

UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE ENGENHARIA ELÉTRICA

GISELE PINHEIRO DA SILVA

SUPERVISÃO DE PROCESSOS VIA OPC

VIÇOSA - MG
2012

UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE ENGENHARIA ELÉTRICA

GISELE PINHEIRO DA SILVA

SUPERVISÃO DE PROCESSOS VIA OPC

VIÇOSA - MG
2012

GISELE PINHEIRO DA SILVA

SUPERVISÃO DE PROCESSOS VIA OPC

Trabalho de conclusão de curso submetido à Universidade Federal de Viçosa para obtenção dos créditos referentes à disciplina Monografia e Seminário do curso de Engenharia Elétrica.

Orientador: Prof. Dr. André Gomes Torres

VIÇOSA - MG
2012

GISELE PINHEIRO DA SILVA

SUPERVISÃO DE PROCESSOS VIA OPC

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 18 de Junho de 2012

COMISSÃO EXAMINADORA

Prof. Dr. André Gomes Torres – Orientador
Universidade Federal de Viçosa

*“Não tentes ser um homem bem sucedido, tenta antes ser um homem de valor.”
(Albert Einstein)*

DEDICATÓRIA

Dedico este trabalho aos meus pais, amigos e ao Felipe.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela vida concedida cheia de oportunidades e privilégios.

Agradeço também aos meus pais, Alberto e Denise, à minha irmã, Natália, ao meu namorado Felipe, pelo carinho, amor, força e compreensão de sempre.

Agradeço também ao meu orientador André e co-orientadora Kétia, pelo conhecimento compartilhado e pela paciência durante o desenvolvimento do trabalho.

Enfim, agradeço a todas as pessoas que contribuíram de alguma forma para meu sucesso e para meu crescimento como pessoa. Sou o resultado da confiança e da força de cada um de vocês. Muito obrigada.

RESUMO

SUPERVISÃO DE PROCESSOS VIA OPC

O presente trabalho tem como objetivo a implementação de uma rotina no software MATLAB através de seu cliente OPC (OLE for Process Control) para comunicar com os equipamentos da SMAR localizados no Laboratório de Automação Industrial do Departamento de Engenharia Elétrica da Universidade Federal de Viçosa, realizando assim o controle e supervisão de processos. O OPC é um servidor através do qual vários programas de configuração, controle e supervisão conseguem comunicar variáveis entre si e com equipamentos físicos do processo. Visando tal objetivo fez-se necessário configurar um processo físico utilizando o System302 da Smar, implementar uma rotina cliente OPC no MATLAB e implementar exemplos de rotinas de controle avançados para assim promover a comunicação cliente OPC no MATLAB com os elementos de campo.

Palavras-chave: OPC, MATLAB, Planta Didática da SMAR, comunicação cliente OPC.

ABSTRACT

PROCESSES SUPERVISION BY OPC

This work aims to implement a routine in the software MATLAB through its client OPC (OLE for Process Control) to communicate with the SMAR' equipment located in the Industrial Automation Laboratory, Department of Electrical Engineering, Federal University of Viçosa, thus performing control and supervision of processes. The OPC is a server through which programs of configuration, control and supervision can communicate varying among themselves and with physical equipment the process. To achieve this objective it was necessary to configure a physical process using System302 of SMAR, to implement a routine client OPC and to implement examples of the advanced control routines so as to promote communication client OPC in MATLAB with the field elements.

Keywords: OPC, MATLAB, Didactic Plant SMAR, communication client OPC.

SUMÁRIO

DEDICATÓRIA.....	7
AGRADECIMENTOS.....	8
RESUMO.....	9
ABSTRACT.....	10
1. INTRODUÇÃO	12
1.1 – OBJETIVO.....	14
1.2 – DISTRIBUIÇÃO DO TRABALHO.....	14
2. REVISÃO BIBLIOGRÁFICA.....	15
2.1 – OPC.....	15
2.2 – PLC.....	16
2.3 – PID.....	17
2.4 – Lógica Fuzzy.....	18
3. METODOLOGIA.....	19
3.1 – Comunicação entre OPC, MATLAB e Panta Didática.....	19
3.1.1 – Servidor.....	19
3.1.2 – Cliente.....	20
3.1.3 – Exportação de Tag.....	20
3.2 – Controlador PI.....	20
3.2.1 – Controlador PI sem realimentação de ganho.....	20
3.2.2 – Controlador PI com realimentação de ganho.....	21
3.3 – Lógica Fuzzy.....	22
4. RESULTADOS OBTIDOS.....	25
5. CONCLUSÕES.....	31
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	32

CAPÍTULO 01

1. INTRODUÇÃO

Uma atividade muito comum e extremamente importante para a ciência e a indústria atuais é o processo de supervisão e controle de processos. Através de programas de supervisão, é possível obter gráficos e atuar no processo com recursos mais sofisticados.

O controle de processos é a base fundamental da automação industrial. Desde o processo mais simples até os processos de produção contínua, como em uma planta petroquímica, são analisados e desenvolvidos através dos elementos de controle.

Em um sistema controlado, como o representado na Figura 1, deseja-se manter uma grandeza (temperatura, nível, pressão, vazão, velocidade, posição, etc.) em um determinado valor. Para manter essa grandeza sob controle algumas informações são necessárias:

- Valor desejado – Set Point (SP);
- Valor do processo – Process Value (PV);
- Algoritmo de controle.

Com base nessas informações, o controlador compara o valor desejado (SP) com o valor do processo (PV) e determina, com base no algoritmo de controle, o valor de correção na saída para que o valor do processo se aproxime do valor desejado.

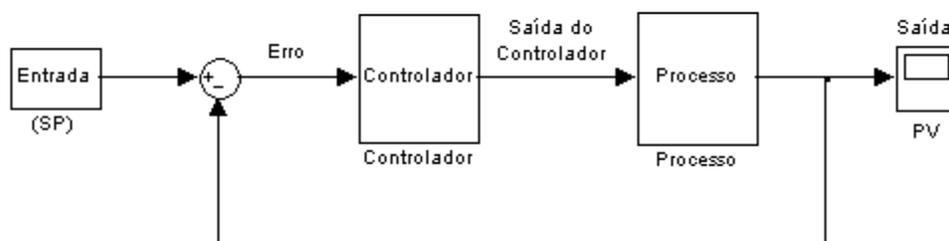


Figura 1- Diagrama de Bloco de um sistema controlado.

Existem diversos algoritmos de controle utilizados em diversos tipos de processo, cada um com sua característica, onde tais algoritmos podem operar individualmente ou juntos. Porém, os algoritmos mais utilizados são:

- Controle on-off;
- Controle PID;
- Controles avançados.

O controle on-off é também conhecido como controle de “liga e desliga”. O sinal de saída tem apenas duas posições que vão de um extremo ao outro, podendo ser: válvula aberta ou válvula fechada, resistência ligada ou resistência desligada, por exemplo.

O uso de controle on-off é ideal em aplicações onde a variável a ser controlada possui um tempo de resposta lento, como estufas, ar-condicionado, ferro de passar roupas, entre outros.

O controlador PID é uma operação em conjunto do controle proporcional (P), do controle integral (I) e do controle derivativo (D).

O controle proporcional produz em sua saída um sinal de controle que é proporcional ao erro, ou seja, quanto maior o erro maior será a ação corretiva produzida pelo controlador. Esta proporcionalidade de ganho é representada por uma constante K_p que define o ganho do controlador.

O controle integral produz em sua saída um sinal proporcional ao somatório do sinal de erro em um determinado instante de tempo, ou seja, a integral do erro. Este tipo de controlador minimiza os erros de regime permanente do sistema, entretanto sua ação de resposta é lenta. Nesta estratégia deve-se definir o valor do ganho integral através da constante K_i .

