

Utilização do protocolo de comunicação *OLE for Process Control* em processos industriais

Use of communication protocol OLE for Process Control in industrial processes

Ivan Carlos Franco

Doutorando em Engenharia Química – Departamento de
Engenharia de Sistemas Químicos
Universidade Estadual de Campinas – Unicamp.
Campinas, SP [Brasil]
icfranco@feq.unicamp.br

Jones Erni Schmitz

Doutor em Engenharia Química – Departamento
de Ciências Exatas e da Terra;
Universidade Federal de São Paulo – Unifesp.
Diadema, SP [Brasil]
jones.schmitz@unifesp.br

Ana Maria Frattini Fileti

Doutora em Engenharia Química – Departamento de
Engenharia de Sistemas Químicos Universidade Estadual de
Campinas – Unicamp.
Campinas, SP [Brasil]
fratini@feq.unicamp.br

Flávio Vasconcelos da Silva

Doutor em Engenharia de Alimentos – Departamento de
Engenharia de Sistemas Químicos
Universidade Estadual de Campinas – Unicamp.
Campinas, SP [Brasil]
flavio@feq.unicamp.br

Neste artigo, propõe-se a utilização de um *software* matemático juntamente com o protocolo *OLE for Process Control* (OPC) para o desenvolvimento de sistemas de controle em processos industriais, transformando o *software* matemático em um sistema supervisor capaz não somente de monitorar o processo, mas também de desenvolver e implementar algoritmos de controle inteligente. Um estudo foi então realizado sobre a confiabilidade da comunicação OPC entre o *software* matemático, o Controlador Lógico Programável (CLP) e um sistema de refrigeração industrial. Nesta pesquisa, constatou-se que a comunicação é adequada para a aplicação, pois o *software* matemático utilizado na comunicação entre o CLP e o sistema de refrigeração apresentou boa confiabilidade referente à qualidade do sinal de comunicação, além de tempo real de comunicação. Conclui-se, portanto que essa forma de comunicação é uma potente ferramenta para o monitoramento, o desenvolvimento e a implementação de controladores avançados.

Palavras-chave: *OLE for Process Control*. *Software* matemático. Sistema de refrigeração. Controlador Lógico Programável. Controle e automação.

In this work it is proposed the use of a mathematical software and OPC (OLE for Process Control) protocol for development of control systems in industrial processes. The mathematical software was converted into a supervisory system capable of monitoring the process, and also implementing intelligent control algorithms. A study was then conducted on the reliability of OPC communication between the mathematical software, the Programmable Logic Controller (PLC) and the experimental system under study (industrial cooling system). This study showed that the OPC communication is suitable for this application because the mathematical software used in communication between the PLC and the cooling system showed good reliability on the communication signal quality, besides achieving real-time communication. As conclusion, this communication strategy showed to be a powerful tool for monitoring, developing and implementing advanced controllers.

Key words: OLE for Process Control. Mathematical software. Programmable Logic Controller. Control and automation.

1 Introdução

Para realizar o desenvolvimento de sistemas de controle inteligente, a fim de proporcionar uma melhor integração e facilitar a automação e o controle, deve-se pensar na melhor maneira de efetuar a comunicação entre os dispositivos envolvidos nesse controle. Uma forma é a utilização do padrão *Object Linking and Embedding (OLE) for Process Control (OPC)*, protocolo baseado no modelo de componentização criado pela Microsoft e denominado *Component Object Model (COM)*, que é uma maneira eficiente de estabelecer interfaces para aplicações e considerado hoje o protocolo padrão da indústria.

A vantagem mais acentuada em utilizar esse tipo de padrão de comunicação é a uniformidade de acesso aos *tag* (variáveis numéricas ou alfanuméricas envolvidas na aplicação que podem executar funções computacionais ou representar pontos de entrada/saída de dados do processo) com sistemas supervisórios sem a necessidade de utilizar *drives* proprietários de controladores. Outras vantagens desse padrão de comunicação são uniformidade de interface para diferentes redes e protocolos; integração plena com a rede e entre diferentes ferramentas de supervisão (CARDOSO; ZANAROTTI, 2005).

O controle de processo na indústria pode ser representado por três níveis distintos: 1) nível de campo, em que a presença de dispositivos inteligentes por redes *fieldbus* se torna cada vez mais comum; 2) nível de processo, no qual sistemas de controle distribuído (DCS) e os sistemas de supervisão e aquisição de dados (SCADA) disponibilizam dados, e 3) nível gerencial, em que se utilizam sistemas de bancos de dados e planilhas. A integração desses níveis é de grande interesse para a indústria. O procedimento usual, para possibilitar essa integração, é o desenvolvimento de *drivers* de comunicação entre diferentes sistemas.

Entretanto, esse procedimento não é ideal; pois, demandam-se grandes esforços para desenvolver *drivers* para diversos equipamentos e *softwares* (SOUZA et al., 1998).

