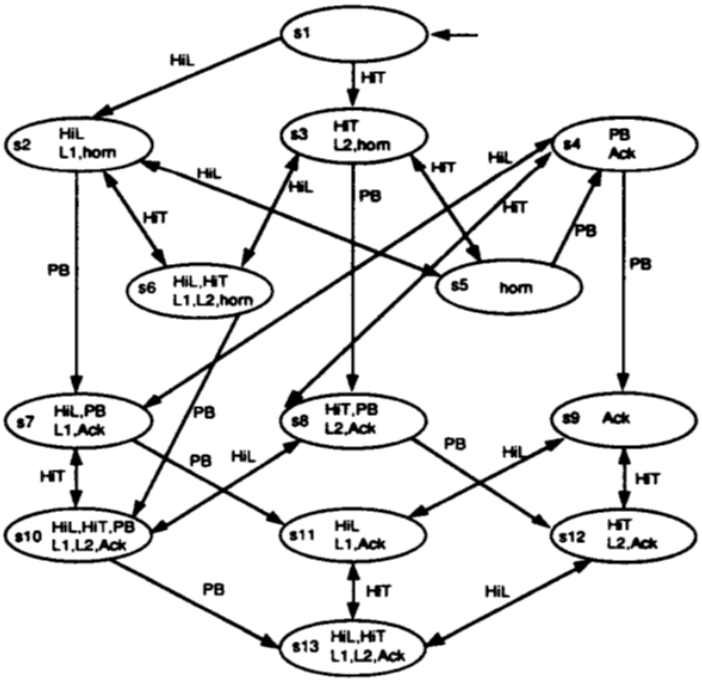


Figura 3 – MEFs do diagrama ladder da Figura 1. Fonte: adaptado de Moon *et al* (1992).

2. METODOLOGIA

A metodologia proposta se baseia no método de Euler para integração numérica Equação 1.

$$y_{i+1} = y_i + \frac{dy_i}{dt}\Delta t \qquad (1)$$

Fixado o passo de tempo Δt, o novo valor da variável y dependerá apenas do valor atual, da derivada avaliada em y_i e do instante de tempo atual. Se cada um dos componentes da Equação 1 (a variável *y*, a derivada de *y* e o tempo) for representado por um autômato finito, é possível efetuar uma integração numérica "simbólica" cujo resultado será interpretado dependendo do estado que cada uma das máquinas esteja.

Para representar a Equação 1, três autômatos finitos são utilizados: um autômato que controla o passo de tempo (gera um sinal de *clock* para sincronizar todas as máquinas); um autômato que gerencia o valor da derivada da variável de integração; e um autômato que armazena o valor da variável de integração. Esses autômatos compartilham variáveis a fim de se comunicar e efetuar as transições de forma sincronizada.

A máquina de estados $\mathcal{M}_t$ oscila entre dois estados durante a integração (S1 e S2). Durante a transição de S1 para S2, a passagem de tempo é computada, armazenando o valor em uma variável inteira *t* e atualizando o valor da variável de integração. A transição de S2 para S1 coordena a mudança de estado da máquina de derivadas. O "intervalo de integração" pode ser definido estipulando o valor máximo para *t*, que se for atingido bloqueia as transições de $\mathcal{M}_t$ e consequentemente interrompe as transições dos outros autômatos.

O autômato $\mathcal{M}_{dy}$ mapeia o valor da variável de integração no respectivo valor da derivada. Durante a transição de $\mathcal{M}_t$ para S1, o valor $\Delta y = dy/dt \cdot \Delta t$ é atualizado, o qual será usado para calcular o novo valor de *y*, conforme Equação 1. Visto que não é possível implementar diretamente a equação diferencial na máquina de estados, os valores das derivadas foram calculados e armazenados nas transições de $\mathcal{M}_{dy}$. Isso é feito inicialmente no Matlab escolhendo valores representativos para cada intervalo (estados da máquina $\mathcal{M}_y$) para a qual a variável de integração foi discretizada e avaliando a derivada nesses valores. As derivadas são normalizadas para números inteiros e armazenadas em uma lista, que é utilizada para gerar as transições do autômato $\mathcal{M}_{dy}$. Esta etapa está esquematizada na Figura 4.

