

IDENTIFICAÇÃO REMOTA DE PLANTAS INDUSTRIAIS UTILIZANDO TECNOLOGIAS OPC E *CYBEROPC*

Renato Ferreira Fernandes Júnior

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para a obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Dennis Brandão

São Carlos
2009

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

AGRADECIMENTOS

Agradeço primeiramente a minha esposa pelo enorme apoio, carinho, auxílio nas horas de crise, e nas revisões de texto que tanto precisei neste trabalho;

Agradeço também ao mestre e amigo Dr. Dennis Brandão, pela orientação, dedicação e amizade. Não fosse sua orientação, este trabalho não se realizaria;

Aos companheiros de laboratório de Automação Industrial : Eduardo Mossin e Rodrigo Palluci Pantoni pelo companheirismo que mostraram durante todos estes anos;

Ao meu parceiro de trabalho e amigo Nunzio Torrisi, por todo o apoio no desenvolvimento do projeto;

À Universidade de São Paulo, por colocar à disposição sua estrutura;

À empresa SMAR , por permitir que eu conciliasse o trabalho com os estudos;

Aos meus pais que me incentivaram e me deram saúde para conseguir ser um alguém na vida;

E a Deus, por me ajudar sempre.

RESUMO

FERNANDES, R.F. (2009). *Identificação remota de plantas industriais utilizando tecnologias OPC e CyberOPC*. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2009.

A identificação de sistemas industriais tem aplicação industrial principalmente na sintonia de malhas de controle. A sintonia freqüente das malhas de controle possibilita a manutenção das características de desempenho do processo industrial durante o ciclo de vida da produção. O fato de ser remota, neste trabalho realizada via as interfaces OPC e *CyberOPC*, possibilita inúmeras aplicações práticas como a terceirização do serviço por empresas especializadas ou a centralização da otimização nas empresas constituídas por áreas distribuídas.

Este trabalho propõe e testa experimentalmente, em sistemas lineares de primeira ordem com tempo morto, uma ferramenta computacional de identificação e sintonia remota de malhas de controle em malha aberta ou fechada utilizando controladores PID em ambiente industrial, que preenche os requisitos acima citados em uma plataforma única, e desta forma tem aplicação no ambiente industrial para ajustar as malhas de controle e em ambiente acadêmico como ferramenta de estudo de aplicações de controle e de redes industriais.

Palavras-chave: controle PID, identificação, sistemas industriais, comunicação remota.

ABSTRACT

FERNANDES, R.F. (2009). *Remote identification of industrial plants using OPC and CyberOPC technology*. Dissertation (Master) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2009.

The identification of industrial systems has industrial application mainly in control loop tuning. The frequent tuning of the control loop allows the maintenance of performance characteristics for industrial process during the production life cycle. Remote tuning, using OPC and *CyberOPC* interfaces, supports several practical applications, such as specialized companies outsourcing services or companies distributed in different areas centralizing optimization.

This project proposes and validates experimentally in first order plus dead time systems a software tool for remote identification and tuning of open or close PID control loops in an industrial environment that fulfils the requirements described above, in a single platform. The software tool could be used in control loops tuning in industrial systems, as well as in an academic environment simulating control applications and industrial networks.

Keywords: PID control, identification, industrial systems, remote communication.

LISTA DE FIGURAS

Figura 1 – Arquitetura de sistema de automação industrial	6
Figura 2 – Exemplo de uma malha de controle AI-PID- AO no protocolo PROFIBUS	9
Figura 3 – Esquemático de um bloco funcional FF	11
Figura 4 - Exemplo de uma malha de controle AI-PID-AO no protocolo FF.....	11
Figura 5 - Esquemático do Bloco PID padrão FF segundo especificação (FOUNDATION, 2003c).....	14
Figura 6 – Arranjo típico do sistema de automação industrial em uma planta	16
Figura 7 – Comunicação de SCADA e equipamentos de campo: (a) através de drivers de dispositivo e (b) através de OPC	18
Figura 8 – Exemplo de uma aplicação industrial usando comunicação remota.....	25
Figura 9 – Diagrama geral de uma arquitetura orientada a serviço (SOA).....	27
Figura 10 – Exemplo do caminho do dado dentro do servidor web (Torrise, 2007a)	30
Figura 11 – Exemplo do caminho do dado dentro do CyberOPC (Torrise, 2007a).....	32
Figura 12 – Arquitetura de sintonia típica de processos industriais.....	36
Figura 13 – Controle por realimentação clássico	47
Figura 14 – Diagrama de blocos de um sistema de controle usando IMC	54
Figura 15 – Interpretação gráfica do IAE. A área hachurada é o valor de IAE	57
Figura 16 – Arquitetura do Tele-Sintonizador	64
Figura 17 – Arquitetura do CyberOPC gateway no servidor de dados remoto da planta industrial.....	67
Figura 18 – Arquitetura do tele-sintonizador comunicando com a planta industrial remotamente através do CyberOPC gateway	68
Figura 19 – Esquemático do Tele-Sintonizador cliente CyberOPC.....	69
Figura 20 – Esquema dos tratamentos dos dados recebidos pelo OPC e CyberOPC.....	70
Figura 21 – Fluxograma do Tele-Sintonizador	71
Figura 22 – Modelo do sistema FOPDT com PI simulado no MATLAB/SIMULINK.....	73
Figura 23 – Arquitetura do sistema de teste usando FBSIMU e o tele-sintonizador remoto	74
Figura 24 - Arquitetura do sistema de teste usando DF51 e o tele-sintonizador.....	74
Figura 25 - Gráfico comparativo entre os índices de desempenho FIT, FPE e AIC.....	75
Figura 26 - Resposta dos sistemas malha aberta a uma entrada a degrau unitário.....	77
Figura 27 - Resposta dos sistemas malha fechada a variação de carga de degrau unitário..	78
Figura 28 - Gráfico da identificação local do sistema 2 em malha aberta	80
Figura 29 - Gráfico das identificações do sistema 2 com ruído de 0.05 em malha aberta ...	81
Figura 30 - Gráfico das identificações do sistema 2 com ruído 0.05 em malha aberta e utilizando filtro	82
Figura 31 - Gráfico da identificação local do sistema 2 em malha fechada.....	83
Figura 32 - Gráfico de identificação remota do sistema 2 em malha fechada	84
Figura 33 - Gráfico das identificações do sistema 2 com ruído em malha fechada	85
Figura 34 – Gráfico comparativo entre as identificações dos diferentes sistemas em malha aberta	88
Figura 35 – Gráfico comparativo das identificações dos diferentes sistemas em malha fechada.....	90
Figura 36 - Gráfico da identificação final do sistema em malha aberta em teste local.....	93
Figura 37 - Gráfico da identificação final do sistema em malha aberta em teste remoto ...	94

