Copyright © 2014 by ABCM

SISTEMA DE AUTOMAÇÃO INTELIGENTE PARA SIMULAÇÃO E CONTROLE DE TRÁFEGO DE VEÍCULOS

Érika Reinaldo Marques da Silva, erikarmsilva@gmail.com[1]
André Murilo, andremurilo@gmail.com[2]

[1]Universidade Federal de Pernambuco, Av. Prof. Moraes Rego, 1235 - Cidade Universitária
[2]Universidade de Brasília, Faculdade Gama, Asa Norte s/n

Resumo:*Congestionamentos em avenidas estão cada vez mais presentes e intensos nos centros urbanos, fruto de um aumento importante da frota de veículos, além do crescimento desordenado das cidades. Desta forma, o estudo das técnicas de engenharia de tráfego tem ganhado importância considerável nos últimos anos no intuito de identificar os problemas e apontar soluções para o fluxo de veículos dentro das cidades. Assim, incialmente, nesse projeto foi desenvolvido um modelo simplificado de um sistema a eventos discretos a partir de uma rede de petri temporizada capaz de simular o funcionamento do cenário proposto. Modelado o sistema com auxílio de rede de petri, foi implementada a arquitetura do sistema de automação, na qual CLPs foram integrados em Redes Ethernet, assim como o sistema de supervisão de controle SCADA para monitoramento do processo. A programação foi realizada de modo a representar cruzamentos próximos, onde o fluxo de veículos em determinada via era simulado pelos potenciômetros das bancadas. Finalmente, foram desenvolvidas funções matemáticas a fim de temporizar todos os semáforos de modo a reproduzir uma condição ótima de fluxo, a partir do ajuste automático dos temporizadores dos semáforos das vias de trânsito.*
Palavras-chave: *Engenharia de tráfego, Rede de Petri, CLP, SCADA, Simulação*

1. INTRODUÇÃO

A aplicação de técnicas de controle e implementação de modelos que visam à melhoria dos congestionamentos tem ganhado cada vez mais importância, visto que problemas de tráfego são recorrentes em grandes metrópoles. Vários fatores podem ser considerados para tais problemas, entre eles, semáforos configurados de forma ineficiente. Assim, otimizar o fluxo de veículos nos cruzamentos é uma opção para melhoria da qualidade na mobilidade urbana. Um aspecto básico para tal é garantir a sincronia semafórica, a qual pode deve ser obtida na modelagem do sistema, nesse caso utilizando uma representação formal de Redes de Petri, que consiste em uma linguagem gráfica eficiente de modelagem de sistemas a eventos discretos (Cassandras and LAfortune (2008), David and Alla (2010), Foyo (2009), Valette and Cardoso (1997)). Nesses casos, o sistema de controle de tráfego pode ser automatizado e sincronizado de forma a se obter uma condição ótima de funcionamento (Febbraro, Giglio and Sacco (2004), Gadelha, Repolho, Almeida and Jnior (2010), Garcia (2007), List and Cetin (2004),Tolba, Lefebvre, Thomas and Moudni (2005)).

Uma das formas de se implementar este tipo de solução é utilizar arquiteturas de automação baseada em Controladores Lógico-Programáveis (CLP) e sistemas SCADA (*Supervisory Control and Data Acquisition*). O CLP basicamente consiste em um dispositivo microprocessado que possui canais de entrada e saída pelas quais realiza a interface com o processo. Várias vantagens decorrem da utilização deles, tais como facilidade de programação e manutenção, robustez, fácil capacidade de expansão e flexibilidade. Os sistemas SCADA por sua vez, são responsáveis pela supervisão, monitoramento e aquisição de dados provenientes dos CLPs (Boyer, 2010).

Dessa forma, esse relatório mostra o desenvolvimento de um sistema inteligente para o controle de um modelo simplificado de tráfego viário com o *set-up* automático de temporizadores de semáforo através de uma arquitetura de automação CLP-SCADA.

Além disso, as filas formadas com o acúmulo de veículos são aspectos importantes a serem analisados visando a otimização do fluxo viário. Assim, o sistema foi modelado no *software Matlab* utilizando por base a teoria de filas (Paravisi *et al.* (2006), Garcia (2007), Moraes *et al.* (2011)), a fim de determinar a solução ótima que minimiza o tamanho destas.

2. METODOLOGIA

O presente trabalho foi dividido em 4 partes: a primeira consistiu na modelagem do sistema com base na teoria de filas, onde foi desenvolvida uma solução global capaz de avaliar o sistema como um todo, analisando a dependência entre os

cruzamentos. Em seguida, foi realizada a modelagem de um sistema a eventos discretos, mais especificamente utilizando uma rede de petri temporizada. Esta, por sua vez, fora utilizada como base nas etapas posteriores de implementação e simulação. Assim, na terceira etapa, foi montada a arquitetura do sistema, estabelecendo a comunicação CLP-SCADA, e foi determinado o cenário de simulação para finalmente ser desenvolvida a solução local, a qual avalia cada cruzamento como independente dos demais. Com isso, enfim, foi feita a programação dos CLPs e elaboração do sistema SCADA de supervisão e monitoramento, sendo possível implementar o sistema e atestar o seu funcionamento

2.1 Modelagem com Teoria de Filas

Em situações práticas de tráfego, os veículos formam filas sob duas condições:

• O sinal vermelho está acionado e, então, todo veículo que chega ao cruzamento servirá para aumentar o tamanho da fila;

• O sinal verde está acionado, mas a fila na via é grande a ponto de a dispersão ser insuficiente e, assim, os veículos que chegam ao cruzamento ainda entram em fila.