Figura 4 – Etapas para geração dos valores das derivadas da variável de integração.

Finalmente, o autômato $\mathcal{M}_y$ tem por finalidade apenas armazenar o valor da variável de integração *y*. Este autômato observa o passo de tempo e o estado de $\mathcal{M}_{dy}$ para efetuar as transições, de modo a imitar uma integração numérica. O programa que executa essa metodologia foi escrito em Matlab, onde são definidas as discretizações, o arquivo com a derivada (escrito da mesma forma que é utilizado pelas funções de integração numérica do Matlab) e o "intervalo de integração". Depois de passar pelo procedimento mostrado na Figura 4, o programa escreve o arquivo para ser lido pelo software NuSMV (NuSMV, 2016). O NuSMV é um *software* de verificação *open source* que também tem a capacidade de executar simulações de máquinas de estados. Os resultados gerados pelo NuSMV são armazenados em um arquivo .xml que é lido pelo Matlab e posteriormente são gerados os gráficos das variáveis. A Figura 5 resume as etapas executadas pelo Matlab e NuSMV.

Com essa estratégia é possível não apenas modelar o sistema físico, mas também sistemas de controle contínuos como controladores PID. A lógica de CLPs (lógica ladder) também pode ser traduzida diretamente para máquinas de estados finitos, sem a necessidade da metodologia apresentada, como apresentado por Moon (1992).

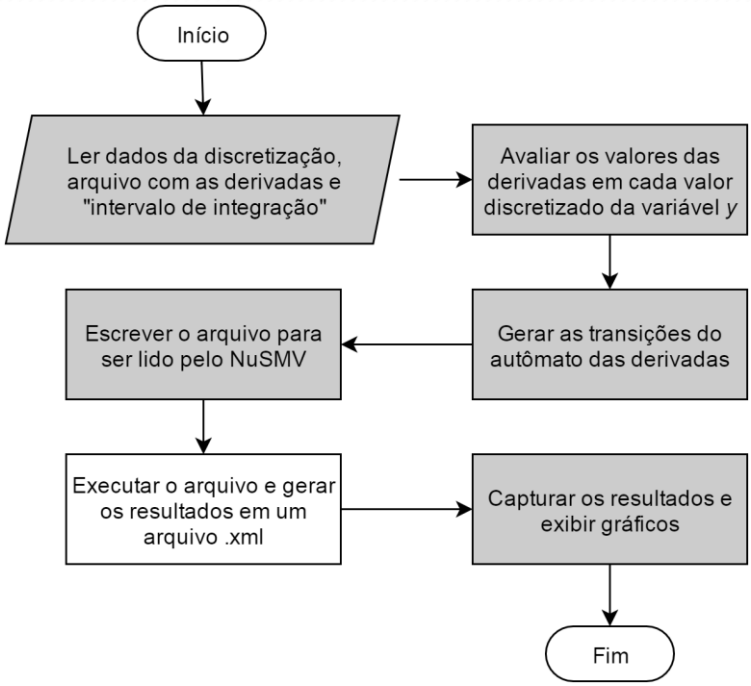


Figura 5 – Fluxograma da execução da metodologia (etapas em cinza foram executadas no Matlab e em branco no NuSMV).

Como estudo de caso foi escolhido um modelo de um tanque a fim de testar o funcionamento da metodologia proposta.

3. PROVA DE CONCEITO

O modelo consiste de uma equação diferencial ordinária, mostrada na Equação 2.

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \frac{Q_e - c_v\sqrt{h}}{A} \tag{2}$$

h denota o nível do tanque [m], A é a área transversal [m²], Q_e é a vazão de alimentação [m³/s] e c_v é a constante da válvula de descarga [$m^{2,5}$/s]. A Figura 6 mostra os resultados da simulação do tanque sem o sistema de controle e a comparação com o resultado obtido por integração numérica da equação diferencial. Foi utilizada a condição inicial de 0 m, uma vazão de 0,5 m³/s e c_v igual a 0,2.