Figura 38 - Gráfico da identificação final do sistema com ruído de 0.05 em malha aberta.	95
Figura 39 - Gráfico da identificação final do sistema com ruído de 0.05 e filtro para sistema em malha aberta.....	96
Figura 40 - Gráfico da identificação final do sistema em malha fechada em ambiente local	97
Figura 41 - Gráfico da identificação final do sistema em malha fechada em ambiente remoto.....	98
Figura 42 - Gráfico do sistema real e de todos os modelos obtidos na identificação.....	100
Figura 43 - Gráfico do sistema real e de todos os modelos obtidos na identificação com aplicação de filtro	100
Figura 44 - Comparação da transformação do modelo dos diferentes sistemas em malha aberta nas diferentes categorias	103
Figura 45 - Comparação da transformação do modelo dos diferentes sistemas em malha fechada para as diferentes categorias da tabela 20	106
Figura 46 - Gráfico de sintonia dos diferentes métodos para teste local em malha aberta	107
Figura 47 - Gráfico de sintonia dos diferentes métodos para teste remoto em malha aberta	108
Figura 48 - Gráfico de sintonia dos diferentes métodos para teste com ruído em malha aberta	109
Figura 49 - Gráfico de sintonia dos diferentes métodos para teste local em malha fechada	110
Figura 50 - Gráfico de simulação da resposta do sistema identificado à variação de carga	111
Figura 51 - Gráfico de sintonia dos diferentes métodos para teste com ruído em malha fechada.....	112
Figura 52 - Gráfico de Identificação do modelo ARX para sistemas em malha aberta	113
Figura 53 - Gráfico de transformação de modelo para sistemas em malha aberta.....	113
Figura 54 - Gráfico da identificação do modelo ARX para sistemas em malha fechada...	114
Figura 55 - Gráfico de transformação do modelo para sistemas em malha fechada	115
Figura 56 - Arquitetura do sistema de teste usando FBSIMU e tele-sintonizador remoto.	128
Figura 57 - Tela de configuração do servidor OPC do FBSIMU (DanSrv).....	130
Figura 58 - Certificado de autenticação digital necessário para comunicação CyberOPC	131
Figura 59 - Tela principal de configuração do CyberOPC gateway.....	132
Figura 60 - Tela de configuração do CyberOPC no tele-sintonizador	133
Figura 61 - Tela do tele-sintonizador comunicando com o CyberOPC	134
Figura 62 - Estratégia de simulação de planta industrial fieldbus utilizado DF51.....	135
Figura 63 - Arquitetura do sistema de teste usando DF51 e tele-sintonizador local	136
Figura 64 - Tela principal do tele-sintonizador online capturando dados do PID	137
Figura 65 - Tela de identificação do tele-sintonizador	138
Figura 66 - Gráfico do erro de identificação comparado com a aquisição	139
Figura 67 - Gráfico de resposta a degrau dos sistemas Arx e malha aberta	139
Figura 68 - Parâmetros do PID para os métodos de sintonia escolhido	140
Figura 69 - Gráfico de simulação de variação de carga do sistema atual e os métodos escolhidos	140

LISTA DE TABELAS

Tabela 1 – Principais softwares de sintonia de PID comercial (ANG et AL., 2005)(1)	24
Tabela 2 – Principais índices de desempenho	58
Tabela 3 – Sintonia PI para mudança de SP	59
Tabela 4 – Sintonia PID para mudança de SP	60
Tabela 5 – Sintonia PI para variação de carga	60
Tabela 6 – Sintonia PID para variação de carga.....	61
Tabela 7 – Relações de sintonia para variação de carga	61
Tabela 8 - Equações dos sistemas utilizados para validação do trabalho.....	76
Tabela 9 - Resultados da identificação em malha aberta	87
Tabela 10 - Resultados da identificação em malha fechada.....	89
Tabela 11 - Resultados de transformação do modelo para teste local em malha aberta	91
Tabela 12 - Resultados de transformação do modelo para o teste remoto em malha aberta.	92
Tabela 13 - Resultados de transformação do modelo para o teste em malha aberta com ruído de sinal.....	94
Tabela 14 - Resultados de transformação do modelo para o teste local em malha fechada..	96
Tabela 15 - Resultados de transformação do modelo para o teste remoto em malha fechada	98
Tabela 16 - Resultados da de transformação do modelo para o teste em malha fechada com ruído no sinal	99
Tabela 17 - Resultados da transformação de modelo para sistemas em malha aberta	102
Tabela 18 - Resultados da transformação do modelo para sistemas em malha fechada	105
Tabela 19 - Resultados de sintonia para teste local em malha aberta	107
Tabela 20 - Resultados de sintonia para o teste remoto em malha aberta.....	107
Tabela 21 - Resultados de sintonia para o teste em malha aberta com ruído de sinal.....	108
Tabela 22 - Resultados de sintonia para o teste em malha aberta com ruído de sinal e filtro	109
Tabela 23 - Resultados de sintonia para o teste local em malha fechada.....	110
Tabela 24 - Resultados de sintonia para o teste remoto em malha fechada	110
Tabela 25 - Resultados de sintonia para o teste em malha fechada com ruído de sinal.....	111
Tabela 26 - Resultados de sintonia para o teste em malha fechada com ruído de sinal e filtro	112
Tabela 27 – Tabela de escalonamento da configuração fieldbus utilizada no FBSIMU.....	129
Tabela 28 – Configuração dos parâmetros dos blocos FF para os testes de simulação da planta fieldbus	135