A fim de garantir um melhor desempenho do sistema de tráfego veicular, o comportamento das filas geradas no sistema deve ser analisado, sendo um parâmetro de avaliação da coordenação semafórica. Assim, foram feitas modelagens do sistema no programa *Matlab* usando como base a Teoria das Filas e avaliando todos os cruzamentos de uma só vez, caracterizando a solução global.

A equação das filas é dada pela equação apresentada a seguir:

$$X(t) = X(t-1) + a(t, t-1) - c(t, t-1) \tag{1}$$

Onde $X(t)$ é a quantidade de veículos em fila no tempo t; $a(t, t-1)$ é o número de veículos que chegam no intervalo de tempo $(t, t-1)$ e $c(t, t-1)$ é o número de veículos que saem da fila no intervalo de tempo $(t, t-1)$.

Como o dado de entrada do sistema é o fluxo de veículos, ou seja, o número de veículos que chega ao cruzamento por segundo, optou-se por fazer uma interação para analisar o acúmulo de veículos da via a cada intervalo de tempo, o qual fora definido pela variável $passo$. Assim, determinado o tempo do sinal verde de uma via (t_{verde}), foram criadas variáveis (u_A e u_B) responsáveis por habilitar a dispersao dos veículos nas duas vias do cruzamento. Para tal, elas foram definidas como:

$$u_A = max(sign(t_{verde} - t + passo), 0) \tag{2}$$

$$u_B = max(sign(t - t_{verde}), 0) \tag{3}$$

Onde t representa o tempo corrente.

Portanto, $u_A = 1$ e $u_B = 0$ para $t <= t_{verde}$ (caso em que o semáforo da via A está aberto), enquanto que $u_A = 0$ e $u_B = 1$ para $t > t_{verde}$ (caso em que o semáforo da via A está fechado). Assim, as equações das filas para as duas vias do cruzamento podem ser escritas como:

$$X_A(t+1) = max(0, X_A(t) + a_A(t) - c_A(t).u_A) \tag{4}$$

$$X_B(t+1) = max(0, X_B(t) + a_B(t) - c_B(t).u_B) \tag{5}$$

Onde X_A é a quantidade de veículos em fila na Via A do Cruzamento; $a_A(t)$ é o número de veículos da via A que está chegando ao cruzamento no instante t; e $c_A(t)$ é o número de veículos que estão sendo dispersados com a abertura do semáforo no instante t. Com isso, foi possível realizar-se testes com diversos valores de tempos para o sinal verde, variando-se os fluxos das vias.

Quando interseções semaforizadas são localizadas nas proximidades de outras, o controle de cada interseção isoladamente interfere no comportamento de chegada de veículos nas outras interseções. Baseado nisso, foi proposta uma solução global para o sistema onde o fluxo de uma via é diretamente dependente do fluxo proveniente do cruzamento anterior. Para isso, admitiu-se que 80% do fluxo de uma via permanecia nesta com a abertura do semáforo e, consequentemente, 20% se encaminhava à via transversal. Então, por exemplo, quando o sinal da Via A no primeiro cruzamento está aberto, tem-se os seguintes fluxos para os cruzamentos posteriores:

$$a_{A2}(t) = min(0.8.c_{A1}, 0.8.(X_{A1}(t) + a_{A1}) \tag{6}$$

$$a_{B2}(t) = min(0.2.c_{A1}, 0.2.(X_{A1}(t) + a_{A1}) \tag{7}$$

Onde $a_{A2}(t)$ é o número de veículos que chega ao cruzamento 2 pela via A no instante t; $a_{B_2}(t)$ é o número de veículos que se encaminham ao segundo cruzamento da via B no instante t.

Feito isso, foram inseridos valores numéricos para temporização semafórica e para os fluxos e avaliadas as filas formadas. Para isso, foram plotados gráficos representativos da acumulação de veículos nas vias e calculadas as áreas sob estes. Os valores do tempo de abertura dos semáforos foram alterados até que fosse obtido um valor para tal que resultasse em uma mínima área sob os gráficos. Ou seja, uma solução global em que o valor da soma dos tamanhos das filas formadas nos cruzamentos fosse a menor possível.

Prosseguindo com essa mesma perspectiva, foi desenvolvida, então, uma função custo representando a soma das filas formadas em cada via do cruzamento:

$$J = p_A.J_A + p_B.J_B \tag{8}$$

Onde p_A corresponde à ponderação do custo da via A, p_B corresponde à ponderação do custo da via B, J_A é o custo associado ao acúmulo de carros na via A, J_B é o custo associado ao acúmulo de carros na via B. Assim, esses custos para cada via podem ser escritos como:

$$J_A(t+1) = J_A(t) + x_A(t).passo.u_B \tag{9}$$

$$J_B(t+1) = J_B(t) + x_B(t).passo.u_A \tag{10}$$

Finalmente, com o auxílio do *software Matlab*, os resultados da minimização da função custo puderam ser obtidos de forma automática, o que corresponde aos *setups* ótimos de abertura dos semáforos.

2.2 Sistemas a Eventos Discretos

Os Sistemas de tráfegos de veículos podem ser modelados e analisados como sistemas a eventos discretos de modo a poder ser desenvolvida uma estrutura lógica capaz de simular o comportamento destes. Rede de Petri é reconhecidamente uma das mais adequadas ferramentas para descrição e análise da sincronização, comunicação e fonte de compartilhamento entre processos concorrentes, sendo responsável por modelar e garantir sincronia e concorrência entre os sistemas a serem modelados através de eventos e condições.