O modelo de referência *Open Systems Interconnection (OSI)* determina como a parte física (*hardware*) e a parte lógica (*software*) se comunicam. Essa comunicação está baseada nas diferentes camadas existentes entre a parte física e a lógica do processo. Cada camada na hierarquia define um subconjunto de funções necessárias para o OSI. Sete camadas foram consideradas suficientes para dividir o sistema em lógicas similares e funções compreensíveis. Esse tipo de ordenação das camadas do modelo OSI descreve uma hierarquia de funções, (MELENDEZ; PETERSEN, 1999). As camadas são apresentadas hierarquicamente da mais baixa (camada física) para a mais alta (camada de aplicação), como mostra a Figura 1.

7. Aplicação Fornece a interface entre o protocolo de rede e o <i>software</i> em execução.
6. Apresentação Traduz os dados para o formato da rede.
5. Sessão Realiza a comunicação a ser estabelecida entre processos ou aplicações em execução.
4. Transporte Transporta os dados e perdas de dados de forma sequencial.
3. Rede Move os pacotes de dados entre os dispositivos por um roteamento.
2. Enlace de dados Possibilita a circulação de dados entre os dispositivos para permitir a comunicação.
1. Física Estabelece uma conexão física entre computadores para permitir a comunicação.

Figura 1: Hierarquia das camadas do modelo de referência OSI

Fonte: Baseada em informações do modelo de referência OSI.

A camada de aplicação é a que a maioria dos programas de rede usa para comunicar-se por meio de uma rede com outros programas. Processos que rodam nessa camada são específicos da aplicação;

monitorar e adicionar valores a esse servidor. Além do interesse de apresentar informação sobre a comunicação de um *software* matemático com um processo a ser controlado por meio do CLP/OPC há também o de utilizar um *software* matemático eficiente como um “Sistema Supervisório”.

1.1 Padrão OPC

O OPC é um padrão de comunicação de dados em que várias informações localizadas em diversos pontos (controladores, dispositivos de E/S, banco de dados, etc.) podem ser conectadas diretamente a um “Cliente”, o qual pode executar, entre outras, funções de Interface Homem-Máquina (IHM), de supervisão e de tratamento gráfico. A fim de realizar essa comunicação, o OPC utiliza-se de um protocolo universal para envio de dados entre “Servidores” (fontes de dados) e “Clientes” (aplicações), eliminando, dentro do possível, o uso de *drivers* e conversores, na maioria das vezes proprietários que criam uma dependência, muitas vezes, custosa e ineficiente, entre o usuário e o fornecedor do sistema proprietário.

Quando se utiliza de um padrão OPC, é necessário somente um *driver* padrão que se torna o Servidor (*Server*). Os dados são convertidos para uma forma que qualquer Cliente (*Client*) que tenha uma disposição OPC possa se conectar sem qualquer tipo de preparo prévio (Figura 2).

O uso de servidores OPC permite algumas vantagens em relação ao desenvolvimento de projetos em que se utilizam *drivers* (MACIEL, 2003). Algumas delas são:

- Uniformidade de interface para diferentes redes e protocolos, ou seja, dispensa a necessidade de ajustes dos *drivers* de cada rede, o que costuma levar tempo na configuração;
- Integração plena com a rede, mesmo quando alterações de protocolo forem implementadas o OPC será utilizado pelo fabricante;
- Eliminação da necessidade de *drivers* de comunicação;
- Integração entre diferentes ferramentas de supervisão.