LISTA DE SIGLAS

AI	<i>Analog Input</i> (Bloco de função de entrada analógica)
AO	<i>Analog Output</i> (Bloco de função de saída analógica)
ARMAX	<i>Auto-Regressive Moving eXogenous Model</i>
ARX	<i>Auto-Regressive eXogenous Model</i>
ASP	<i>Active Server Pages</i>
CLP	Controlador Lógico Programável
COM	<i>Component Object Model</i>
CORBA	<i>Common Object Request Broker Architecture</i>
CSMA/CD	<i>Carrier Sense Multiple Access with Collision Detected</i>
CyberOPC	<i>Cybernetic OPC</i> (padrão OPC desenvolvido para uso sob http)
DAIS	<i>Data Acquisition from Industrial System</i>
DCOM	<i>Distributed Component Object Model</i>
DCS	<i>Distributed Control System</i>
DDE	<i>Dynamic Data Exchange</i>
DPM1	DP Mestre Classe 1
FBSIMU	<i>Fieldbus Simulator</i>
FDDI	<i>Fiber Distributed Data Interface</i>
FIT	Índice de percentagem da variação da saída
FF	<i>Foundation Fieldbus</i>
FOPDT	<i>First order plus dead time</i>
HART	<i>Highway Addressable Remote Transducer</i>
HSE	<i>High Speed Ethernet</i>

HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
IAE	Integral do erro absoluto
IHM	Interface homem máquina
IMC	<i>Internal model control</i>
ISA	<i>International Society for Measurement and Control</i>
ISE	Integral do erro quadrático
ITAE	Integral do erro absoluto ponderado pelo tempo
ITSE	Integral do erro quadrático ponderado pelo tempo
JSON	<i>JavaScript Object Notation</i>
MPC	<i>Model Predictive Control</i>
MQ	Método dos mínimos quadrados
MQR	Método dos mínimos quadrados recursivo
OLE	<i>Object Linking Embedding</i>
OPC	<i>OLE for Process Control</i>
OPC UA	<i>OPC Unified Architecture</i>
OSI	<i>Open Systems Interconnection</i>
PHP	<i>Hypertext Preprocessor</i>
PID	Controlador Proporcional-Integral-Derivativa
PN	<i>Power Noise</i>
PRBS	Seqüência pseudo-randômica
PROFIBUS DP/PA	<i>Process Fieldbus for Discrete Process /Process Automation</i>
PV	<i>Process Variable</i>
RSVP	<i>Reservation Protocol</i>

RTP	<i>Real-time Transport Protocol</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
SD	Método de sintonia de síntese direta
SISO	<i>Single Input Single Output</i>
SNMP	<i>Simple Network Management Protocol</i>
SOA	<i>Service Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
SP	<i>Setpoint</i> (ponto de referência do processo)
SQL	<i>Structured Query Language</i>
SSL	<i>Secure Sockets Layer</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
UDP	<i>User Datagram Protocol</i>
WSDL	<i>Web Services Description Language</i>
XML	<i>Extensible Markup Language</i>
ZN	Método de sintonia de Ziegler-Nichols

LISTA DE SÍMBOLOS

G_p	Função de transferência do processo
G_c	Função de transferência do controlador
Y, y	Resposta do processo
U, u	Variável manipulada
K_c	Ganho do controlador
T_i	Termo Integral do Controlador
T_d	Termo derivativo do controlador
T_a	Tempo atraso da internet
C	Velocidade da luz
l_i	Distância do enésimo link da rede
v_i	Velocidade do enésimo roteador
Q	Quantidade de dados
r_i	Largura de banda do enésimo link
t_i^K	Atraso causado pela carga do enésimo nó da rede
K_p	Ganho estático do processo
τ	Constante de tempo do processo
θ	Tempo morto
K_u	Ganho crítico
P_u	Período crítico

T_{Filter}	Tempo do filtro de primeira ordem.
Y_{Filter}	Sinal de saída após aplicação de um filtro
$a_1 \dots a_{na}$	Coefficientes das saídas da estrutura ARX
$b_1 \dots b_{nb}$	Coefficientes das entradas da estrutura ARX
na	Ordem da matriz dos coeficientes das saídas da estrutura ARX
nb	Ordem da matriz dos coeficientes das entradas da estrutura ARX
Ψ, φ	Vetor de regressores contendo os valores de entradas e saídas do sistema.
Φ, ξ	Vetor de parâmetros estimados de acordo com a estrutura escolhida
E, e	Vetor dos erros de predição
J	Função custo – erro quadrático médio
P	Matriz de covariância
\mathcal{E}	Erro de predição
K	Ganho de Kalman
\hat{y}	Saída estimada
\bar{y}	Média de y
D	Distúrbio de excitação do sistema (variação de carga)
SP	Setpoint do sistema
bz_1, \dots, bz_n	Coefficientes do numerador da função de transferência contínua do modelo
as_1, \dots, as_n	Coefficientes do denominador da função de transferência contínua do modelo
bz_1, \dots, bz_n	Coefficientes do numerador da função de transferência discreta do modelo
az_1, \dots, az_n	Coefficientes do denominador da função de transferência discreta do modelo

T_0	Taxa de amostragem do sistema
B_i	Vetor de elementos do numerador estimado pelo ARX
A_i	Vetor de elementos do denominador estimado pelo ARX
$p_1 \dots p_n,$ $c_1 \dots c_n$	Variáveis auxiliares
\tilde{G}_c	Função de transferência do controlador estimada para o método IMC
\tilde{G}_p	Função de transferência do sistema estimada para o método IMC
$\tilde{G}_p^-, \tilde{G}_p^+$	Função de transferência do sistema estimada dividida em duas partes
f	Filtro passa baixa do controlador IMC
λ	Parâmetro de sintonia do filtro do IMC
\tilde{Y}	Saída do sistema do modelo interno IMC
m, n	Constantes dos métodos dos critérios integrais
$FTMA$	Função de transferência em malha aberta estimada do sistema
$FTMF$	Função de transferência em malha fechada estimada do sistema
yE_i	Solução i do sistema y estimada
Y_{real}	Gráfico do Y real (dado obtido direto da fonte simulada)
Y_{arx}	Gráfico do Y estimado pelo modelo ARX
Y_{red}	Gráfico do Y reduzido a partir do modelo ARX
Y_{est}	Gráfico do Y estimado em malha aberta
X	Parâmetro do controlador