Então, dando sequência ao projeto, foi desenvolvido um modelo discreto simplificado do sistema (Figura 1), o qual atende às especificações deste. O uso de Redes de Petri para modelagem apresentou-se como solução para problemas de sincronização, como erros cumulativos de temporização, os quais podem ocasionar congestionamentos; sendo esses, obviamente, indesejáveis para esse sistema. Além disso, possibilitou a simulação do sistema com visualização simultânea de toda a estrutura e comportamento dele, tornando possível definir limites e verificar a ocorrência de falhas sem causar danos.

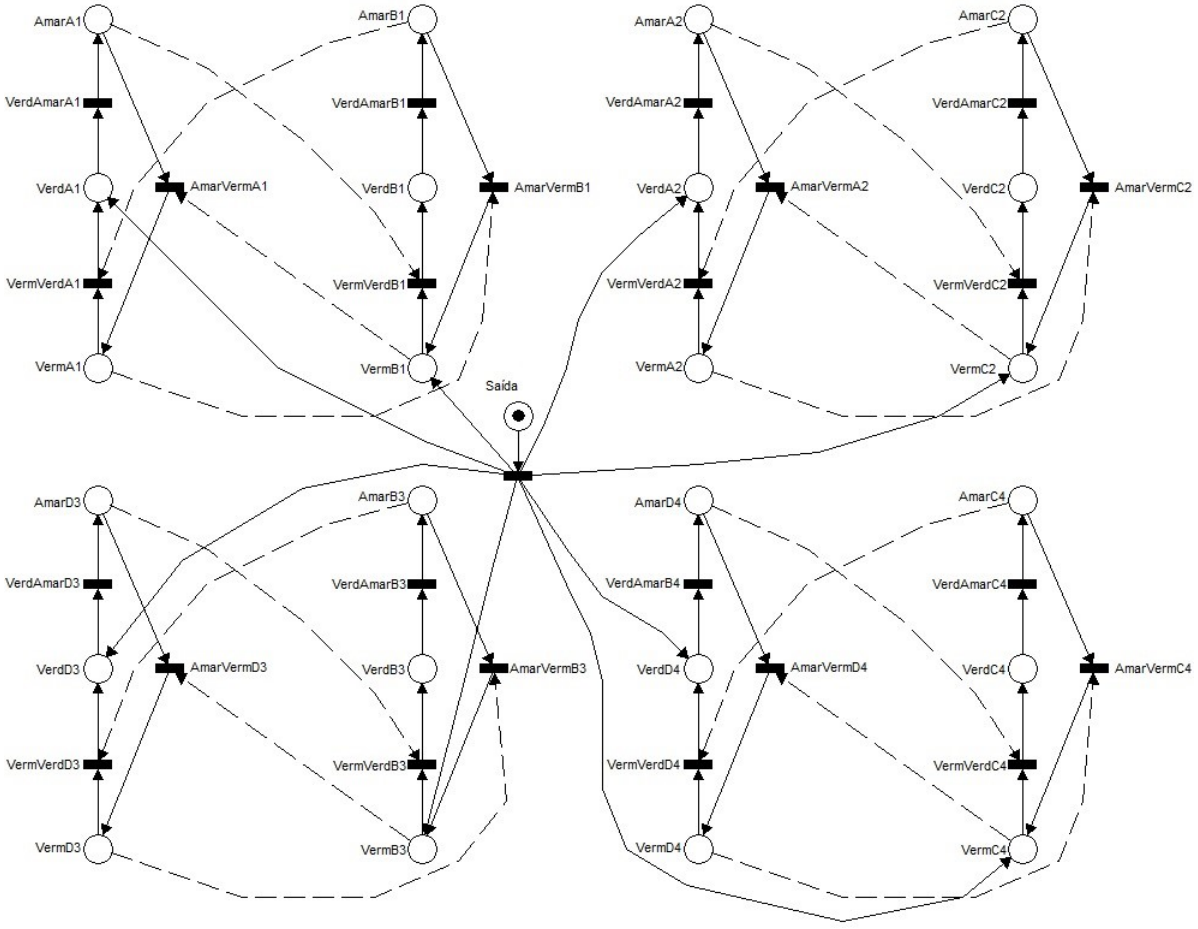

Figura 1: **Rede de petri do sistema**

Essa representação matemática consiste de posições, transições e arcos, onde esses últimos partem de uma posição para uma transição (ou vice-versa) e estão associados a um número inteiro fixo que serve para ponderar aquele caminho.

Para esse projeto, a Rede de Petri foi desenvolvida de modo que as posições representassem o semáforo com a respectiva sinalização (luzes vermelha, amarela ou verde acesas); marcações indicassem qual a condição do semáforo naquele instante; arcos mostrassem em qual sentido as marcações deveriam percorrê-los; e transições habilitadas fossem responsáveis por imprimir uma condição de mudança de estado.

Por haver a necessidade de que o disparo da transição ocorresse apenas após determinado intervalo de tempo (aquele em que o semáforo permanece na posição anterior à transição), foram colocados *delays* (atrasos) associados às transições, tornando o modelo uma Rede de Petri Temporizada. Com isso, finalmente, conseguiu-se chegar à modelagem base para implementação do sistema nos controladores lógico-programáveis, de modo a assegurar a sincronização dos semáforos nos quatro cruzamentos, evitando acidentes ou ainda formação desnecessária de fila de veículos.