O controle derivativo produz em sua saída um sinal proporcional a taxa de variação do sinal de erro, ou seja, a derivada do erro. Este tipo de controlador proporciona uma ação bastante rápida sempre que ocorrerem variações na saída, entretanto não minimiza os erros em regime permanente. Nesta estratégia deve-se definir o valor do ganho derivativo através da constante K_d .

Os controles avançados são os Neurais, Fuzzy, adaptativos, dentre outros, estes controles são mais complexos em suas concepções e, portanto mais complexos em seus algoritmos precisando mais recursos computacionais para serem implementados.

A importância de se supervisionar sistemas reais se dá no âmbito de que as variáveis obtidas podem ser usadas para proporcionar um melhor conhecimento sobre tal sistema. Os programas de supervisão do ambiente industrial possuem blocos de funções limitados, onde o principal objetivo destes é atender aos consumidores com simplicidade e confiabilidade.

Este trabalho propõe implementar uma rotina no MATLAB através de seu cliente OPC para comunicar com os equipamentos da SMAR localizados no Laboratório de Automação Industrial do Departamento de Engenharia Elétrica da Universidade Federal de Viçosa, realizando assim o controle e supervisão de processos.

1.1. Objetivo

O presente trabalho tem como objetivo implementar uma rotina no MATLAB através de seu cliente OPC para comunicar com os equipamentos da SMAR localizados no Laboratório de Automação Industrial do Departamento de Engenharia Elétrica da Universidade Federal de Viçosa, realizando assim o controle e supervisão de processos.

Para atingir a meta principal, outros objetivos específicos devem ser salientados:

- Configurar um processo físico utilizando o System302 da Smar;
- Implementar uma rotina cliente OPC no MATLAB;
- Promover a comunicação cliente OPC no MATLAB com os equipamentos de campo;
- Implementar exemplos de rotinas de controle avançados.

1.2. Disposição do Trabalho

O presente trabalho está disposto como se segue:

No Capítulo 01, foi explanada uma breve introdução sobre o tema a ser abordado no trabalho.

O Capítulo 02 apresenta a revisão bibliográfica do trabalho explicitando conceitos que devem-se ter em mente a fim de se obter uma melhor compreensão de toda metodologia utilizada para a realização deste trabalho.

O Capítulo 03 apresenta a metodologia em que se realizou a supervisão do sistema térmico do tanque de água quente da Planta Didática SMAR via OPC.

O Capítulo 04 refere-se à exposição dos resultados obtidos.

O Capítulo 05 refere-se às conclusões que podem ser obtidas perante os resultados apresentados.

O Capítulo 06 apresenta as referências bibliográficas utilizadas para a realização deste trabalho.

2. REVISÃO BIBLIOGRÁFICA

Algumas informações são necessárias para uma maior compreensão deste relatório. Assim, nas próximas seções, algumas definições serão elucidadas, com o objetivo de proporcionar maior compreensão da metodologia, execução e resultados do projeto.

2.1. OPC

OPC (OLE for Process Control) é um padrão industrial publicado para interconectividade de sistema. A interoperabilidade é assegurada através da criação e manutenção de especificações de padrões abertos [10]. Inúmeras são as vantagens de se utilizar o OPC como protocolo de comunicação, entre elas estão:

- Padronização das interfaces de comunicação entre os servidores e clientes de dados de tempo real, facilitando a integração e manutenção dos sistemas;
- Eliminação da necessidade de drivers de comunicação específicos (proprietários);
- Melhoria do desempenho e otimização da comunicação entre dispositivos de automação;
- Interoperabilidade entre sistemas de gestão empresarial, de execução de manufatura e aplicações Windows.
- Redução dos custos e do tempo para desenvolvimento de interfaces e drivers de comunicação, com consequente redução do custo de integração de sistemas;
- Facilidade de desenvolvimento e manutenção de sistemas e produtos para comunicação em tempo real;
- Facilidade de treinamento.

As plataformas industriais de programação e supervisão de processos utilizam servidores de dados para gerenciamento das informações. Como exemplo destes servidores, existe o OPC (OLE for Process Control), onde através deste, vários programas de configuração, controle e supervisão conseguem comunicar variáveis entre si e com os equipamentos físicos do processo.

A arquitetura, descrita na Figura 2 abaixo, exhibe um exemplo de um sistema compreendido por equipamentos do fabricante SMAR. Neste exemplo os sensores e atuadores comunicam com os controladores lógicos programáveis (PLCs) utilizando diversos protocolos de redes, como a FieldBus, ProfiBus, DeviceNet, Hart. Os CLPs executam as suas funções de controle e além destas servem de ponte (Bridge) entre as redes de campos descritas e a rede HSE (High Speed Ethernet). Conectado na HSE, torna-se necessário um computador (Host) executando um processo servidor de base de dados, neste caso o OPC. O OPC irá disponibilizar as informações de campo

para todos os clientes OPC conectados a ele. Os clientes OPCs podem ser processos hospedados no mesmo computador ou em outro conectado pela rede Ethernet, independente da distância física dentre os mesmos.

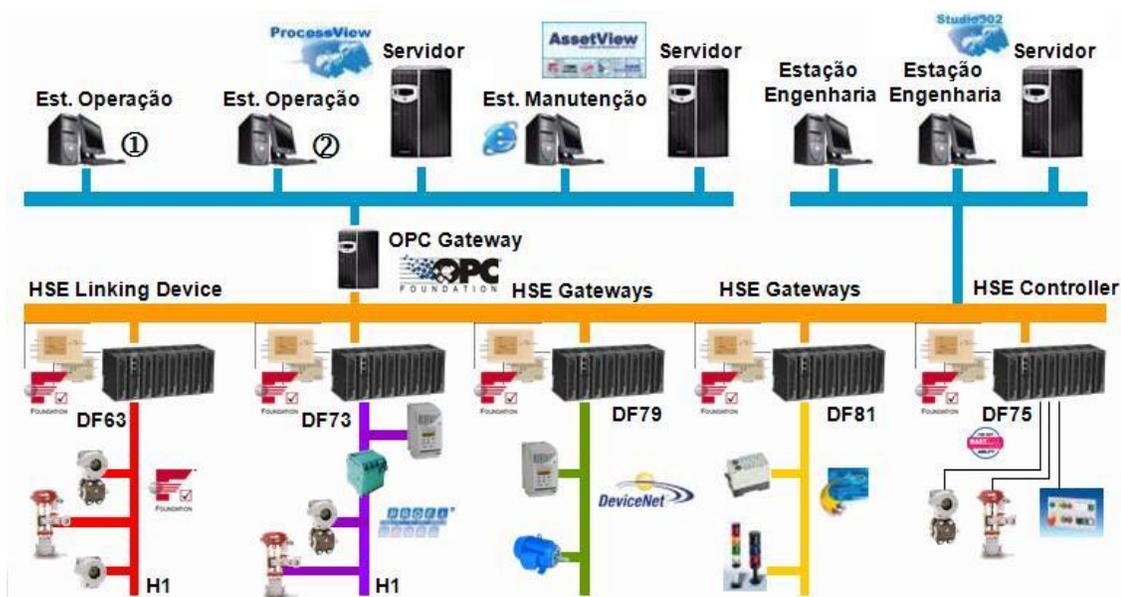


Figura 2 - Arquitetura de um sistema industrial

A SMAR utiliza para configuração de seus equipamentos a plataforma SYSTEM302 e o ProcessView para supervisão, sendo estes clientes OPC. Os recursos de controle destes clientes são voltados para aplicações industriais possuindo confiabilidade e simplicidade, mas é limitado para utilizações científicas de pesquisa.

Uma solução para contornar este problema. É utilizar o MATLAB como cliente OPC através do Toolbox específico para este fim. Com isto torna-se possível utilizar todos os recursos de cálculos complexos do MATLAB e aplicá-los no processo físico.