SUMÁRIO

AGRADECIMENTOS	i
RESUMO.....	ii
ABSTRACT	iii
LISTA DE FIGURAS.....	iv
LISTA DE TABELAS.....	vi
LISTA DE SIGLAS.....	vii
LISTA DE SÍMBOLOS	x
SUMÁRIO.....	xiii
1. Introdução.....	1
1.1. Motivação do trabalho	3
1.2. Objetivos do trabalho	4
1.3. Organização do trabalho.....	5
2. Estado da arte de supervisão e controle distribuído	6
2.1. Sistemas de chão de fábrica.....	7
2.1.1. PROFIBUS.....	8
2.1.2. <i>Foundation Fieldbus</i> (FF).....	10
2.1.3. Controladores PID.....	12
2.1.4. O bloco funcional PID.....	13
2.2. Sistemas de gestão de processos.....	14
2.2.1. Tecnologia OPC	17
2.2.2. Sintonia de malhas de controle.....	18
2.3. Sistemas SCADA remotos	24
2.3.1. XML e JSON.....	26
2.3.2. SOA e serviços <i>web</i>	27
2.3.3. Problemas atuais de supervisão e controle baseado na <i>internet</i>	28
2.3.4. Tecnologia <i>CyberOPC</i>	31
2.4. Pesquisas atuais de supervisão e controle sobre internet.....	32
3. Identificação e sintonia de processos industriais	35
3.1. Identificação de sistemas	37
3.1.1. Etapas de identificação	39
3.2. Redução do modelo estimado.....	46
3.3. Transformação de modelo matemático.....	47
3.4. Sintonia de controladores PID.....	54
3.4.1. Método IMC (<i>Internal Model Control</i>).....	54
3.4.2. Critérios de otimização baseados em índices de desempenho integrais.....	56
3.4.3. Sintonia PID baseada em critérios de erros integrais	58
4. Proposta de arquitetura para o Tele-Sintonizador de plantas industriais através da <i>internet</i>	63
4.1. Arquitetura genérica de um Tele-Sintonizador	63
4.1.1. Planta Industrial	64
4.1.2. Servidor	65

4.1.3. O cliente remoto	68
5. Descrição dos testes e resultados	72
5.1. Metodologia dos testes	72
5.1.1. Descrição dos sistemas	75
5.2. Resultados dos testes	77
5.2.1. Identificação do modelo	79
5.2.2. Transformação do modelo	90
5.2.3. Sintonia	106
5.2.4. Análise final dos resultados	112
6. Conclusão	116
Referências	118
Apêndice	128
I. Exemplo de configuração do tele-sintonizador comunicando com CyberOPC.....	128
II. Exemplo de comunicação do tele-sintonizador comunicando com o DF51.....	134
III. Exemplo da identificação e sintonia utilizando o tele-sintonizador.....	136

1. Introdução

A competitividade crescente do setor industrial tem exigido melhorias contínuas nos processos de qualidade de produtos, otimização na produção e redução dos custos operacionais, além de padrões rígidos de segurança relacionados tanto ao processo produtivo quanto à gestão empresarial.

Atender estes requisitos pode implicar ao setor, entre outros aspectos, investimentos em automação industrial. Investimentos estes aplicados desde o chão de fábrica, com o controle distribuído no campo através de inteligência nos sensores e atuadores, até o nível de supervisão e otimização de processos nas estações de trabalho em centros de controle.

Deve-se, notar, entretanto, que a simples substituição dos sistemas de processos antigos, baseados na transmissão ponto a ponto pneumática ou elétrica, por uma tecnologia moderna com equipamentos com alta capacidade de processamento conectados em rede, não acarreta obrigatoriamente a melhoria da qualidade do produto final nem tampouco o aumento da produtividade da planta. Isto porque é necessário se considerar também uma correta sintonia das malhas de controle como fator primordial do aumento de qualidade e de otimização de custos de produção. De acordo com pesquisas realizadas no setor industrial, a maioria das malhas de controle em operação automática apresenta problemas de sintonia ou sintonia deficitária (HARRIS et al., 1999; RUEL, 2003 ; TORRES et. al. , 2006).

A possibilidade de se atender aos requisitos citados de segurança e de otimização de processos motivou o desenvolvimento direcionado de soluções tecnológicas voltadas à excelência operacional de equipamentos e sistemas de automação industrial, englobando equipamentos de campo, *softwares* de supervisão de processos, de manutenção, de diagnóstico preditivo e de auxílio à sintonia de malhas de controle.

Para a integração destes produtos em um processo ou planta automatizada, pode-se optar por uma solução “fechada”, constituída de produtos, protocolos e interfaces “proprietárias” de determinado fabricante. Ou, de forma diferente, a outra opção disponível é a solução “aberta”, composta por tecnologias dotadas de padrões de domínio público ou amplo de comunicação e de interface que permitem a integração de sistemas e de equipamentos independentemente dos fabricantes destes. A tendência atual é uma maior valorização de sistemas de automação “abertos” e padronizados que possam ser facilmente adaptáveis ou integráveis a outros sistemas e plataformas.

Esta padronização se estende desde os equipamentos de chão de fábrica, como, por exemplo, protocolos de comunicação Modbus, Hart ou Fieldbus, até o nível de supervisão e de entrega de dados consolidados de produção a sistemas de gestão corporativa através de tecnologias e interfaces padronizadas, como, por exemplo, OPC (*OLE for Process Control*), XML (*Extended Markup Language*), entre outros.