2.3 Arquitetura do Sistema

Para montagem da arquitetura do sistema de automação, os quatro CLPs M340 presentes no laboratório foram interligados ao SCADA por rede Ethernet, como esquematizado na Figura 2. Assim, foram atribuídos IPs a cada um dos CLPs e, então, eles foram integrados em rede Ethernet conjuntamente com as CPUs, utilizando-se um *switch*. Assim, toda a comunicação tanto para programação, quanto para o controle supervisório foi feito via Ethernet TCP/IP, permitindo comunicação de todo o sistema de maneira simples, confiável e com resposta rápida. Depois de configurado, ele foi devidamente testado e constatou-se um eficiente desempenho da arquitetura, com as comunicações SCADA-CLP e CLP-CLP funcionando como desejado.

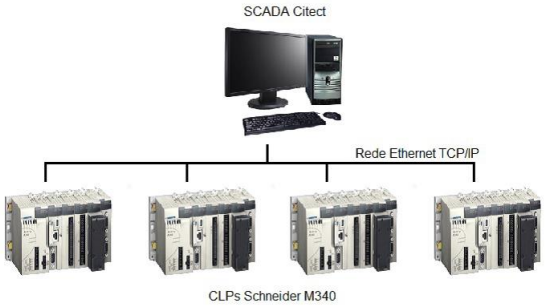

Figura 2: **Esquema da arquitetura do sistema**

A quantidade de CLPs disponíveis no laboratório limitou o número de cruzamentos a ser analisado em quatro, onde cada CLP seria responsável por transmitir os fluxos de carros de cada uma das vias de um cruzamento, através de variáveis analógicas.

Dessa maneira, foi desenvolvido o cenário de simulação, a fim de que a partir dele o sistema pudesse ser modelado e simulado. Os cruzamentos foram dispostos conforme mostrado na Figura 3.

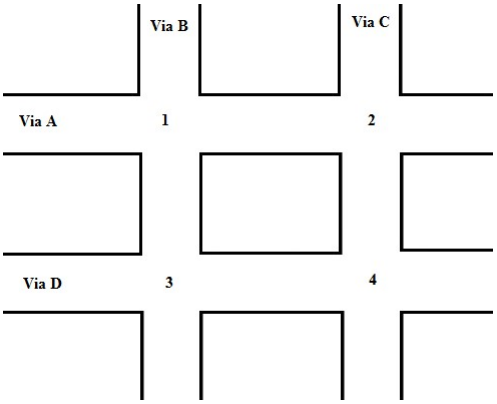

Figura 3: **Representação dos cruzamentos de veículos**

2.4 Controladores Lógico-Programáveis e SCADA

O próximo passo consistiu em dar início à programação dos CLPs para realização do controle de tráfego, utilizando o *software Unity-Pro M*. Para importar dados de um CLP para outro, a linguagem utilizada foi FBD (*Function Block Diagram*), a qual é uma linguagem gráfica que permite descrever um processo através de um conjunto de blocos interligados de forma semelhante a um circuito eletrônico. Na Figura 4, é mostrado um exemplo de comunicação entre dois CLPs através de uma área de memória, acessada pelos dispositivos a partir de blocos de leitura (*Read*) via Ethernet TCP/IP,

sendo gerados vetores compostos por variáveis de recepção. Para transmissão de dados, foram utilizados blocos de escrita (*Write*).

Figura 4: **Programa em FBD para recepção de dados de um CLP**

A programação foi feita essencialmente utilizando a linguagem *Ladder*, a qual consiste de uma técnica gráfica para programação, de modo que o acionamento de saídas é controlado em função da combinação lógica dos contatos de entrada. A Figura 5 ilustra o funcionamento do programa para acionar o sinal verde da Rua A.