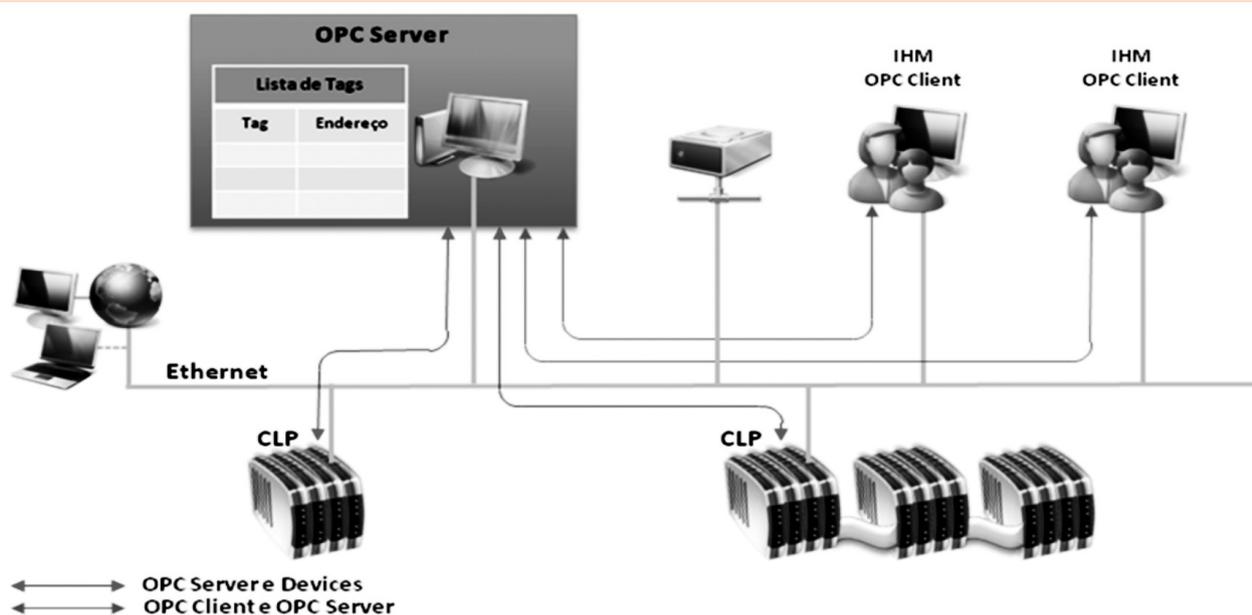


Figura 2: Acesso a dados de processos com protocolo OPC

Fonte: Os autores.

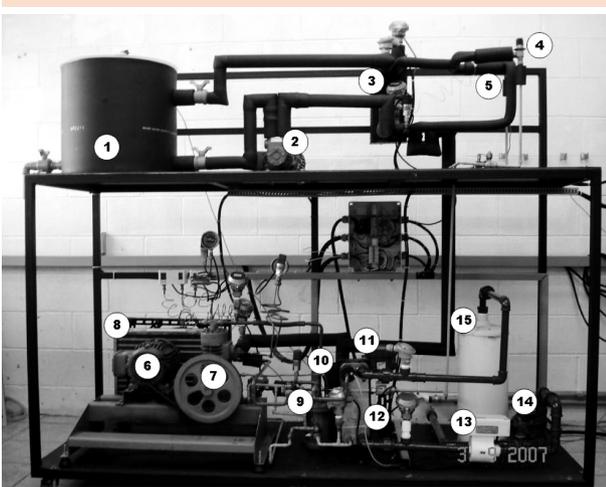


Figura 3: Módulo de refrigeração industrial em estudo (LCAP, 2007)

Fonte: Os autores.

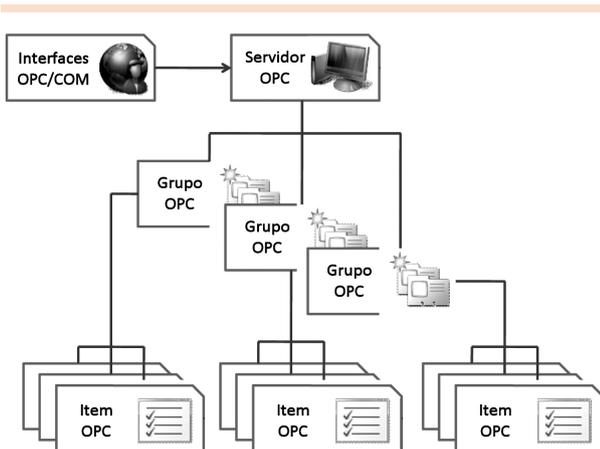


Figura 4: Hierarquia da criação dos grupos e itens da comunicação via protocolo OPC

Fonte: Os autores.

- Grupo OPC

O grupo é uma forma apropriada de organização dos dados utilizados no sistema de controle. Sua interface permite à aplicação: adicionar e remover itens do grupo; definir a taxa de leitura dos dados no grupo (cada grupo de dados pode ter uma taxa de leitura específica); ler e escrever valores para os itens do grupo. Neste trabalho, não foi gerado grupos contendo mais de uma variável em razão da presença de poucos *tags* de interesse.

- Itens OPC

Os itens são os próprios *tags* que são o objeto que proporciona uma conexão com uma entrada física de dados (Figura 5). Cada variável envolvida no estudo é tratada como um item do servidor OPC. Nesse caso, os itens (*tags*) são correspondentes ao sensor da temperatura de descarga da saída do compressor (TT_101); ao da temperatura de sucção do compressor (TT_106); ao da pressão de saída do evaporador (PT_103); ao da frequência do motor do compressor (FQ_101), e ao sensor da frequência da bomba de evaporação (FQ_201).

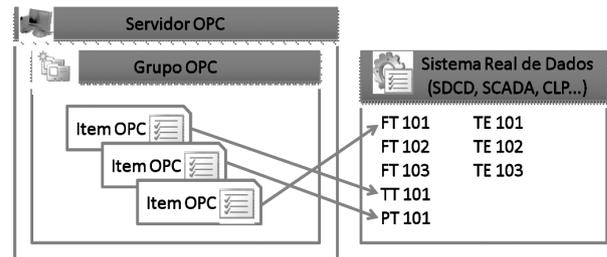


Figura 5: Mapeamento de itens OPC no sistema real de dados

Fonte: Os autores.

2.3 O sistema supervisório

As Figuras 6 e 7 representam a interface de comunicação para o sistema a ser controlado, utilizando o *software* em questão. A interface apresenta a configuração do servidor OPC e a criação dos grupos de “*tags*”, além de mostrar as saídas de dados em forma de gráficos “*real-time*”.