2.2. PLC

PLCs (programmable logic controller) são considerados a espinha dorsal da maioria projetos de automação em controle de processos industriais e automação.

Os CLPs (Controlador lógico programável) são frequentemente definidos como microcomputadores industriais que contém um hardware e um software que são utilizados para realizar as funções de controle. Um CLP consiste em duas seções básicas: a unidade central de processamento (CPU) e a interface de entradas e saídas do sistema [11].

Um diagrama de lógica é uma interface gráfica de linguagem de programação que é usada para desenvolver algoritmos para PLCs. Ele permite ao desenvolvedor escrever programas usando símbolos gráficos em vez de texto, como é o caso de linguagens de programação como Java, C / C ++ etc. Estes símbolos gráficos são dispostos sequencialmente no diagrama formando uma rede de dispositivos que

compõem o sistema. Como a maioria das linguagens gráficas, a Lógica Ladder, um exemplo de linguagem gráfica, muito usada, tem sua própria gramática e regras de sintaxe que ditam como seus elementos de linguagem devem ser arranjados [11].

A Figura 3 abaixo apresenta dois diagramas de blocos representando interação de um PLC com uma planta e o seu diagrama interno.

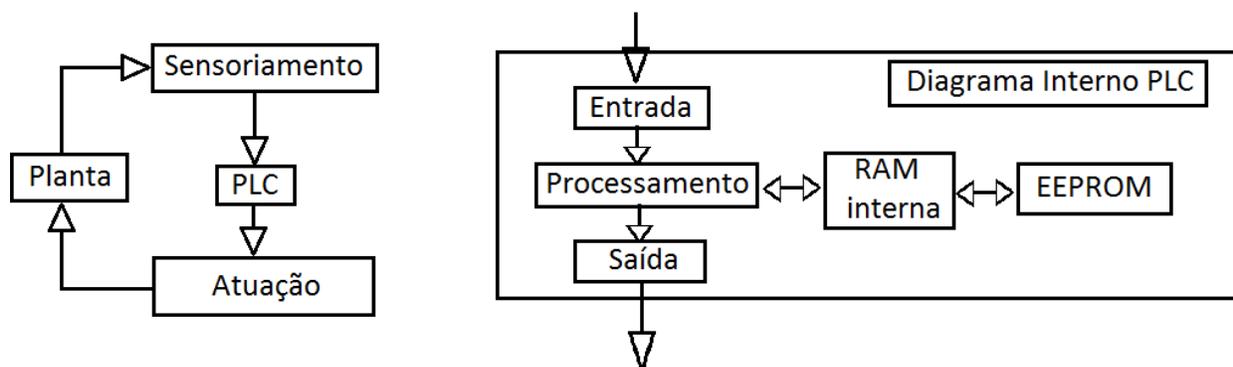


Figura 3 – Diagramas de blocos.

De forma geral, os dados são adquiridos pelos sensores e enviados como entradas para o PLC onde são processadas.

Seu funcionamento interno consiste em um processo cíclico de leitura das entradas que são armazenadas como variáveis na memória RAM. Estas variáveis são processadas segundo o software desenvolvido pelo próprio usuário seguindo as prioridades nele definidas. Logo após, as variáveis são levadas aos dispositivos de saída que atuam diretamente na planta realizando assim, o controle.

2.3. PID

O controlador PID é uma operação em conjunto do controle proporcional (P), do controle integral (I) e do controle derivativo (D).

O controle proporcional produz em sua saída um sinal de controle que é proporcional ao erro, ou seja, quanto maior o erro maior será a ação corretiva produzida pelo controlador. Esta proporcionalidade de ganho é representada por uma constante K_p que define o ganho do controlador.

O controle integral produz em sua saída um sinal proporcional ao somatório do sinal de erro em um determinado instante de tempo, ou seja, a integral do erro. Este tipo de controlador minimiza os erros de regime permanente do sistema, entretanto sua ação de resposta é lenta. Nesta estratégia deve-se definir o valor do ganho integral através da constante K_i .

O controle derivativo produz em sua saída um sinal proporcional a taxa de variação do sinal de erro, ou seja, a derivada do erro. Este tipo de controlador proporciona uma ação bastante rápida sempre que ocorrerem variações na saída, entretanto não minimiza os erros em regime permanente. Nesta estratégia deve-se definir o valor do ganho derivativo através da constante K_d .

2.4. Lógica Fuzzy

A Lógica Fuzzy ou Lógica Difusa diferente da Lógica Clássica, que apenas permite a classificação de “Verdadeiro” ou “Falso”, é capaz de atribuir valores lógicos intermediários. Trabalhar em uma lógica que permite classificar dados ou informações vagas, imprecisas e ambíguas, abre muitas possibilidades de desenvolver soluções para problemas que envolvem muitas variáveis. [8]

O controle executado pela lógica Fuzzy imita um comportamento baseado em regras ao invés de um controle explicitamente restrito a modelos matemáticos como equações diferenciais. O objetivo da lógica Fuzzy é gerar uma saída lógica a partir de um conjunto de entradas não precisas, com ruídos ou até mesmo faltantes [9].

A lógica Fuzzy apresenta as seguintes características em relação a outras técnicas de controle:[8]

- Robusta porque não requer entradas precisas.
- Modificada facilmente pois é baseada em regras.
- Controle de sistemas não-lineares sem modelo matemático.
- Solução mais rápida e barata em alguns casos.
- Implementável facilmente em microprocessadores.

A utilização da Lógica Fuzzy em áreas de tomada de decisão proporciona o desenvolvimento de ferramentas heurísticas melhores para o homem, facilitando tomadas de decisão de forma mais ágil e eficaz.[8].

3. METODOLOGIA

3.1. Comunicação entre OPC, MATLAB e Planta Didática.

3.1.1. Servidor

A planta didática tem a função de “obedecer” os comandos vindos do cliente. O cliente deve acessar os blocos do servidor, no entanto deve-se configurar o servidor de modo que a comunicação com o cliente possa ser realizada. Para configurar esse sistema, basta executar os passos abaixo:

1. Criar um novo projeto em “*HSE Project*”.
2. Inserir a “*Bridge*” em “*HSE Network 1*” na *CPU DF51*.
3. Criar o Canal Fieldbus H1 em *DF51*.
4. Inserir os “*Devices*” em *H1*
5. Adicionar os “*Blocos*” em *FB VFI*.
6. Caracterizar os blocos no modo Offline.
7. Parametrizar os blocos.
8. Fazer o download para a planta.
9. Exportar as Tags.

É interessante observar no passo 6 descrito acima que não foi necessário configurar os Blocos Transducer, Resource block, Diagnostics e Display uma vez que a versão do Syscon utilizada no laboratório de Automação Industrial do Departamento de Engenharia Elétrica da Universidade Federal de Viçosa é uma versão superior a 6.0. Portanto, estes blocos já são instanciados automaticamente quando os equipamentos são criados na configuração.

De forma análoga a descrita acima, a estratégia de controle para todos os parâmetros da planta foi criada.

3.1.2. Cliente

MATLAB é um ambiente de programação para desenvolvimento de algoritmos, análise de dados, visualização e cálculo numérico. Através do MATLAB, podem-se resolver problemas de computação técnica mais rápido do que com linguagens de programação tradicionais, como C, C++ e Fortran.

O MATLAB é usado em uma ampla gama de aplicações, incluindo processamento de sinais, comunicação, design de controle, teste e medição, modelagem e análise financeira, e biologia computacional.

O MATLAB exerce a função de cliente. Assim, tem acesso direto aos dados da planta com permissão para ler e escrever as variáveis que desejar. Para configurar o sistema, deve-se obedecer ao fluxograma apresentado na Figura 4 abaixo.