Os sistemas de auxílio à sintonia automática de malhas de controle existentes em campo atualmente localizam-se tipicamente em dois níveis da hierarquia de sistemas de automação: podem estar presentes nas estações de trabalho, sendo executadas juntamente com *softwares* de sistemas de controle e aquisição de dados ou SCADA (*Supervisory Control and Data Acquisition*), ou embarcados dentro de equipamentos de controle distribuído no campo, como, por exemplo, em controladores lógicos programáveis (CLPs), transmissores inteligentes ou DCS (*Distributed Control System*).

Quando se encontram embarcados, os sistemas de sintonia automática geralmente são realizados através de controle adaptativo onde os parâmetros do controlador são continuamente ajustados para acomodar as mudanças e distúrbios do processo. Já os sistemas de sintonia instalados em microcomputadores apresentam vantagem em relação a sistemas embarcados devido sua superior capacidade de processamento e de armazenamento de informações. Tal característica possibilita o desenvolvimento de algoritmos mais sofisticados, além de possibilitar recursos adicionais como simulações e análises gráficas, por exemplo.

Em um contexto amplo, sistemas de acesso remoto que utilizam a *internet* como meio de comunicação têm se disseminado nos últimos anos tanto em pesquisas acadêmicas como em aplicações industriais. Estudos como o de AVOY et. al. (2004) mostram o grande potencial de crescimento e diversificação de aplicações remotas principalmente em ambiente industrial.

Entre as vantagens do uso de acesso remoto via *internet* em aplicações industriais merecem destaque: a possibilidade das empresas com unidades distribuídas conseguirem acessar, compartilhar, analisar, e processar informações de chão de fábrica em tempo real e com maior agilidade, e a possibilidade de terceirização de serviços técnicos ou administrativos especializados com maior grau de interação entre os parceiros, evitando assim a necessidade de especialistas em seu quadro de funcionários.

Contudo, quando se considera a utilização da *internet* diretamente no chão de fábrica, deve-se salientar que a natureza dos sistemas de produção e de automação demandam certos requisitos que precisam ser garantidos, como o gerenciamento de

múltiplos acessos, segurança na comunicação e no sistema de controle, períodos máximos de atualização de dados de processo e a garantia de integração entre diferentes plataformas de processamento e entre equipamentos de tecnologias diversas.

As soluções de mercado e as pesquisas acadêmicas que objetivam viabilizar este cenário de integração “aberta” de sistemas de chão de fábrica via *internet* utilizam-se de tecnologias orientadas a objeto como o OPC sobre DCOM e o DAIS (*Data acquisition from industrial system*) através do CORBA, todos estes baseados em serviços *web*.

Neste escopo de aplicação, Torrisi (2007) propôs um padrão original de mecanismo de comunicação OPC voltado à *internet*, independente de plataforma e alternativo ao uso de serviços *web*. Tal padrão, denominado *CyberOPC*, faz uso de bibliotecas “abertas” de componentes leves, possibilitando, portanto, um melhor desempenho e maiores garantias de segurança de dados quando comparado às tecnologias baseadas em serviços *web*.

A literatura registra trabalhos que apresentam propostas e sistemas de sintonia automática e de controle dinâmico que fazem uso da comunicação via *internet* de diferentes formas. Uma aplicação importante registrada nestes trabalhos se dá em centros de ensino e pesquisa que disponibilizam experimentos de robótica, controle de manufatura e controle de processos para acessos remotos através da *internet*, como visto nos trabalhos de Batur et al. (2000), Parkin et al. (2002), Zeilmann et al. (2003), Ko e Chen (2005). Também se classificam nesta categoria equipamentos e funções de comunicação remota integrados a *softwares* de simulação e análise como o MATLAB (BALDA et.al., 2004) e o LabVIEW (MOSSIN, 2007).

1.1. Motivação do trabalho

A principal motivação para este trabalho é o fato de que embora a automatização esteja cada vez mais presente nas plantas industriais, uma parcela significativa das malhas de controle industrial pode ter sua eficiência aumentada significativamente através de procedimentos simples de sintonia. Segundo Desborough e Miller (2001), um terço das malhas de controle na indústria tem desempenho considerado aceitável, enquanto que os dois terços restantes têm ainda grande potencial de melhoria.

Por trás da segunda motivação está o fato de que os atuais *softwares* de sintonia de sistemas de controle disponíveis no mercado são inacessíveis para a pequena e média indústria. Estão presentes apenas em sistemas de automação de grande porte, como sistemas

digitais de controle distribuído ou SDCD, que em geral possuem toda uma gama de funções de análise, de sintonia e de simulação de malhas de controle, possuindo, portanto, preços elevados e acessíveis somente a grandes indústrias. Adicionalmente, a falta de pessoal especializado em otimização dinâmica capacitado a operar tais *softwares* dentro do quadro de funcionários das indústrias torna o custo da otimização, por vezes, impraticável para pequenas empresas.

Isto posto, a segunda motivação deste trabalho é explorar a possibilidade de desenvolvimento de uma alternativa tanto tecnológica como de modelo de negócio onde são integradas tecnologias de *internet* e padrões “abertos” de comunicação industrial com o objetivo de viabilizar tecnicamente a supervisão e a otimização remota das plantas industriais via *internet*.

1.2. Objetivos do trabalho

O objetivo maior deste trabalho é propor e verificar a viabilidade técnica de uma arquitetura computacional com a finalidade de realizar sintonia à distância de sistemas de controle industriais sobre a *internet*, com requisitos de desempenho e segurança satisfatórios.