Observa-se que o aumento do número de faixas de discretização refina a solução, aproximando-a da solução numérica obtida com a equação diferencial (Figura 6 (a) e (b)). Notou-se, no entanto, que um aumento excessivo piora a solução nas regiões onde a variável muda mais rápido (Figura 6(c)), porque a máquina só consegue efetuar transição para o estado adjacente e um grande número de faixas de discretização exigiria transição para estados não adjacentes, o que gera um atraso na solução.

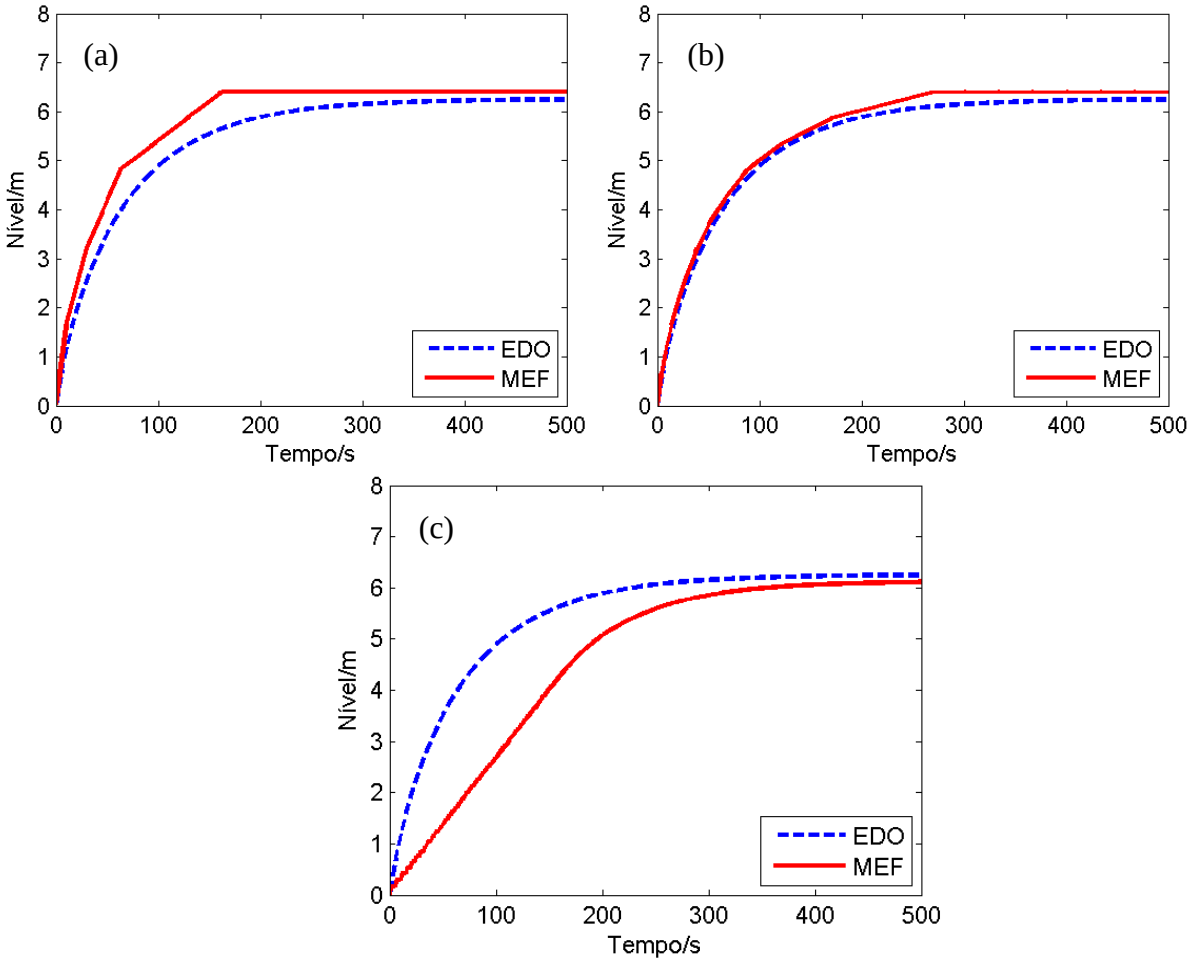


Figura 6 – Simulação com diferentes discretizações do nível: (a) 4, (b) 15 e (c) 300 faixas de discretização.