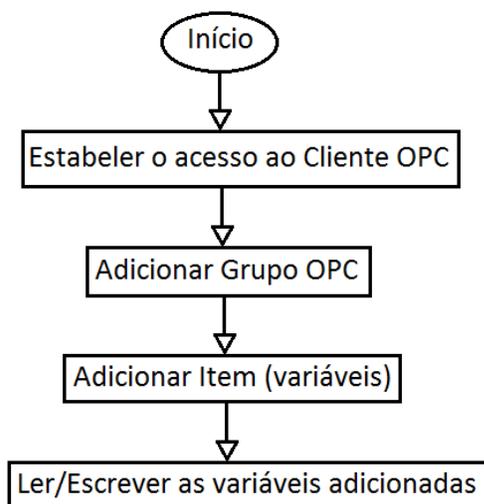


Figura 4 – Fluxograma de configuração da comunicação entre MATLAB e OPC.

3.1.3. Exportação de “Tag”

Uma vez e o procedimento de download foi concluído com sucesso e a configuração salva, um último procedimento foi exigido para habilitar a visualização da comunicação On Line com os parâmetros dos blocos funcionais. Este comando é chamado de *Export Tag*.

O Export Tag gera um arquivo (Taginfo.ini) contendo todos os tags (devices de blocos funcionais) presentes na configuração. Este arquivo é usado pela interface OPC (OLE for Process Control) com o propósito de supervisão.

A exportação dos Tags é feita através do ícone PID_Control.

É importante notar que em caso de mudanças de algum tag na configuração será necessário repetir o procedimento Export Tag. Se o comando não for executado a supervisão do tag mudado não será permitida.

O último passo dado foi, então, a comunicação Online, através do comando *Online Characterization*. Pode-se assim, ver a estratégia rodando através do comando *PID Loop Control*.

3.2. Controlador PI

3.2.1. Controlador PI sem realimentação de ganho

A fim de se fazer o controle da temperatura da planta didática 3 da smar foi projetado um controlador PI utilizando o software MATLAB. A Figura 6 abaixo apresenta o diagrama de blocos utilizado para se projetar tal controlador.

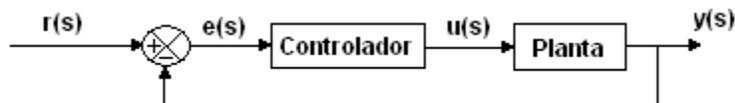


Figura 6 – Diagrama de Blocos controlador PI

A função de transferência referente ao Diagrama de Blocos mostrado na Figura 4 acima é dada pela Equação 1 abaixo:

$$u = u_a + q_0 * \text{erro} + q_1 * e_a \quad (1)$$

onde,

$$q_0 = k_p + t_a * \frac{k_i}{2}$$

$$q_1 = -k_p * \left(1 - t_a * \frac{k_i}{2}\right)$$

e_a = erro anterior

Para projetar o controlador, a princípio foram utilizados os valores da resistência (r_1) e do capacitor (c_1) de 0,017500 e 89000 respectivamente, como parâmetros referentes a uma vazão de 48,80 l/h.

Fez-se o teste com os valores de K_p e K_i de 1000 e 1 respectivamente, referentes à polos de -0.0108 e -0.0010. Os gráficos de potência, temperatura de referência e temperatura real obtidos serão apresentados nas Figuras 7 e 8 a seguir.

A fim de se melhorar o controle utilizaram-se os valores de k_p e k_i de 45 e $\frac{1}{100}$ respectivamente, referentes à polos de -0.0010 e -0.0001. Os gráficos de potência, temperatura de referência e temperatura real obtidos são apresentados nas Figuras 9 e 10 a seguir.

Como se observou ao analisar o comportamento dos gráficos obtidos nos testes descritos acima, o controlador projetado para uma vazão de 48,00l/h não forneceu uma resposta interessante.

Para tentar melhorar tal resposta utilizou-se $r_1 = 0.002200$ e $c_1 = 89000$ como valores dos parâmetros referentes a uma vazão de 400,08l/h e valores para K_p e K_i de 1000 e 2 respectivamente, referentes à polos de -0.0148 e -0.0015. Os gráficos de potência, temperatura de referência e temperatura real serão apresentados nas Figuras 11 e 12 a seguir.

Com o objetivo de melhorar o controle da temperatura final e obter uma resposta mais rápida optou-se pelos valores de K_p e K_i de 500 e 1 respectivamente, referentes à polos de -0.0095 e -0.0012. Os gráficos de potência, temperatura de referência e temperatura real serão apresentados nas Figuras 13 e 14 a seguir.

3.2.2. Controlador PI com realimentação de ganho

Ao analisar os gráficos das Figuras 11, 12, 13 e 14 notou-se uma melhora na resposta do sistema ao serem comparados aos gráficos das Figuras 7, 8, 9 e 10. Entretanto tal melhora ainda não pode ser considerada satisfatória.

Acrescentou-se então, no código já implementado o ganho da planta realimentando positivamente o sistema juntamente com uma realimentação negativa do ganho estimado da planta. A Figura 15 abaixo apresenta o novo diagrama de blocos com a inclusão do “ganho da planta” ($G'(s)$).

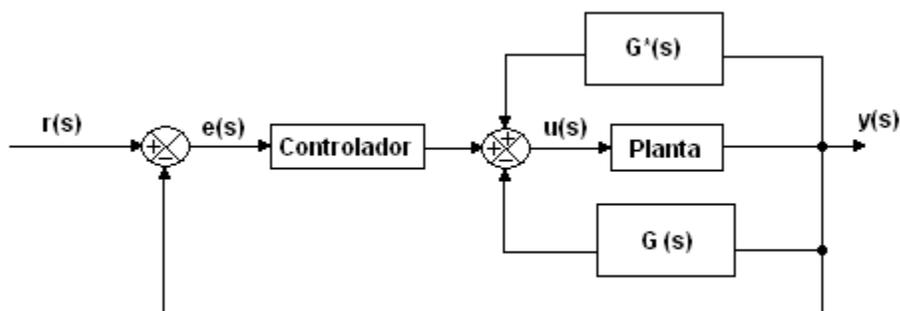


Figura 15 – Diagrama de Blocos com ganho da planta

A princípio foram propostos quatro testes. Para todos eles utilizaram-se os parâmetros referentes a uma vazão de 400,08l/h. O primeiro utilizando a temperatura real do sistema com polos mais afastados da origem. Os gráficos da potência, temperatura de referência e temperatura real obtidos serão apresentados nas Figuras 16 e 17 a seguir.

No segundo teste utilizou-se a temperatura de referência ainda com os polos mais afastados da origem. Os gráficos da potência, temperatura de referência e temperatura real obtidos serão apresentados nas Figuras 18 e 19 a seguir.

O terceiro teste consiste em repetir o primeiro utilizando polos mais próximos da origem. Os gráficos da potência, temperatura de referência e temperatura real obtidos serão apresentados nas Figuras 20 e 21 a seguir.

O quarto teste consiste em repetir o segundo utilizando polos mais próximos da origem. Os gráficos da potência, temperatura de referência e temperatura real obtidos serão apresentados nas Figuras 22 e 23 a seguir.

3.3. Lógica Fuzzy

Primeiramente elaborou-se no MATLAB, através do “*FIS editor*” uma lógica contendo uma entrada e uma saída, temperatura e corrente, respectivamente.

A temperatura foi variada de 20°C a 40°C, e a corrente de 4mA a 20mA. A Figura 24 a seguir mostra esta lógica no MATLAB.