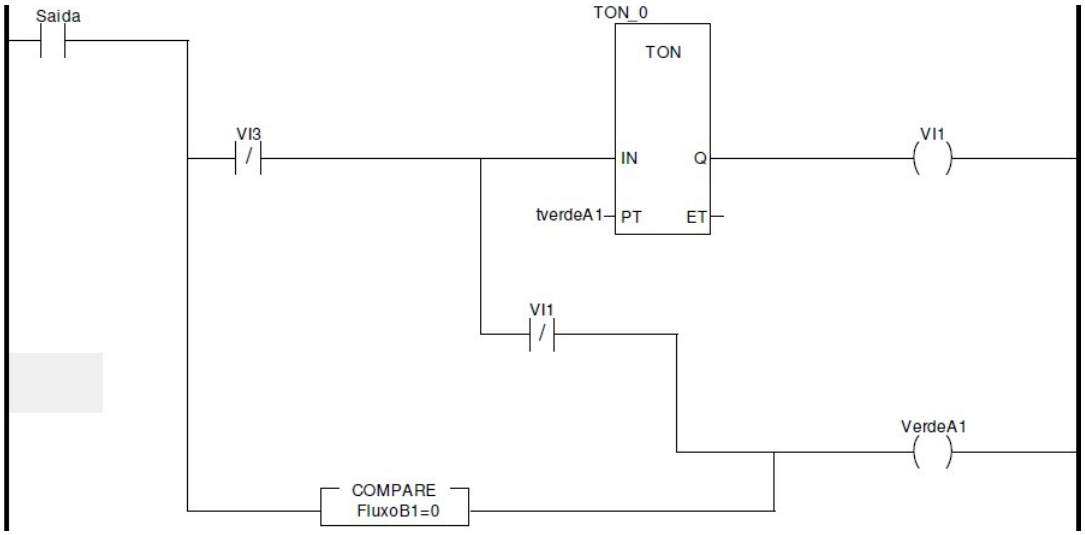

Figura 5: **Programa em Ladder para acionamento do sinal verde da via A**

Na Figura 5, os contatos nomeados como “VI” são variáveis internas do programa, utilizadas para memorizar e transmitir sinais. O contato “Saída” é ativado com o acionamento de um Botão de Liga responsável por dar início ao programa, enquanto que um de Desliga interromperia o funcionamento dele, desativando o sinal do contato. O bloco funcional TON representa o temporizador, o qual é habilitado ao aplicar-se um sinal na entrada “IN” e inicia a contagem do tempo estabelecido em “PT”. Transpassado esse tempo, o temporizador passa ao estado de concluído, ativando a saída “Q”, como mostrado na Figura 6.

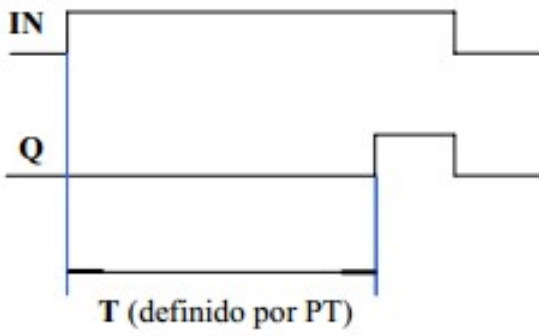

Figura 6: **Diagrama de tempos TON (D’Avila, 2011)**

Os tempos “PT” (*presets time*), portanto, serão os que vão definir as temporizações das vias. Como se deseja que haja um ajuste automático em função do fluxo de veículos, estes vão ser dados por variáveis tipo *TIME*, na Figura 5 nomeada por “$tverde_{A1}$” para o caso da via A, as quais são funções dos fluxos que chegam ao cruzamento. Para todos os outros semáforos, a programação para acionamento desenvolvida foi idêntica à mostrada para o sinal verde na Figura 5, havendo variações apenas nas variáveis para que fosse acionada a saída física correta com a temporização correspondente.

A fim de simular o fluxo de veículos em cada uma das vias e nas diferentes posições delas, foram usadas variáveis analógicas, através do uso de potenciômetros, sendo possível verificar a variação em tempo real das condições de tráfego em determinado entroncamento.

A partir de então, começou-se o desenvolvimento de funções matemáticas a fim de temporizar todos os semáforos das vias de trânsito de modo a reproduzir uma condição ótima de fluxo. Assim, analisando cada cruzamento individualmente, foi desenvolvida uma solução local, onde a temporização semafórica se apresentou como uma função linear baseada nos fluxos das vias. Tal solução definiria os "PTs" da Figura 5 e, portanto, havendo variação nos fluxos, os temporizadores eram ajustados automaticamente. Para a solução encontrada, foi considerado um tempo base de 40 segundos que corresponderia ao tempo total de abertura dos semáforos de um cruzamento. Para o Cruzamento 1, por exemplo:

$$tverde_{A1} = tempobase.(\frac{F_{A1}}{(F_{A1} + F_{B1})}) \tag{11}$$

onde $tverde_{A1}$ é o tempo que o semáforo da Via A do cruzamento 1 ficará verde; F_{A1} é o fluxo de veículos da Via A que está chegando ao cruzamento 1 e F_{B1} é o fluxo da Via B que está chegando ao cruzamento 1.

Em consequência a isso, a tempo que o semáforo da via B para esse mesmo cruzamento deve permanecer aberto é dado por:

$$tverde_{B1} = tempobase - tverde_{A1} \tag{12}$$

Para que os semáforos dos cruzamentos estejam sincronizados, os tempos dos sinais vermelhos devem ser consequência da temporização dos sinais verdes e amarelos. Sendo assim, para a via A do cruzamento 1, por exemplo, temos:

$$tverm_{A1} = tverde_{B1} + tamar_{B1} \tag{13}$$

Enquanto as temporizações dos sinais verdes e vermelhos são diretamente dependentes do fluxo de veículos da via de acordo com as equações (11) e (13), sendo, portanto, variáveis; a temporização dos sinais amarelos é fixa e igual a 2 segundos. Isso porque este sinal serve apenas para garantir que os veículos parem até a sinalização vermelha sem que resulte em situação de perigo, como a que ocorreria no caso de uma parada abrupta ou uma frenagem insuficiente para que o veículo permanecesse parado antes do cruzamento. Dessa maneira, o programa foi feito de modo que houvesse um ajuste automático dos temporizadores em função dos fluxos das vias.

Finalizada a programação, as entradas e saídas físicas foram ligadas nas bancadas presentes no Laboratório de Mecatrônica, como a mostrada na Figura 7, e os programas foram transferidos aos CLPs. Portanto, para simulação dos quatro cruzamentos, os valores das entradas analógicas (potenciômetros simulando os fluxos de veículos) eram transmitidos a um CLP, responsável por recebê-los e executar a rotina de programação com tais números, retornando ao CLP de origem o valor booleano para acionamento das lâmpadas da respectiva bancada.