Para cumprir este objetivo serão propostas as seguintes atividades:

1. Criação de um aplicativo computacional que possibilite a supervisão e atualização remota de dados de sistemas de controle em ambiente industrial, através da utilização do *CyberOPC* quando conectado ao sistema de controle via *internet* e através da interface OPC quando conectado ao sistema de controle de forma local. O projeto deverá atender aos requisitos de segurança de dados e do sistema de controle, desempenho para aplicações de tempo-real. Deverá também utilizar padrões abertos de forma que possa ser integrado facilmente em qualquer aplicativo industrial.
2. Implementação da técnica experimental de identificação de sistemas. A identificação será restrita aos sistemas lineares de primeira ordem que englobam parte significativa das aplicações de controle de processos industriais. Na validação da técnica de identificação será utilizado um sistema industrial simulado através do simulador de *Foundation Fieldbus* (FF) denominado FBSIMU.
3. Implementação de um método de sintonia de malhas de controle PID a ser aplicado ao sistema identificado, de acordo com as restrições citadas na descrição da atividade

2. Embora o projeto utilize um método específico para os sistemas propostos, salienta-se que poderá ser utilizado qualquer outro método de sintonia que atenda às especificações do projeto em aplicações futuras.
4. Validação da arquitetura proposta através de ensaios a serem realizados em rede local e remota.

1.3. Organização do trabalho

Este trabalho está organizado da seguinte forma:

No capítulo 2 são descritos o estado da arte de sistemas de supervisão e controle industrial, seus fundamentos, aplicabilidade e especificações.

O capítulo 3 apresenta o desenvolvimento teórico sobre identificação de sistemas e alguns algoritmos bastante difundidos na literatura para sintonia de controladores PID aplicáveis em ambientes industriais.

No capítulo 4 é apresentado, em detalhes, o sistema proposto, onde são descritos os aspectos de implementação e de execução do tele-sintonizador.

No capítulo 5 são mostrados todos os resultados de identificação e sintonia nos diferentes cenários de aplicação.

Por fim, no capítulo 6 é apresentada a conclusão e os trabalhos futuros.

2. Estado da arte de supervisão e controle distribuído

A década passada e mais intensamente a década atual têm sido cenários de mudanças nos sistemas de supervisão e controle dos ambientes industriais. Isto fez com que aparecessem novos requisitos para as soluções destes sistemas, como: flexibilidade, escalabilidade, adaptação, algoritmos inteligentes, interoperabilidade, reusabilidade e acesso à *internet*.

Cunha (2004) cita que as redes industriais possuem um alto nível de integração devido à flexibilidade das redes digitais, saindo do nível do chão de fábrica, passando pelo nível de gerenciamento do processo e chegando até ao nível de gerenciamento de negócios, como representa a figura a seguir.

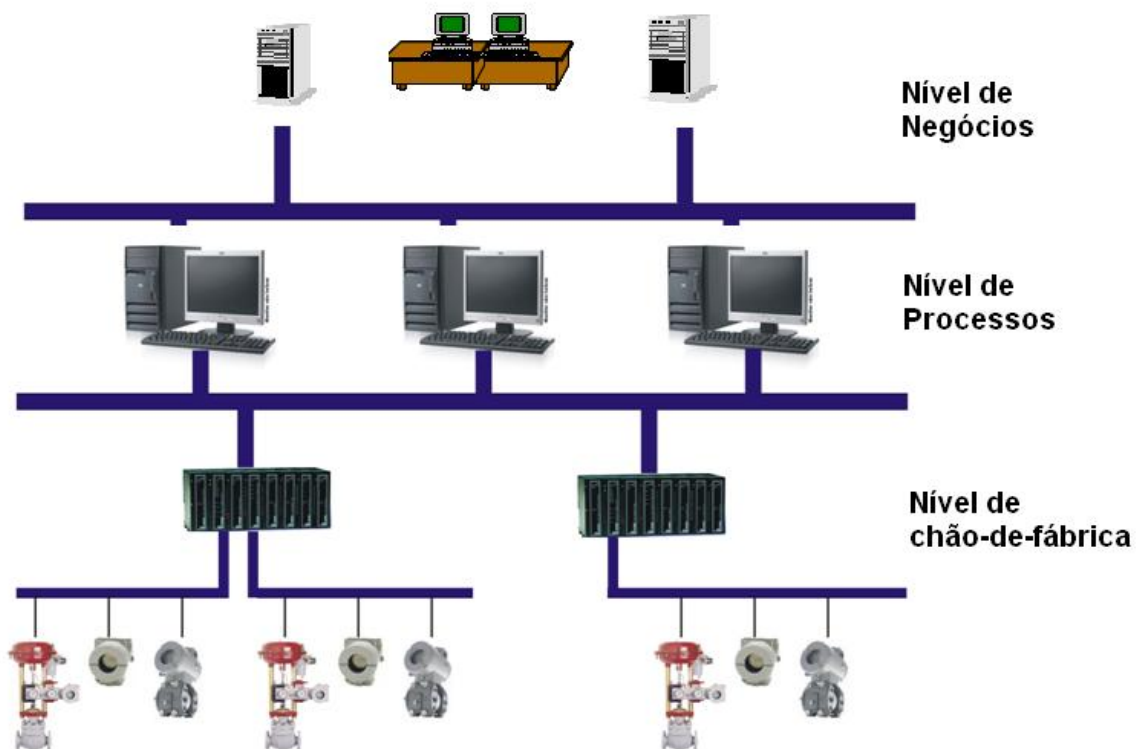


Figura 1 – Arquitetura de sistema de automação industrial

O nível do chão de fábrica compreende os equipamentos de campo e os controladores. Atualmente equipamentos de diferentes fabricantes e protocolos podem ser integrados através de barramentos de campo, ou *fieldbus*, possibilitando a integração de multiprotocolos no sistema (WANG; XIE, 2002).

No nível de gerenciamento de processo, os dados do chão de fábrica são fornecidos para sistemas do tipo SCADA e posteriormente disponibilizados para a completa operação das células de fábrica. Os sistemas SCADA englobam a parte de configuração e operação do sistema, e também as etapas de diagnóstico e sintonia das malhas de controle envolvidas. Diversos programas de sintonia têm sido propostos nos últimos anos, compreendendo desde *softwares* de controle em níveis básicos como sintonia de PID, até complexos *softwares* de controle de processo como MPC (*Model Predictive Control*) (SAYYAR-RODSARI, 2001).