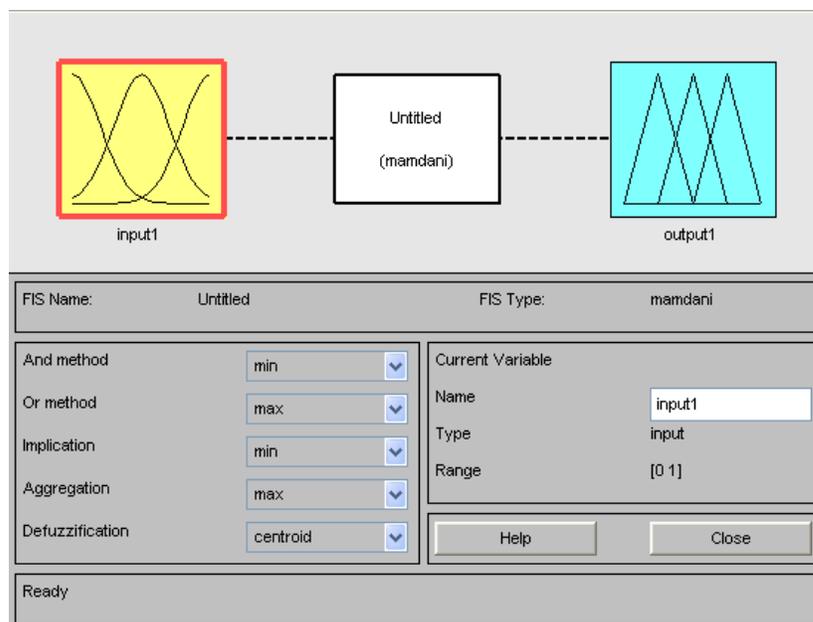


Figura 24 – Lógica Fuzzy com uma entrada.

As Figuras 25 e 26 apresentam os gráficos obtidos da entrada pelo tempo e da saída pelo tempo, respectivamente.

Em seguida elaborou-se uma lógica com duas entradas, temperatura e vazão, e uma saída, corrente.

A temperatura, novamente foi variada de 20°C a 40°C. A vazão foi variada, para fins didáticos, de -700l/h a 2000l/h. E a corrente foi variada de 4mA a 20mA. A Figura 27 abaixo mostra esta lógica no MATLAB.

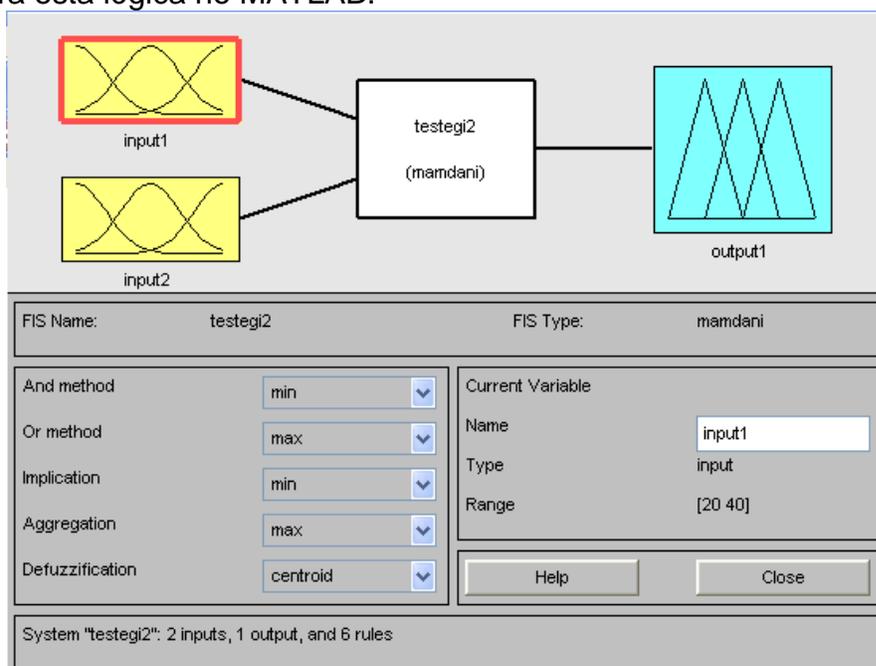


Figura 27- Lógica Fuzzy com duas entradas

As Figuras 28, 29 e 30 apresentam os gráficos obtidos das entradas pelo tempo e da saída pelo tempo, respectivamente.

4. RESULTADOS OBTIDOS

As Figuras 7 e 8 abaixo apresentam os gráficos de potência, temperatura de referência e temperatura real, obtidos no teste realizado com parâmetros referentes a uma vazão de 48,80l/h, com $k_p=1000$ e $k_i=1$.

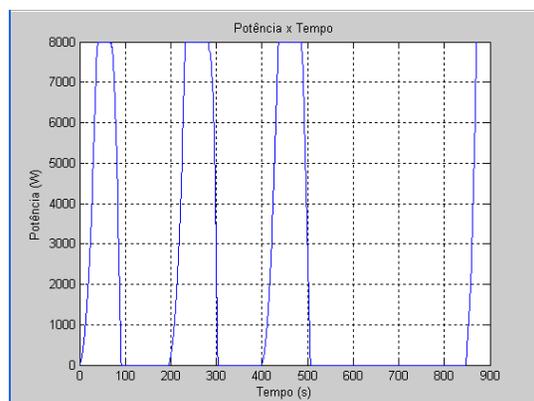


Figura 7 – Gráfico de Potência

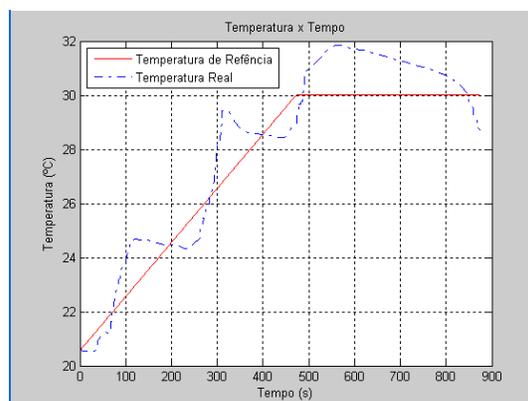


Figura 8 – Gráfico das Temperaturas

As Figuras 9 e 10 abaixo apresentam os gráficos de potência, temperatura de referência e temperatura real, obtidos no teste realizado com parâmetros referentes a uma vazão de 48,80l/h, com $k_p = 45$ e $k_i = \frac{1}{100}$.

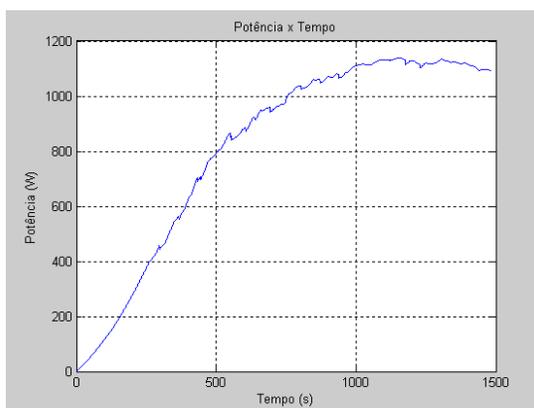


Figura 9 – Gráfico de Potência

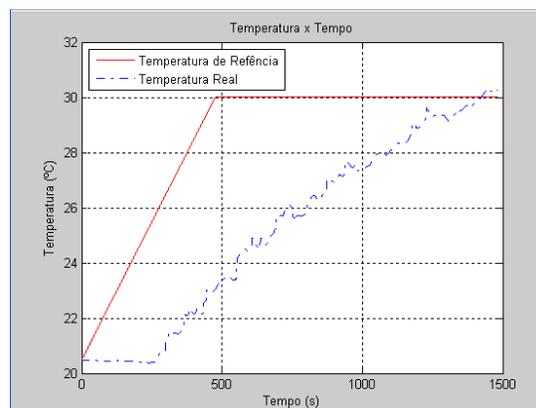


Figura 10 – Gráfico das Temperaturas

As Figuras 11 e 12 abaixo apresentam os gráficos de potência, temperatura de referência e temperatura real, obtidos no teste realizado com parâmetros referentes a uma vazão de 400,08l/h, com $k_p=1000$ e $k_i=2$.