O exemplo também foi modificado para incluir um Controlador Lógico Programável (CLP) para controlar o nível. Foi considerado que o tanque possui uma linha de alimentação com uma válvula solenóide que abre quando energizada. A programação do CLP está mostrada na Figura 7. LL é um sensor que indica o nível baixo (posicionado em 1,6 m), HL é um sensor que indica o nível alto (posicionado em 4 m) e V é o relé que aciona a válvula de alimentação. O resultado para 2000 s de simulação está mostrado na Figura 8. Observa-se que foi possível simular o sistema de controle lógico e que este conseguiu manter o nível dentro da faixa de valores para a qual ele foi programado.

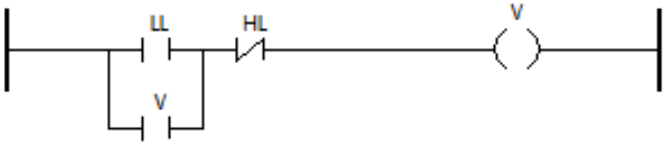


Figura 7 – Linha de programação do CLP que controla a abertura da válvula de alimentação.

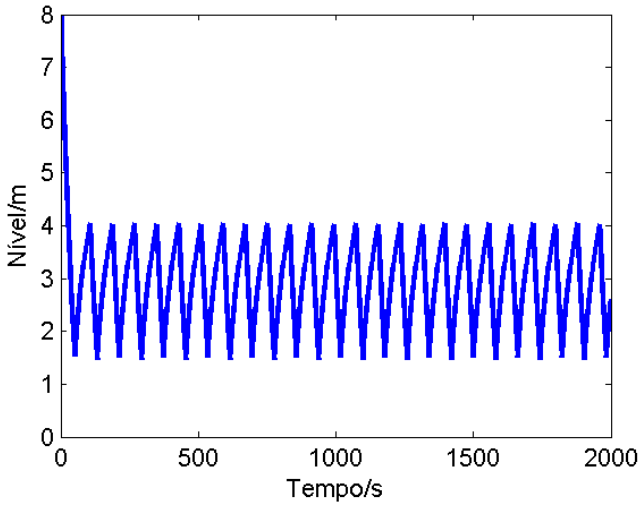


Figura 8 – Nível do tanque sob ação do sistema de controle lógico

4. CONCLUSÕES

O uso de verificação formal para certificar a aderência de modelos aos seus requisitos tem grande potencial nas engenharias. O desenvolvimento de metodologias simples para tradução automática dos modelos tradicionais (baseados em equações diferenciais/algébricas) para MEFs pode facilitar a difusão da técnica no campo das engenharias. Neste trabalho foi demonstrada uma metodologia para tradução automática de equações diferenciais para máquinas de estados finitos, fazendo-se analogias com o método de integração numérica de Euler. Por enquanto foi demonstrado seu funcionamento para EDOs de 1ª ordem não-lineares, mas os autores acreditam ser possível representar EDOs de 2ª ordem e também equações diferenciais parciais ao adaptar a metodologia apresentada.

5. REFERÊNCIAS

BAIER ,C.; KATOEN, J.P. *Principles of Model Checking*. MIT Press, 2008.

MOON, I.; POWERS, G.J; BURCH, J.R.; CLARKE, E.M. Automatic verification of sequential control systems using temporal logic. *AIChE Journal*, 38(1):67-75, 1992.

MOON, Il. *Automatic verification of discrete chemical process control systems*. 1992. 191 f. Tese (Doutorado) - Curso de Engenharia Química, Departamento de Engenharia Química, Carnegie Mellon, Pittsburgh, 1992.

SILVA, A.M. *Aplicação de verificação de modelos a programas de CLP*: explorando a temporização. 2008. 121 f. Dissertação (Mestrado), Instituto Militar de Engenharia, Rio de Janeiro, 2008.

NuSMV. An overview of NuSMV. Disponível em: < http://nusmv.fbk.eu/NuSMV/>. Acesso em: 23 de fevereiro de 2016.