Na Figura 8, apresenta-se a configuração real do *tag* FQ_101 (frequência do compressor) que é um item “*Write OPC*” (*tag* de escrita) que tem como função ligar e desligar o compressor do sistema de refrigeração e a configuração real do *tag* POT_201 (potência do compressor), um item “*Read OPC*” (*tag* de leitura) que faz a leitura do valor da potência do compressor.

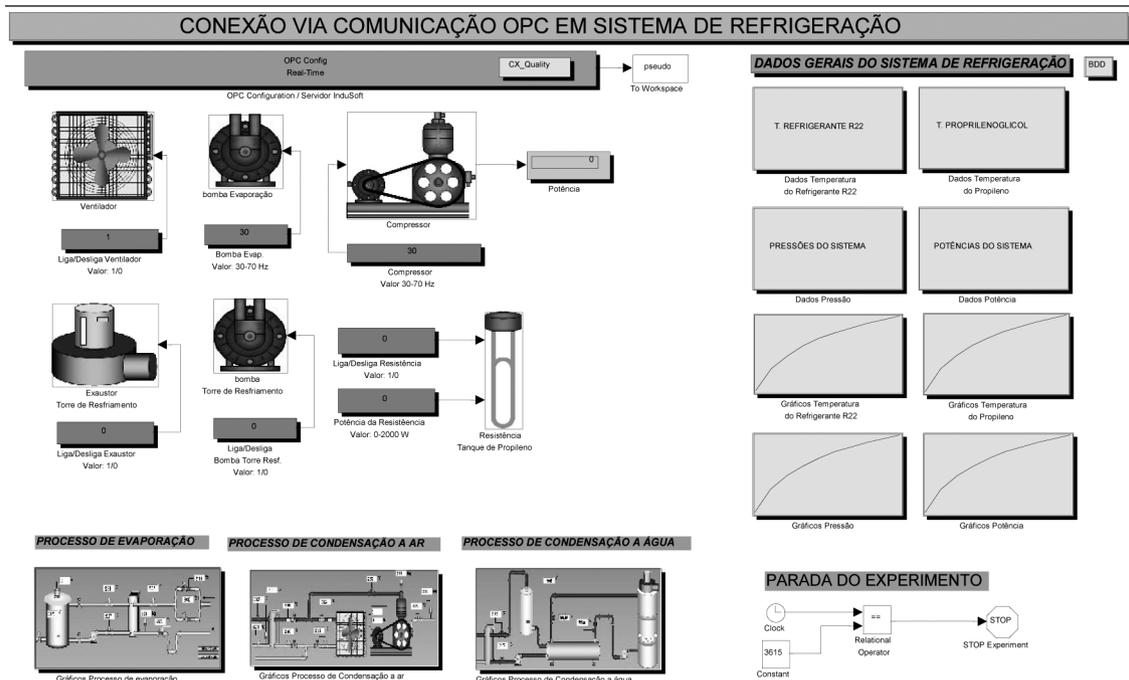


Figura 6: Tela principal do sistema supervisor desenvolvido em um *software* matemático e o protocolo de comunicação OPC

Fonte: Os autores.

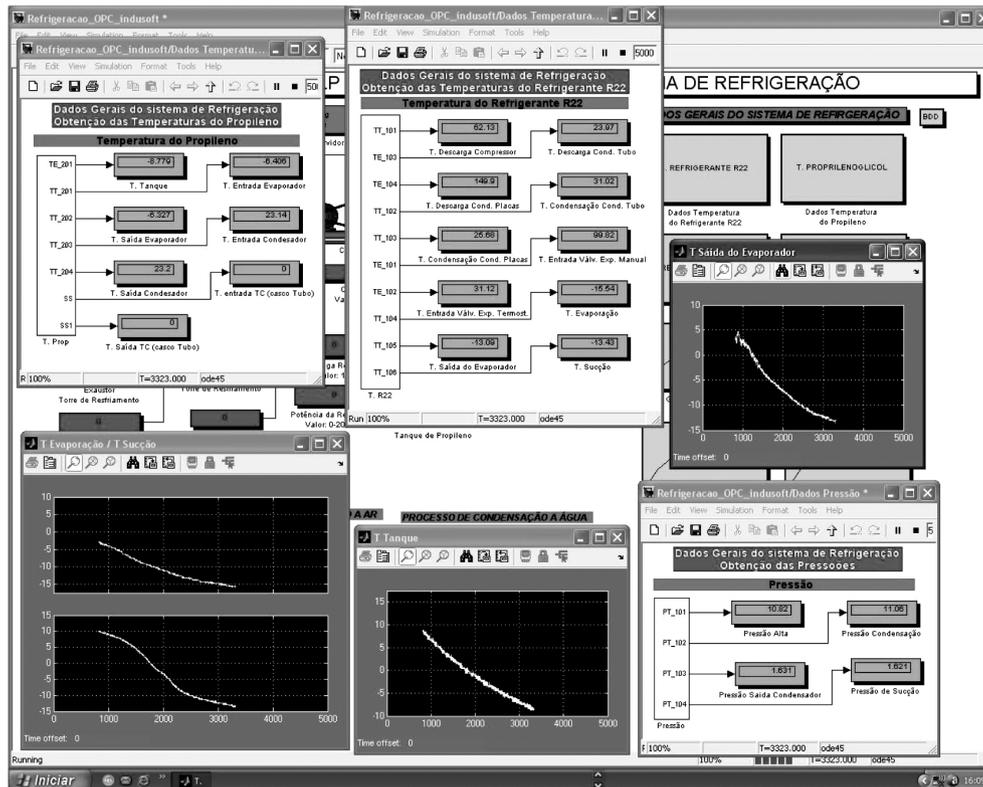


Figura 7: Telas de supervisão das temperaturas e pressões do sistema de refrigeração

Fonte: Os autores.