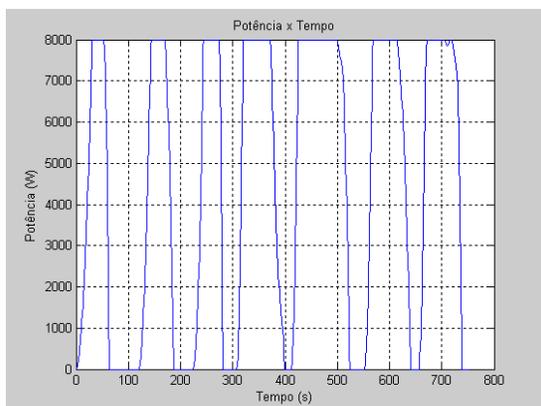


Figura 11 – Gráfico de Potência

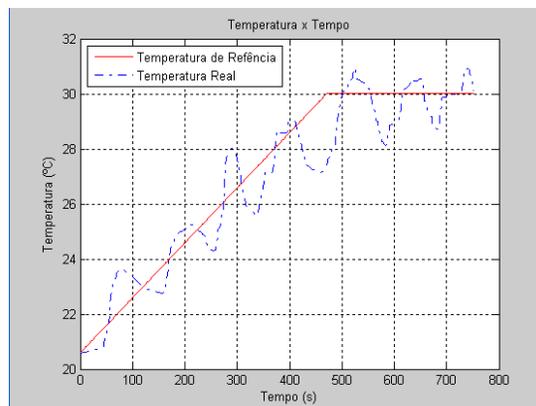


Figura 12 – Gráfico das Temperaturas

As Figuras 13 e 14 abaixo apresentam os gráficos de potência, temperatura de referência e temperatura real, obtidos no teste realizado com parâmetros referentes a uma vazão de 400,08l/h, com $k_p = 500$ e $k_i = 1$.

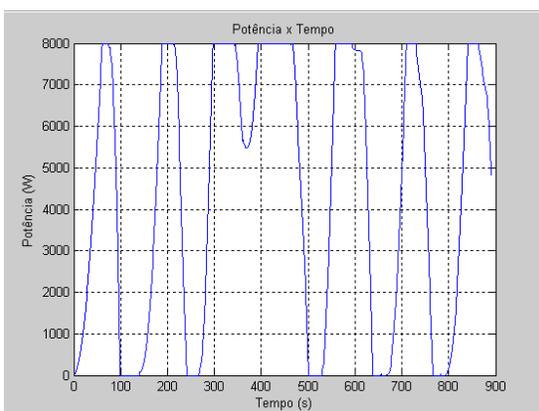


Figura 13 – Gráfico de Potência

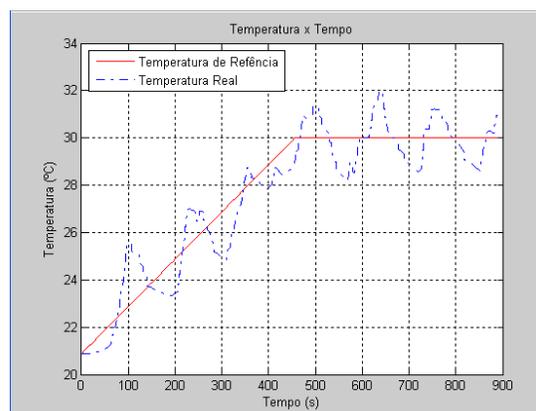


Figura 14 – Gráfico das Temperaturas

As Figuras 16 e 17 abaixo apresentam os gráficos de potência, temperatura de referência e temperatura real, obtidos no teste realizado com parâmetros referentes a uma vazão de 400,08l/h, referentes ao primeiro teste descrito anteriormente.

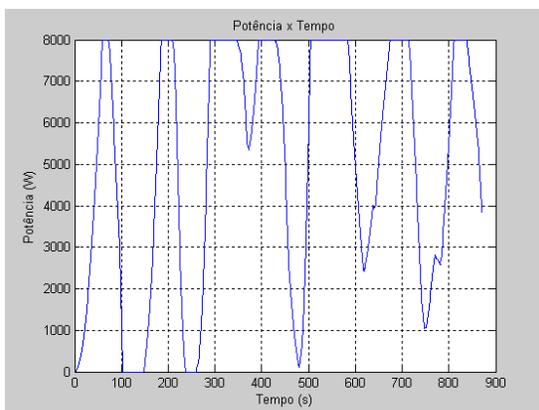


Figura 16 – Gráfico da Potência

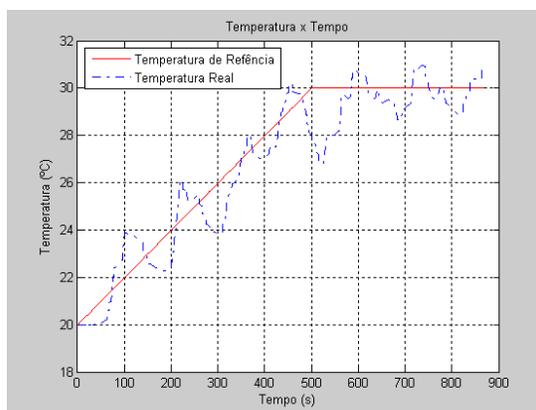


Figura 17 – Gráfico da Temperaturas

As Figuras 18 e 19 abaixo apresentam os gráficos de potência, temperatura de referência e temperatura real, obtidos no teste realizado com parâmetros referentes a uma vazão de 400,08l/h, referentes ao segundo teste descrito anteriormente.

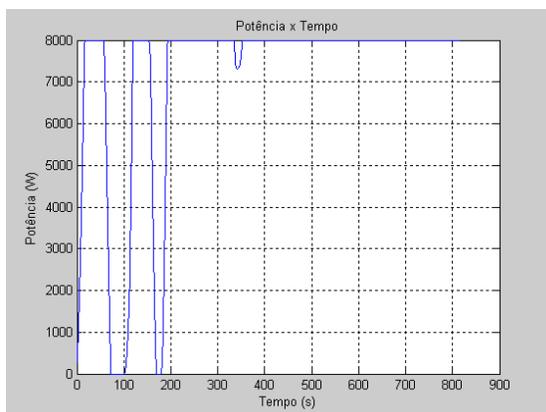


Figura 18 – Gráfico da Potência

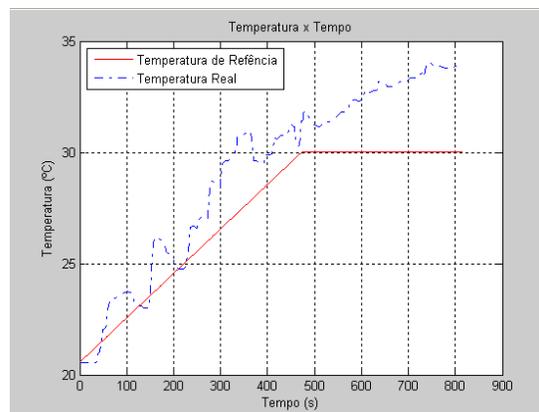


Figura 19 – Gráfico da Temperaturas

As Figuras 20 e 21 abaixo apresentam os gráficos de potência, temperatura de referência e temperatura real, obtidos no teste realizado com parâmetros referentes a uma vazão de 400,08l/h, referentes ao terceiro teste descrito anteriormente.