Figura 7: **Bancada utilizada na simulação**

Feitas as etapas para comunicação, foram criadas telas de supervisão com animações capazes de simular de maneira realística o tráfego de veículos e o funcionamento dos semáforos. Tais animações são feitas através de *frames* e controladas através das variáveis de memória provenientes do arquivo do *Unit Pro M*. Dessa maneira, foi necessário fazer pequenas modificações no programa, colocando as saídas físicas em paralelo com variáveis internas que seriam usadas no SCADA.

3. RESULTADOS

O funcionamento da Rede de Petri pode ser atestado, simulando-a no próprio *software HPSim*. Além disso, foram feitas verificações utilizando o *Model Check LTL*, em que é assegurada a sincronização semafórica dos cruzamentos. Por exemplo, constatando que o acionamento da sinalização verde dos semáforos de determinadas vias implica na sinalização vermelha dos semáforos das vias que as cruzam.

Esses testes foram feitos para as diversas combinações de acionamento dos semáforos, para que houvesse garantia de sincronização. Todas as verificações retornaram como resultado *TRUE*, indicando eficiência da rede de petri desenvolvida, atestando que quando uma via do cruzamento está com fluxo fluindo, na via transversal os veículos estão parados.

Dando sequência ao projeto, foi desenvolvida a solução local, a qual altera a temporização semafórica no cruzamento de acordo exclusivamente com os fluxos de veículos que estão chegando a ele.

Tal solução foi implementada nos CLPs com os potenciômetros das bancadas sendo utilizados para simular o fluxo de veículo em cada uma das vias. Assim, as lâmpadas contidas nelas representaram as luzes dos semáforos e pôde-se verificar o funcionamento do programa.

Entretanto, não foi de fácil percepção a sincronia dos semáforos de todos os cruzamentos e, então, foi desenvolvido um sistema SCADA de supervisão e controle para monitoramento em tempo real do fluxo de veículos e visualização do sistema como um todo com maior facilidade. As telas foram desenvolvidas mostrando o sistema com os quatro cruzamentos funcionando em concomitância (Figura 8) e cada um deles isoladamente.

Figura 8: **Simulação de tráfego em sistema supervisório**

É notável a facilidade de monitoramento com o sistema de supervisão. Além disso, consegue-se constatar falhas no sistema como possíveis bloqueios de vias por fila de carros na via transversal. Nessas ocorrências, é desejável que alarmes sejam acionados, alertando o operador de tais inconveniente.

Finalizadas as etapas de programação e monitoramento, o sistema foi modelado e simulado no *Matlab* usando a teoria de filas. Essa nova modelagem foi criada a fim de definir os tempos ótimos de abertura dos semáforos de modo a obter o menor valor possível para a função custo, isto é, uma mínima formaçao de filas nas vias. Para visualização do comportamento das filas e teste do modelo desenvolvido, foram plotados gráficos de diversas situaçoes criadas, como para uma variaçao na ponderaçao da funçao custo de alguma via, ou alteração do fluxo que chega ao cruzamento em cada cilo ou ainda mudança na capacidade de dispersão da via. Para cada contexto proposto, a temporizaçao semafórica foi ajustada automaticamente, buscando sempre a reduçao o não formação das filas nos sucessivos ciclos.

Como exemplo, são mostrados nas Figuras 9, 10 e 11 gráficos de acúmulo de veículos e do custo associado a cada temporização semafórica possível no ciclo.

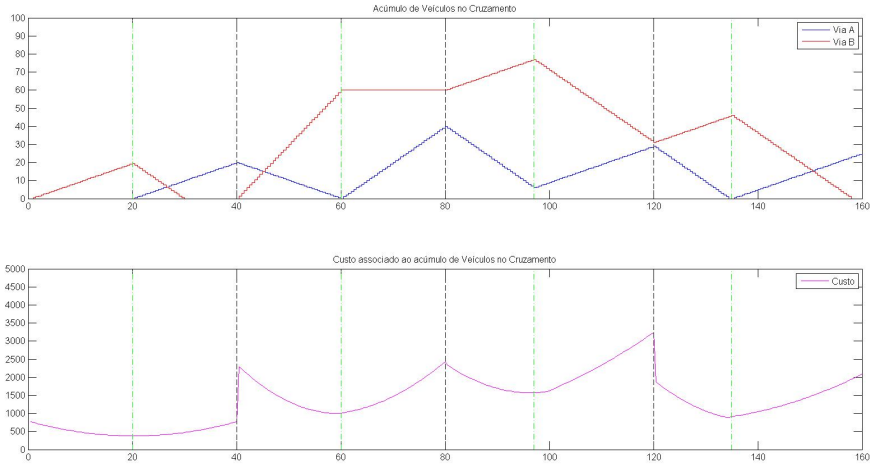

Figura 9: **Gráfico do comportamento dos veículos quando alterado o fluxo de entrada**

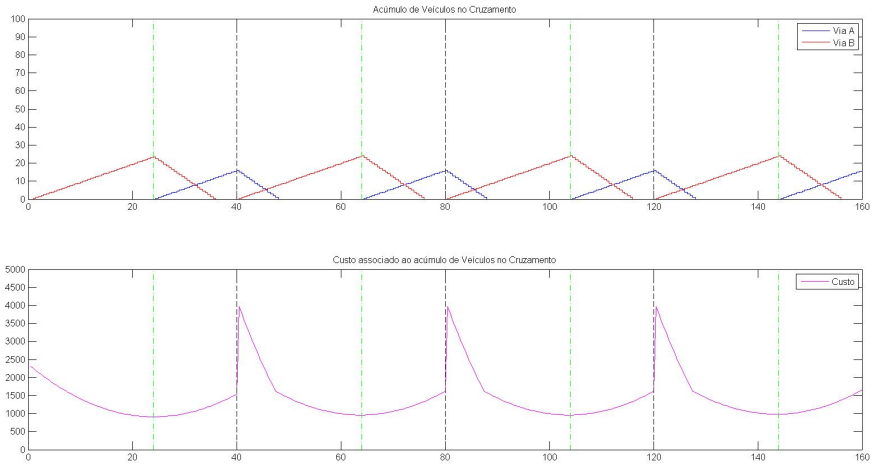

Figura 10: **Gráfico do comportamento dos veículos quando alterada a ponderação do custo**