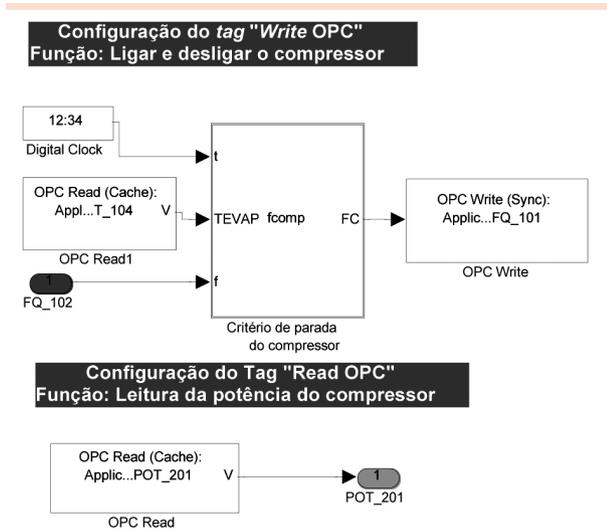


Figura 8: Configuração real dos tags de Escrita (Write OPC) e Leitura (Read OPC) para o compressor do sistema de refrigeração

Fonte: Os autores.

3 Resultados obtidos

A obtenção de dados em tempo real proveniente de sensores (temperatura, pressão, vazão, etc.), comandos de controle (entre outros abrir, fechar, ligar, desligar), *status* de comunicação, dados de desempenho e estatística do sistema foi realizado por meio da comunicação do MATLAB®/SIMULINK®-CLP, via padrão OPC.

Todos os sistemas supervisórios têm sua forma particular de apresentar a qualidade de conexão entre o CLP e o processo supervisionado. Muitos deles apresentam essa qualidade da forma BAD (quando não há sinal retornando para o CLP) e GOOD (quando há sinal retornando para o CLP). Ao se utilizar o *software* mencionado por meio da comunicação OPC, essa qualidade é apresentada por um conjunto de arranjos binários na forma QQSSSSL, em que QQ representa a qualidade principal do sinal; SSSS, o “substatus” da qualidade do sinal, e LL, o *status* do limite do sinal (MATLAB; SIMULINK, 2004-2007).

Nos resultados apresentados, o valor da qualidade é 192, o que equivale a “11000000” binários (QQ=11, SSSS=0000, LL=00), ou seja, a qualidade principal do sinal (QQ) é “11” binários o equivalente a 3 decimais, o substatus do sinal (SSSS) é “0000” equivalendo a 0 decimal e o *status* do limite do sinal (LL) é “00” o equivalente a 0 decimal. QQ=0=BAD (quando não há sinal retornando para o CLP), QQ=1=INCERTO (a razão da incerteza é indicada pelo substatus do sinal) e QQ=3=GOOD (quando há sinal retornando para o CLP).

Pode-se observar nas Figuras 9 a 13 a qualidade do sinal para o sensor TT_101 (temperatura de descarga da saída do compressor), TT_104 (temperatura de evaporação), TT_106 (temperatura de sucção do compressor), PT_103 (pressão da saída do evaporador), FQ_101 (frequência do motor do compressor) e FQ_201 (frequência da bomba de evaporação), respectivamente. Verifica-se que a qualidade apresentada é 192, o que representa QQ=GOOD, SSSS=NÃO ESPECIFICADO (Qualidade do sinal GOOD, porém não há condições especiais de *status*) e LL=SEM LIMITES, do começo ao fim do andamento do experimento.

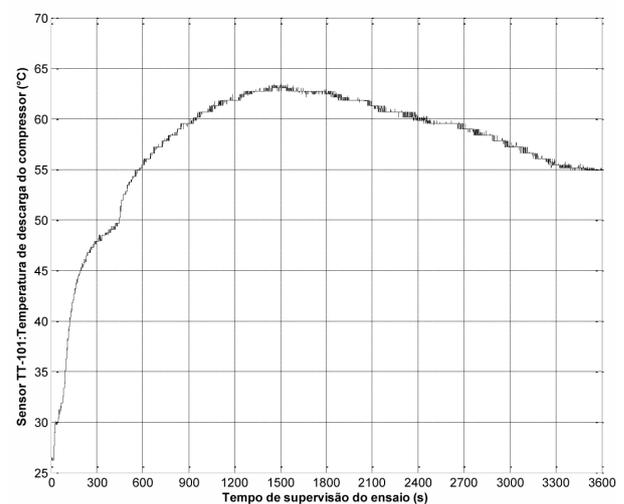


Figura 9: Qualidade do sinal para o sensor TT_101 (temperatura de descarga da saída do compressor)

Fonte: Os autores.