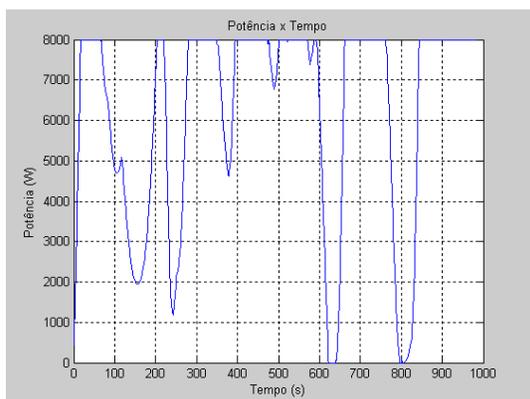


Figura 20 – Gráfico da Potência

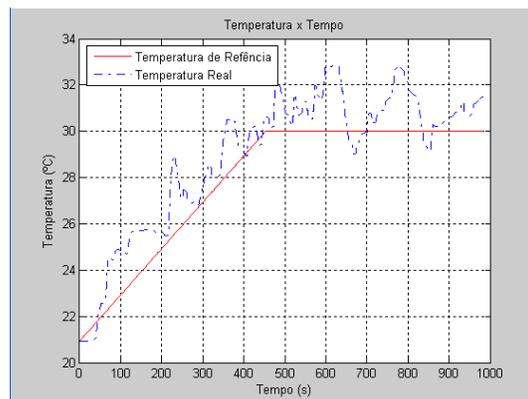


Figura 21 – Gráfico da Temperaturas

As Figuras 22 e 23 abaixo apresentam os gráficos de potência, temperatura de referência e temperatura real, obtidos no teste realizado com parâmetros referentes a uma vazão de 400,08l/h, referentes ao quarto teste descrito anteriormente.

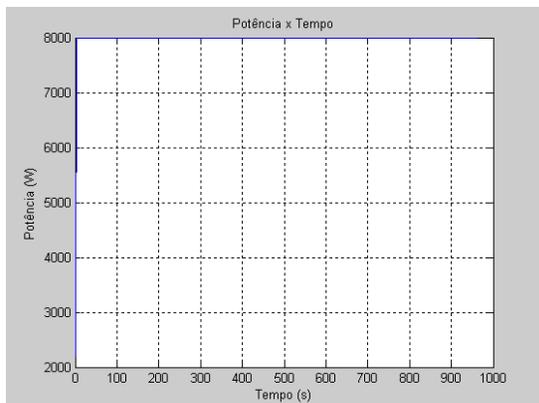


Figura 22– Gráfico da Potência

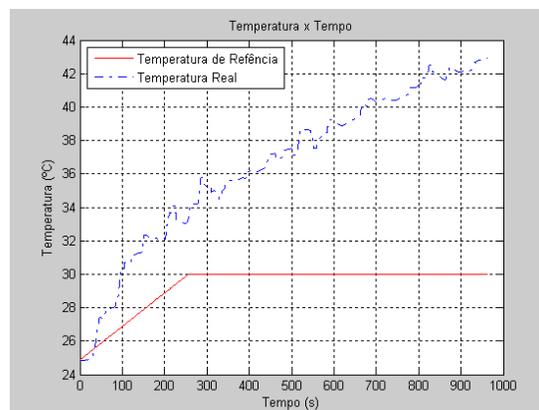


Figura 23 – Gráfico da Temperaturas

Como pode ser observado da análise dos gráficos apresentados acima, o melhor resultado obtido foi o exposto nas Figuras 13 e 14 com uma vazão maior e com polos mais próximos à origem.

As Figuras 25 e 256 apresentam os gráficos de temperatura pelo tempo e corrente pelo tempo, respectivamente obtidos no teste realizado através da Lógica Fuzzy com uma entrada.

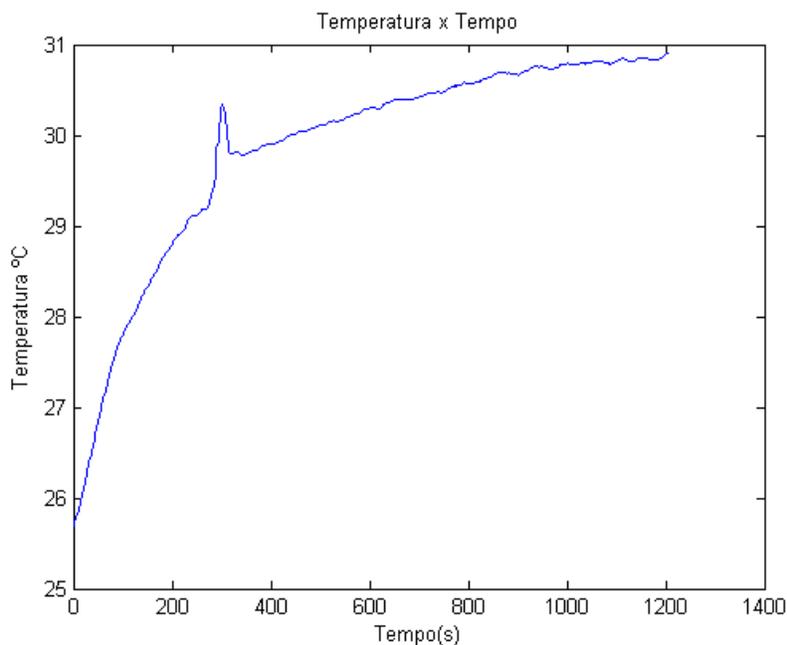


Figura 25- Temperatura x Tempo

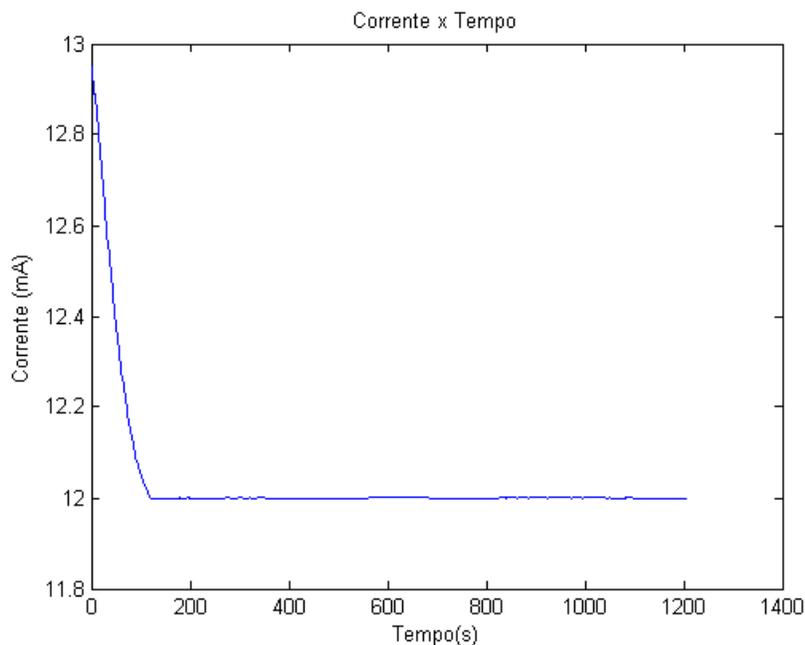


Figura 26- Corrente x Tempo

As Figuras 27, 28 e 29 apresentam os gráficos de vazão pelo tempo, temperatura pelo tempo e corrente pelo tempo, respectivamente obtidos no teste realizado através da Lógica Fuzzy com duas entradas.