A ligação entre os processos de chão de fábrica e os setores corporativos da empresa é realizada no nível de gerenciamento de negócios, onde os dados fornecidos podem ser úteis nas otimizações dos negócios. Gerry e Buckbee (2005) se apóiam no fundamento de que o acompanhamento destes dados possibilita a visualização de falhas e possíveis pontos de melhoria, contribuindo positivamente para a formação dos índices de desempenho dos negócios e auxiliando a empresa a atingir seus objetivos.

Atualmente, as expansões das empresas em unidades de negócios, buscando melhorias no atendimento ao cliente e redução de custos, com posicionamentos estratégicos em diferentes regiões, fizeram com que surgisse a necessidade de interligação destas unidades em tempo-real, utilizando para isto os sistemas baseados na *internet*. Para que se mantenha o desempenho dos controles dos processos num patamar considerado sempre ótimo, o acompanhamento remoto do parque industrial se faz necessário. Isto possibilita que pequenas e médias empresas deleguem a manutenção e otimização dos processos a empresas especialistas (terceirização) sem a necessidade de ter especialistas alocados internamente. Pode-se, neste caso, adicionalmente eliminar a necessidade de *softwares* especialistas para conduzir a manutenção local. Desta forma, tanto recursos financeiros quanto tempo podem ser economizados (YANG et al., 2006).

Baseado nas tendências tecnológicas em automação industrial, aqui serão apresentados conceitos relacionados ao sistema de chão de fábrica, sistemas de supervisão e de otimização de processos e também as tecnologias de supervisão usando *internet*, os quais servirão de base para a compreensão da estrutura proposta como tema desta dissertação.

2.1. Sistemas de chão de fábrica

Os sistemas de automação no nível de chão de fábrica são formados por dispositivos de campo que interagem diretamente no processo, seja através da leitura de grandezas

físicas, como sensores lendo temperaturas e pressões, passando por elementos de controle como os controladores até elementos finais de atuação, como atuadores.

No ambiente industrial os requisitos de comunicação em tempo-real e qualidade de serviços têm mostrado ser imprescindíveis nas aplicações (HANSSEN; JANSEN, 2003). Para a comunicação em tempo-real o principal requisito é o determinismo da rede. A qualidade de serviço visa manter o mesmo desempenho em condições de menor e maior ocupação. Por exemplo, nas linhas de produção onde existem vários procedimentos de automação que devem ser realizados com uma precisão de tempo alta, e que durante todo o ciclo de vida da produção seja garantida a mesma qualidade do serviço.

Atualmente, os barramentos de campo, também chamado *fieldbuses*, têm se tornado peças importantes dentro do ambiente industrial devido a vantagens como distribuição do controle, facilidade de expansão, interoperabilidade e intercambiabilidade, segurança, determinismo e possibilidade de acesso remoto (BRANDÃO, 2005).

As aplicações de controle e sintonia de malhas se concentram principalmente em redes de controle de processo contínuo. A automação de processos trata principalmente dos processos contínuos, como os encontrados em indústrias petroquímicas, farmacêuticas, de bebidas, açúcar e álcool, de geração de energia, entre outros. Esta área apresenta como característica sistemas mais lentos e controle analógico de variáveis de nível, pressão, temperatura, sendo utilizados como exemplo os seguintes protocolos de barramento de campo: FF, PROFIBUS-PA (*Process Fieldbus – Process Automation*) e HART. Os protocolos de maior destaque e base instalada atualmente neste tipo de processo são o PROFIBUS e FF.

Nos tópicos seguintes são apresentadas as principais características e funcionalidades destes protocolos que serão importantes para o entendimento deste trabalho. Informações adicionais sobre os protocolos encontram-se melhor detalhadas em Brandão (2005), Popp (2003), bem como nas próprias especificações PROFIBUS e FF.

2.1.1. PROFIBUS

O protocolo PROFIBUS é do tipo mestre/escravo. Os mestres podem ser classificados em: DP Mestre Classe 1 representado pelos CLPs ou DCSs. Os mestres classe 1 têm como função principal controlar o processo e o fazem através de trocas de dados cíclicos com os dispositivos escravos. Já o DP Mestre Classe 2 tem por função a

configuração, manutenção e diagnóstico da rede de campo. Este último utiliza comunicação acíclica com os dispositivos escravos da rede.

Os dispositivos escravos são estruturados internamente em unidades de funções lógicas, chamadas de módulos ou blocos. Na especificação PROFIBUS PA (para controle de processo), a padronização do protocolo somente inclui blocos de entrada e saída que são basicamente os blocos de entrada analógica (AI), saída analógica (AO) e totalizador (TOT).

Os blocos de controle e cálculo que estão dentro do mestre classe 1 não possuem padronizações, ou seja, cada cliente pode criar blocos customizados com características diferentes para cada fabricante.

A figura abaixo representa uma malha de controle formada pelos blocos AI-PID-AO em um ambiente de comunicação entre o módulo mestre classe 1 e os equipamentos de campo.