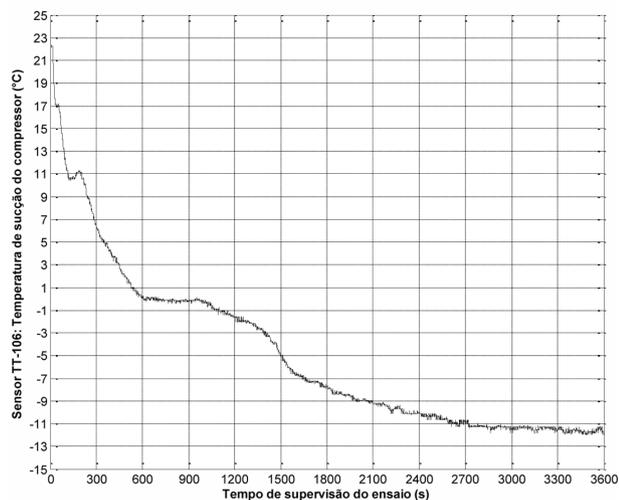


Figura 10: Qualidade do sinal do sensor TT_106 (temperatura de sucção do compressor)

Fonte: Os autores.

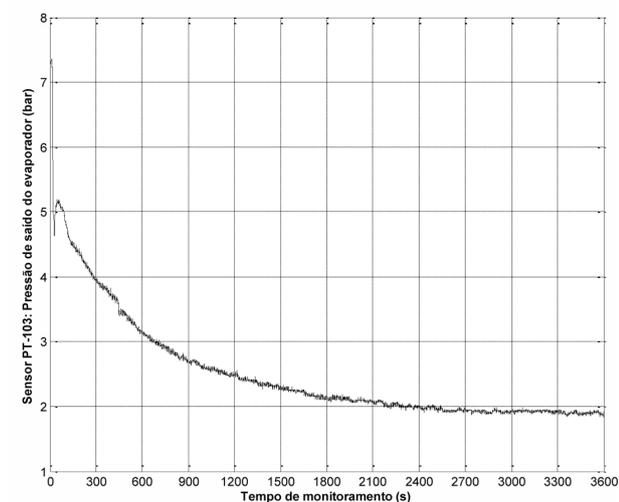


Figura 11: Qualidade do sinal do sensor PT_103 (pressão de saída do evaporador)

Fonte: Os autores.

Os *tags* relacionados com os sensores de temperatura (TT_101, TT_104 e TT_106) e pressão (PT_103) representam *tags* apenas de leitura (recebem informações) no sistema de refrigeração; na comunicação via protocolo OPC são denominadas “*read OPC itens*”. Já os *tags* relacionados com inversores de frequência (FQ_101 e FQ_201) representam *tags* de escrita (enviam informações) no sistema de refrigeração e para

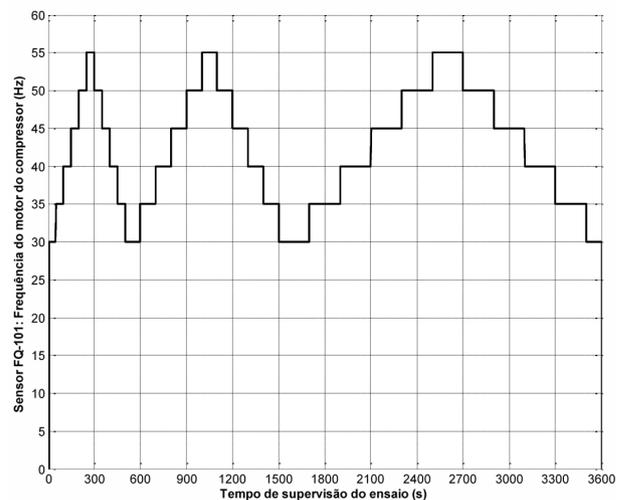


Figura 12: Qualidade do sinal do sensor FQ_101 (frequência do motor do compressor)

Fonte: Os autores.

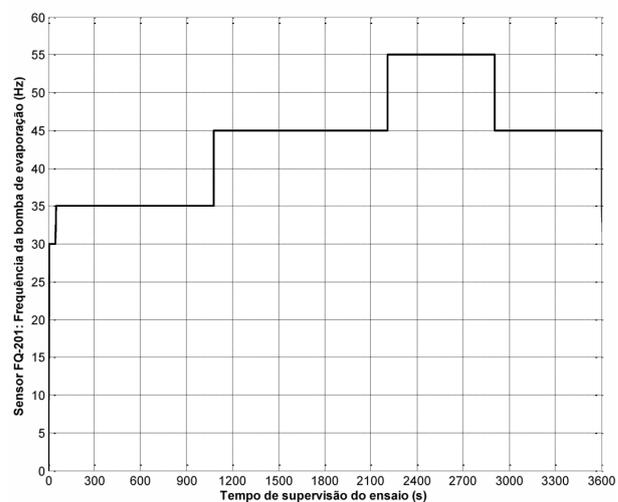


Figura 13: Qualidade do sinal do sensor FQ_201 (frequência da bomba de evaporação)

Fonte: Os autores.

o protocolo OPC são designados “*write OPC itens*”. Analisando as Figuras 9 a 13 verifica-se que tanto os *tags* de leitura como os de escrita apresentam qualidade no sinal.