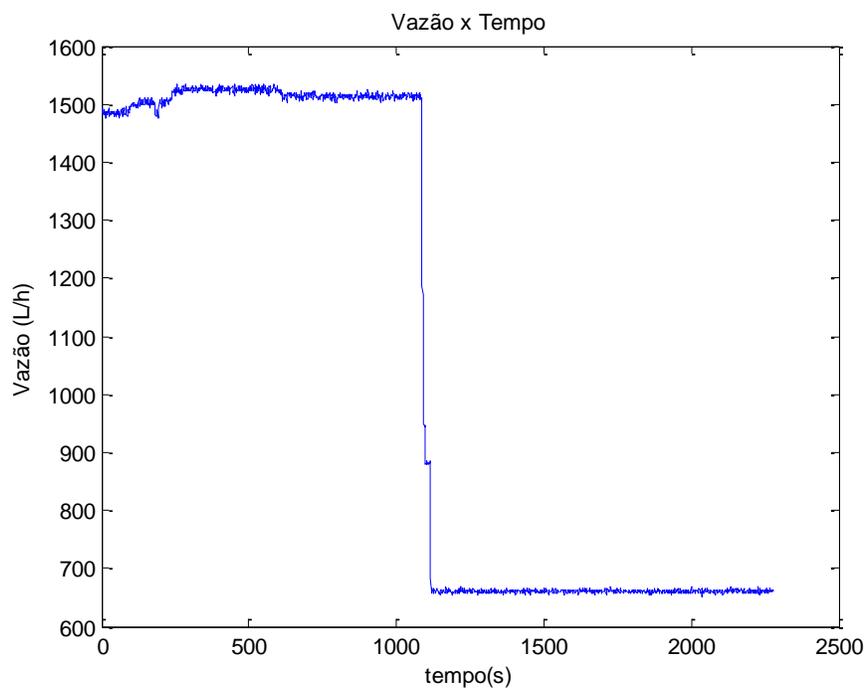


Figura 28- Vazão x Tempo

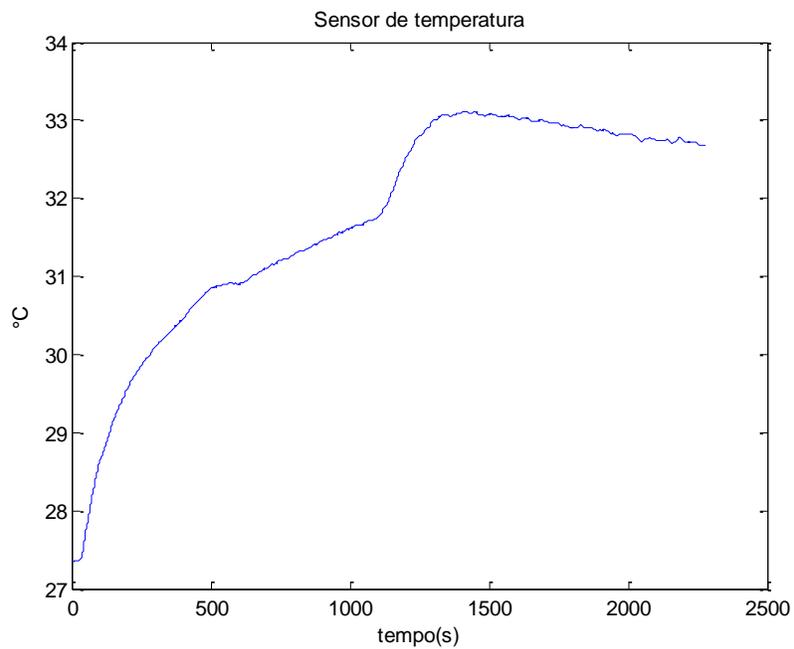


Figura 29 - Temperatura x Tempo

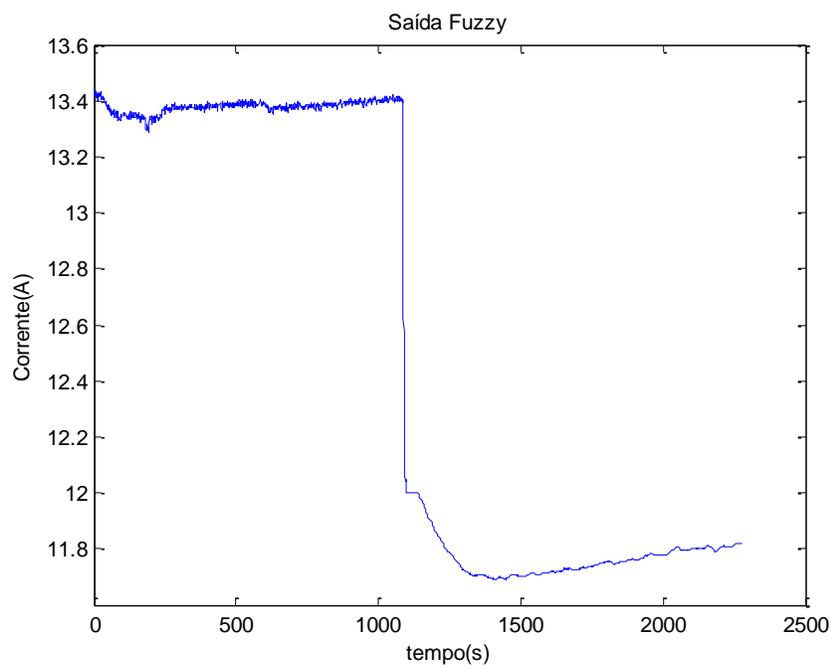


Figura 30 – Corrente x Tempo

5. CONCLUSÕES

Os testes realizados mostraram claramente que a comunicação entre o cliente OPC e os equipamentos da Smar localizados no Laboratório de Automação Industrial do Departamento de Engenharia Elétrica da Universidade Federal de Viçosa através da implementação de um rotina no MATLAB foi realizada com êxito. Uma vez que o objetivo principal era estabelecer esta comunicação e não a sintonia entre controlador e planta.

Efetuuou-se uma correta supervisão de um processo físico, automatizado com a plataforma e equipamentos da SMAR. A supervisão feita monitorou as variáveis de processo bem como interviu nos atuadores através dos controladores avançados implementados no MATLAB.

CAPÍTULO 06

6. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Ogata, Katsuhiko. Engenharia de controle moderno; tradução Paulo Álvaro Maya; revisão técnica Fabrizio Leonardi ... [et al.]. – 4. ed. – São Paulo: PrenticeHall, 2003

[2] Dorf, Richard C; Bishop, Robert H. Sistemas de Controle Modernos; tradução Bernardo Severo da Silva Filho... [et al.]. – 8. ed. – Rio de Janeiro: LTC Editora, 2001.

[3] Bazanella, A. S.; Silva, J. M. G. Sistemas de Controle: Princípios e Métodos de Projeto. Editora da UFRGS, 1ª ed., 2005.

[4] Åström, K; Häggglund, T. PID Controllers: Theory, Design, and Tuning. Instrument Society of America, 2ª ed., 1995.

[5] Chevarria, Guilherme de Calazans; Coelho, Raphael Rocha; “Controle PI: da Teoria à Implementação”.

[6] Projeto PID Tracking FFB, Smar treinamento.

[7] Barros, L. C.; Bassanizi R. C.; “Tópicos de Lógica Fuzzy e Biomatemática. Coleção IMECC. Textos didáticos. Volume 5. Campinas, SP UNICAMP.

[8] Kohagura, Tiago; Lógica Fuzzy e suas aplicações. Universidade Estadual de Londrina. Departamento de Computação. 2007

[9] Junges, José Carlos Dill; “Introdução a Lógica Fuzzy”. Universidade Federal de Santa Catarina, Centro Tecnológico, Departamento de Automação de Sistemas. 2006

[10] OPC Foundation. OPC Data Access Custom Interface Standard Version 2.05[EB/OL](2001-12). <http://www.opcfoundation.org>.

[11] Wamock, Programmable Logic Controllers - Operation and application, Prentice Hall, 1988.

[12] Manual de Aplicação SYSTEM 302; SMAR.

[13] Manual de Instruções dos Blocos Funcionais- Fieldbus Foundation; SMAR.

[14] Manual do Usuário DFI 302- Fieldbus Universal Bridge; SMAR.

[15] Manual de Instalação e Operação SYSCON- Configurador de Sistema; SMAR.

[16] Manual de Treinamento PROCESS VIEW – Software de Supervisão e Scada; SMAR.

[17] Manual de Configuração LOGIC VIEW; SMAR

[18] Manual de Programação do MATLAB: MATHWorks.