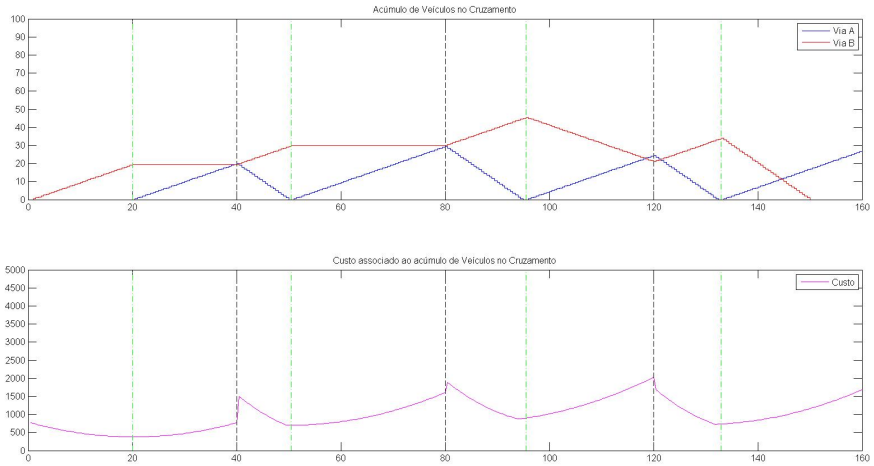

Figura 11: **Gráfico do comportamento dos veículos quando alterado o fluxo de dispersão**

Em cada um desses gráficos, as linhas tracejadas de cor preta representam o fim de um ciclo, ou seja, que os semáforos do cruzamento passaram pelo período aberto e pelo fechado; enquanto que as linhas tracejadas de cor verde representam o fim do tempo de verde de uma via (e, consequentemente, do tempo do sinal vermelho da outra). Com isso, fica evidente que os instantes onde encontramos o valor mínimo da função custo em cada ciclo correspondem exatamente ao tempo de verde utilizado. Assim, enquanto o sinal verde está aberto para a via A (condição ocorrendo à esquerda da linha tracejada verde), o fluxo de veículos que chega vai sendo dispersado. Nesse mesmo período, o semáforo da via B está fechado; portanto, todo o veículo que chega ao cruzamento aumenta a fila formada nesta via. Quando este semáforo abre, essa fila começa a ser dispersada e começa a haver formação de fila na via A1.

Ainda é possível notar ainda que essa simulaçao apresentada foi realizada para um total de 4 ciclos com $k = 40s$ e $passo = 0.5$. Logo, foi necessário escrever tanto os fluxos veiculares que chegam ao cruzamento quanto o fluxo que a via é capaz de dispersar como vetores linha de dimensão [1x4], onde cada elemento é referente a um ciclo.

Para a Figura 9, foram utilizados os seguintes parâmetros de teste: $p_A = p_B = 1$; $a_A = [1211].passo$; $a_B = [1311].passo$ e $c_A = c_B = [3333] * passo$. Nela, é possível notar que ao fim do segundo ciclo, há ainda fila de veículos remanescentes, mas nos ciclos posteriores a temporizaçao se adequa de modo a reduzir a fila tanto quanto possível até que toda a fila formada no ciclo consegue ser dispersa ao final deste.

No gráfico seguinte (Figura 10), os fluxos foram considerados constantes durante toda a simulação, com $a_A = a_B = [1111].passo$ e $c_A = c_B = [3333].passo$. Por outro lado, foi variada a ponderaçao do custo de cada via para $p_A = 2$ e $p_B = 3$. Dessa maneira, a temporizaçao varia de modo a dar prioridade à via B.

Por fim, nos gráficos da Figura 11, as ponderações foram definidas como iguais a 1, os fluxos que chegam ao cruzamento foram propostos como $a_A = a_B = [1111].passo$, todavia foi variado o fluxo disperso pela via B ao longo da simulação, de modo que $c_A = [3333].passo$ e $c_B = [1123].passo$. Nesse caso, há formação de filas pelo fato da capacidade de dispersão da via se tornar insuficiente durante os ciclos e, mais uma vez, a temporização permanece sendo modificada a cada ciclo, visando a minimização das filas.

Foi constatado, então, que ao avaliar-se o sistema como um todo, existem valores para a temporização dos semáforos que se diferem em alguns segundos da solução local obtida com a função linear e ao serem testados no sistema geram gráficos com áreas inferiores, portanto sendo uma melhor solução. Assim, conclui-se que para cruzamentos próximos o suficiente para os fluxos destes sejam influenciados uns pelos outros, a solução local implementada inicialmente nos CLPs é insuficiente, sendo necessário a análise conjunta dos semáforos.

Finalmente, utilizando o sistema SCADA é possível enviar de maneira automática as temporizações ótimas obtidas a partir da minimizaçao do custo aos CLPs, de modo a concretizar uma simulação realística do funcionamento do tráfego.