Ao se trabalhar com sistemas de controle de processos químicos um dado importante é o tempo de resposta do sistema ao controle. Quando utilizados sistemas supervisórios convencionais e *drivers* proprietários de controladores, essa

resposta (baseando-se em resultados práticos de indústrias) é em tempo real. Entretanto, quando usado um *software* matemático para esse fim, deve-se fazer uma análise do tempo necessário para o *software* enviar a informação ao CLP e recebê-la de volta e se o tempo de “simulação” no experimento está compatível com o tempo de execução do *software*. O tempo de “simulação” é o tempo de execução do *software* matemático. Exemplo: para um experimento de 1 hora, ou 3600 segundos, o tempo de simulação será 3600 segundos no *software*.

No caso do *software* aplicado no estudo, o tempo necessário para ele enviar e receber as informações do CLP é fornecido pelo “Pseudo *Real-Time*”. Ele verifica o tempo de espera decorrido do disparo do pulso e a execução no sistema durante cada etapa. Caso o valor do “Pseudo *Real-Time*” venha a ser negativo, isso indica que a simulação está muito mais lenta do que o tempo real, com isso deveriam ser feitas algumas configurações na comunicação OPC, ou seja, configurar parâmetros do bloco OPC mais especificamente o *speedup* da simulação. Nas Figuras 14 e 15, pode-se verificar essa análise de tempo de envio e recebimento de informação entre o *software* e o CLP e o tempo real de execução.

Observando a Figura 14, nota-se que para um intervalo curto de tempo (30 segundos) a comunicação do *software* e o CLP, via OPC, tem valores negativos. O que indica que o tempo de simulação está muito mais lento do que o tempo real de execução, ou seja, há uma grande defasagem entre o envio e o recebimento de informações entre o CLP e o *software*. Contudo, esse curto intervalo de tempo pode ser considerado um tempo mínimo para que o CLP e o *software* estejam em sincronia, podendo ser desprezado. Ainda na Figura 14, observa-se que a partir de 30 segundos essa defasagem não existe apresentando um “Pseudo *Real-Time*” médio de 0,35

segundos, ou seja, o tempo para o envio e o recebimento de informações entre o *software* e o CLP é de 350 milissegundos fazendo com que o tempo de simulação seja praticamente o tempo real de execução. A Figura 15 confirma essa rapidez de troca de informações entre o *software* e o CLP.

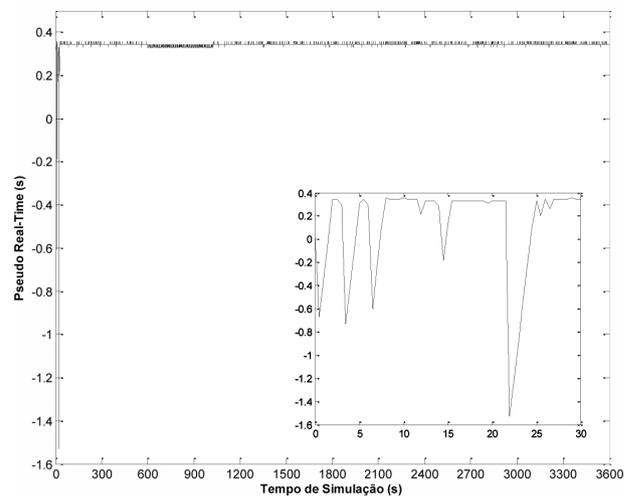


Figura 14: Pseudo *Real-Time*: tempo de envio e recebimento de informações do *software* e CLP, utilizando protocolo de comunicação OPC

Fonte: Os autores.

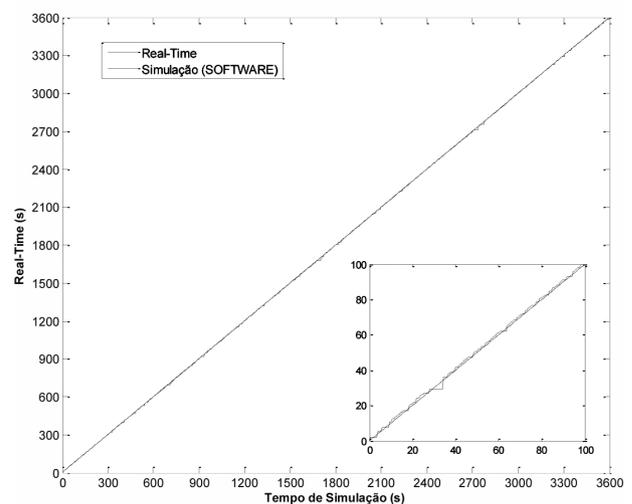


Figura 15: *Real-Time*: tempo real de execução do *software* e a comunicação com o CLP via protocolo OPC

Fonte: Os autores.