4. CONCLUSÕES

O presente trabalho mostrou a utilização eficiente de redes de petri para modelagem de sistemas de tráfego urbano, garantindo a sincronia deste. Além disso, a implementação das soluções nas bancadas de CLPs foi de extrema importância para a simulação do tráfego nos cruzamentos, permitindo uma melhor aproximação da situação real. A regulagem do fluxo das vias através dos potenciômetros tornou possível analisar diversas condições de fluxos, o que tornou o sistema mais realístico, sendo o programa capaz de adaptar-se a essas variações de modo automático.

O sistema SCADA foi utilizado para supervisionar todo o funcionamento do processo, tornando possível o controle deste de modo simples e de fácil visualização, através de animações representativas do tráfego real. Além disso, com a arquitetura de comunicação CLP-SCADA, é possível enviar os valores da temporização ideal a cada ciclo e monitorar o comportamento do tráfego.

A partir do modelo baseado na Teoria de filas, foi possível obter uma solução ótima para a temporização dos semáforos dos cruzamentos a cada ciclo: o critério de minimização das somas dos tamanhos das filas nas vias se mostrou eficaz, conseguindo definir os tempos de abertura dos semáforos de modo a reter o menor número de veículos possível.

5. REFERÊNCIAS

Boyer, S.A., 2010. *SCADA: Supervisory Control and Data Acquisition*. International Society of Automation, 4th edition.

Cassandras, C.G. and LAfortune, S., 2008. *Introduction to Discrete Event System*. Springer, 2nd edition.

David, R. and Alla, H., 2010. *Discrete, Continuous, and Hybrid Petri Nets*. Springer, 2nd edition.

D'Avila, E., 2011. "Controlador programvel: Aplicaes industriais". In *Automaao Industrial*. So Paulo.

Febbraro, A.D., Giglio, D. and Sacco, N., 2004. "Urban traffic control structure based on hybrid petri nets". *Intelligent Transportation Systems*, Vol. 5, pp. 224–237.

Foyo, P.M.G., 2009. *Verificao Formal de Sistemas Discretos Distribudos*. Ph.D. thesis, Escola Politcnica USP, So Paulo.

Gadelha, M.Y.R., Repolho, H.C.R., Almeida, N.R. and Jnior, E.T.P., 2010. "Modelagem de um sistema de trfego". In *INPA*.

Garcia, R.T., 2007. *Modelagem e controle da coordenao do trfego urbano atravs de formalismo para sistemas e eventos discretos e hbridos*. Ph.D. thesis, Universidade Federal de Santa Catarina, Santa Catarina.

List, F.G. and Cetin, M., 2004. "Modeling traffic signal control using petri nets". *IEEE Transactions on Intelligent Transportation Systems*, Vol. 5, pp. 177–187.

Moraes, F.G., Silva, F.G. and Rezende, T.A., 2011. "Introduo s teorias das filas". *Colquio de Matemtica da Regiao Centro-Oeste*, , No. 2.

Paravisi, M., Bicho, A.L. and Musse, S.R., 2006. "Modelagem e simulao do trfego de veculos e controle semafrico em um ambiente virtual". *Revista de Cincias Exatas e Engenharias*, Vol. 16, pp. 16–37.

Tolba, C., Lefebvre, D., Thomas, P. and Moudni, A.E., 2005. "Continuous and timed petri nets for the macroscopic and microscopic traffic flow modeling". *Simulation Modelling Practice and Theory*, , No. 13, pp. 407–436.

Valette, R. and Cardoso, J., 1997. "Redes de petri". *Proceedings of the 10th Brazilian Congress of Mechanical Engineering*, , No. 1, pp. 41–44.

6. RESPONSABILIDADE AUTORAIS

Os autores são os únicos responsáveis pelo conteúdo deste trabalho.

INTELLIGENT AUTOMATION SYSTEM DESIGN FOR URBAN TRAFFIC CONTROL

Érika Reinaldo Marques da Silva, erikarmsilva@gmail.com[1]
André Murilo, andremurilo@gmail.com[2]

[1]Universidade Federal de Pernambuco, Av. Prof. Moraes Rego, 1235 - Cidade Universitária
[2]Universidade de Brasília, Faculdade Gama, Asa Norte s/n

Abstract:*The problems of urban traffic are more and more present in big cities due fundamentally to a significant growth of vehicles in the last decades. Thus, the study of the techniques concerning traffic engineering has gained considerable importance in recent years in order to identify problems and find solutions for the flow of vehicles in the cities. In this paper, a simplified model of a discrete event system based on Timed Petri Nets was developed to model the traffic flow and control logic. In order to implement this control strategy, an automation system architecture based on Programmable Logic-Controllers (PLC) and a Supervisory Control and Data Acquisition (SCADA) was designed. This architecture consists in a set of PLCs communicating into an Ethernet TCP/IP network as well as the supervisory system SCADA for monitoring the process through a Human Machine Interface (HMI). The flow of vehicles was simulated by potentiometers connected in the analog input cards of the PLC. A PLC program based on Timed Petri Nets theory was developed in order to represent and optimize the sequences of signalized intersections. As a result, mathematical functions were established for timing all the traffic lights and reproduce an optimum flow condition from the automatic adjustment of the PLC timers. Experimental results obtained in this test bench showed that the resulting control scheme coordinates correctly the traffic lights, solves synchronization issues and optimizes considerably the system performance.*
Keywords: *Traffic control, Petri Nets, Simulation, PLC, SCADA.*