Conclusão

Neste trabalho, descreveu-se a utilização de um *software* matemático juntamente com o padrão OPC para o desenvolvimento de sistemas de controle em processos industriais. Os testes realizados, enfatizando a comunicação do programa, foram satisfatórios; pois, verificou-se que a comunicação entre o *software* e o processo apresentou uma qualidade de sinal de comunicação muito boa, segundo os critérios de avaliação de qualidade do programa.

Em relação ao tempo necessário para que o *software* envie informações ao CLP e a receba de volta, ou seja, o “Pseudo *Real-Time*”, conclui-se que o tempo de espera decorrido entre o disparo do pulso e a execução no sistema, durante cada etapa, é praticamente o tempo real de execução.

Assim, o programa e o CLP, por meio do padrão OPC, fornecem uma troca de informações muito rápida, podendo ser utilizados como um “sistema supervisor”. Contudo, não um simples “supervisor”, mas sim uma potente ferramenta de monitoramento e desenvolvimento de controles avançados. Uma aplicação futura é a utilização desse tipo de protocolo OPC em conjunto com o MATLAB® para o desenvolvimento de controles, utilizando técnicas de redes neurais artificiais (RNA), *fuzzy* e *neuro-fuzzy*.

Referências

BOFF, S. G. et al. Aplicação de um SCADA a uma unidade experimental de coluna de destilação. In: 3º CONGRESSO BRASILEIRO DE P&D EM PETRÓLEO E GÁS, 2005, Salvador, BA. *Anais...* Salvador: Editora Instituto Brasileiro de Petróleo e Gás – IBP, 2005.

CARDOSO, M. C. F.; ZANAROTTI, G. S. CLP – Evolução e tendências. *Mecatrônica Atual – Automação Industrial de Processos e Manufatura*. Ano 4, n. 25, p. 17-21, 2005.

FONSECA, M. O. Comunicação OPC – uma abordagem prática. In: VI SEMINÁRIO DE AUTOMAÇÃO DE PROCESSOS, ASSOCIAÇÃO BRASILEIRA DE METALURGIA E MATERIAIS, 2002, Vitória, ES. *Anais...* Vitória, ES, 2002, p. 1-12.

HONG, X.; JIANHUA, W. Using standard components in automation industry: a study on OPC specification. *Computer Standards & Interfaces*. v. 28, p. 386-395, 2006.

HUANG, Q.; SHE, Q.; LIN, X. Adaptive fuzzy PID temperature control system based on OPC and modbus/TCP protocol. In: 2ND INTERNATIONAL ASIA CONFERENCE ON INFORMATION IN CONTROL, AUTOMATION AND ROBOTICS, 2010, China. *Proceedings...* China: IEEE, 2010.

LIEPING, Z.; AIQUN.; YUNSHENG, Z. On remote real-time communication between MATLAB and PLC based on OPC technology. In: 26TH CHINESE CONTROL CONFERENCE – IEEE, 2007, China. *Proceedings...* China: IEEE, 2007.

MACIEL, P. H. S. Configuração de OPC client no elipse E3, 2003. Disponível em: <<http://kb.elipse.com.br/pt-br/questions/30/Configura%C3%A7%C3%A3o+de+OPC+Client+no+Elipse+E3>>. Acesso em: 3 dez. 2010.

MATLAB & SIMULINK. OPC Toolbox™ 2 – user’s guide. *The MathWorks™*, 2004-2007.

MELLENDEZ, W. A.; PETERSEN, E. L. The upper layers of the ISO OSI reference model (Part II). *Computer Standards & Interfaces*, v. 20, p. 185-199, 1999.

SOUZA, L. C. A.; SEIXAS FILHO, C.; PENA, R. T. Padrão de acesso a dados OPC e sua implementação em um driver OPC-Modbus. In: II CONGRESSO MINEIRO DE AUTOMAÇÃO, V SIMPÓSIO REGIONAL DE INSTRUMENTAÇÃO DA ISA-BH / GRINST-MG, 1998, Belo Horizonte. *Livro de Anais...* Belo Horizonte, 1998. p. 157-164.

THALAKOTUNA, D.N.P. et al. Remote monitoring and controlling system for a water purification plant. In: INFORMATION AND AUTOMATION FOR SUSTAINABILITY, 2008, Sri Lanka. *Proceeding...* Sri Lanka: IEEE, 2008.

YAN, R.; LU, L. Research of remote real time monitor system for electromechanical equipments based on internet. In: IEEE INTERNATIONAL CONFERENCE ON AUTOMATION AND LOGISTICS, 2007, China. *Proceedings...* China: IEEE, 2007.

Recebido em 18 out. 2010 / aprovado em 12 dez. 2010

Para referenciar este texto

FRANCO, I. C. et al. Utilização do protocolo de comunicação OLE for Process Control em processos industriais. *Exacta*, São Paulo, v. 8, n. 3, p. 319-329, 2010.