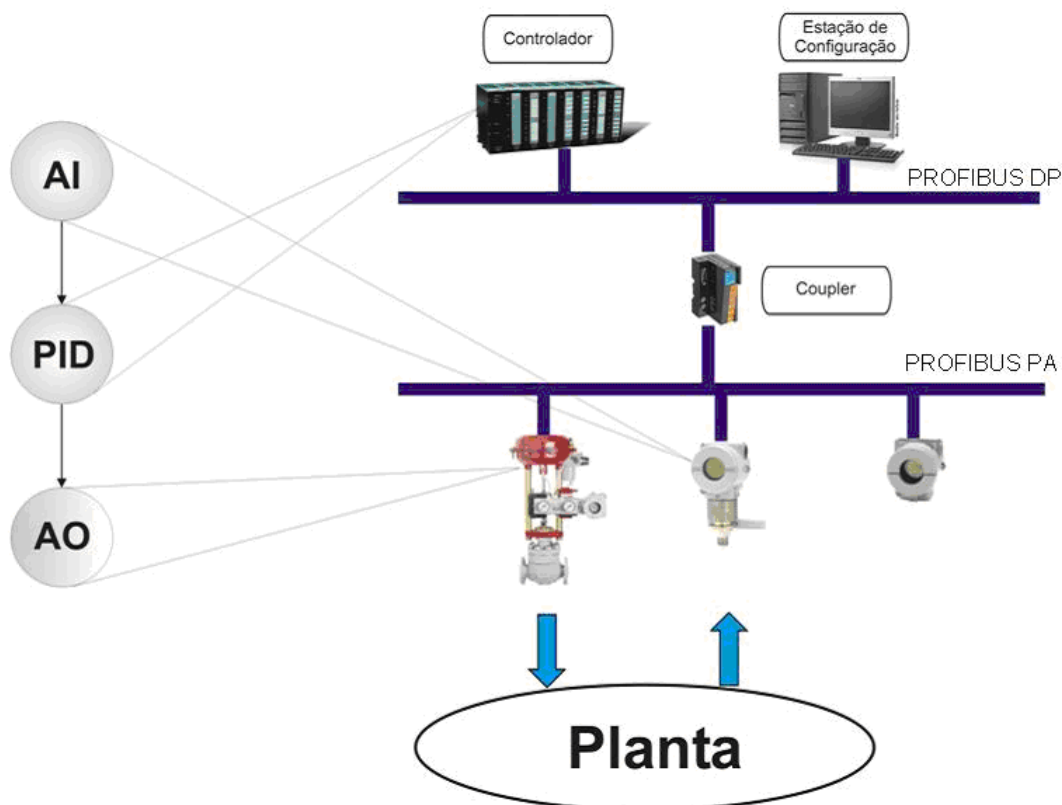


Figura 2 – Exemplo de uma malha de controle AI-PID- AO no protocolo PROFIBUS

No exemplo acima, os transmissores e atuadores possuem somente blocos de entrada e saída da variável a ser controlada. O controle é feito totalmente no controlador (CLP) por um PID não-padronizado. Desta forma, este bloco PID não-padronizado terá seus conjuntos de parâmetros bem como o algoritmo variando para cada controlador. Assim, um *software*

de sintonia de malhas se depara com a necessidade de especificação de *drives* para cada fabricante, tornando este *software* não eficiente, de custo relativamente alto e não integrável.

2.1.2. *Foundation Fieldbus (FF)*

O protocolo FF é baseado no conceito de controle distribuído, onde a troca de informações entre equipamentos de campo e controladores pode se realizar através da comunicação cliente-servidor ou produtor-consumidor (FOUNDATION, 2003a). No controle distribuído, as funções de controle podem estar distribuídas tanto nos equipamentos de campo como nos controladores.

A base fundamental para implementar a distribuição das funções de controle ao longo dos equipamentos em campo é uma entidade chamada bloco, que representa uma função ou algoritmo como, por exemplo, um controlador PID, um integrador, blocos de entrada ou saída analógica, etc. Por essa razão, esta tecnologia é normalmente chamada de tecnologia orientada a blocos.

Nesta tecnologia, toda e qualquer aplicação de usuário é baseada em blocos funcionais, cuja interface e comportamento são definidos na norma que regulamenta o padrão FF e que define uma linguagem universal para descrição de aplicações de controle de processos e automação (FOUNDATION, 2003b).

Todo bloco é composto por um conjunto de parâmetros e um algoritmo associado como mostrado na figura 3. Os parâmetros são classificados como:

- parâmetros de Entrada, que fornecem dados para o algoritmo;
- parâmetros de Saída, que representam o resultado do processamento;
- parâmetros Contidos para todas as outras funções que não sejam troca de dados entre blocos, contendo dados de configuração do bloco e variáveis de pré-cálculo.

A especificação FF define um conjunto básico de parâmetros usados por qualquer bloco, e também uma biblioteca de blocos padrão. Adicionalmente, na especificação estão previstas opções onde os fabricantes adicionem funcionalidades aos blocos já existentes ou mesmo desenvolvam novos blocos. Os blocos funcionais podem ser conectados entre si de forma a trocar informações e formar as estratégias. Estas conexões entre blocos recebem a denominação de links e são efetuadas entre as entradas e saídas dos blocos. Cada parâmetro “linkável” de entrada e saída tem associado a ele um status que determina informações de diagnóstico do link, do bloco e da malha de controle como um todo.

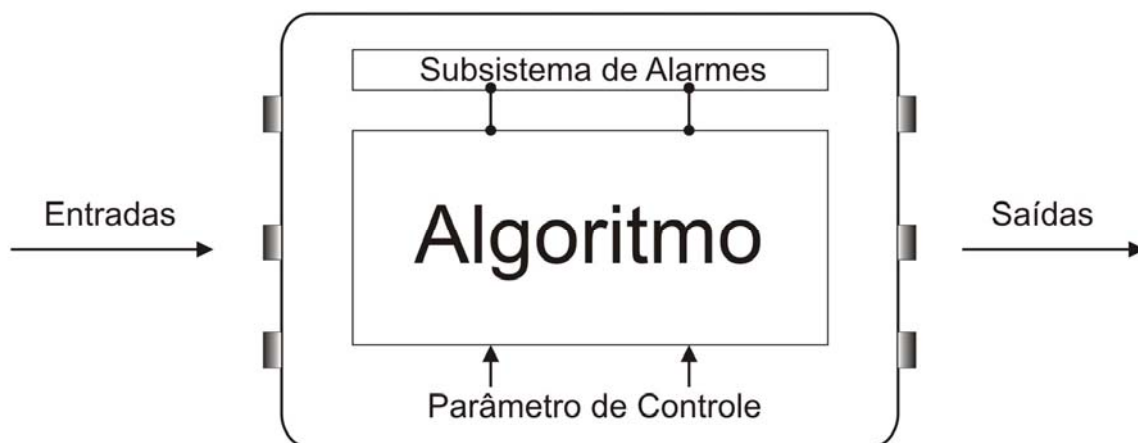


Figura 3 – Esquemático de um bloco funcional FF

Uma malha de controle que utilize um controlador PID deverá ser composta por, no mínimo, os seguintes blocos funcionais: um bloco de entrada analógica (AI), um bloco controlador PID (PID) e um bloco de saída analógica (AO), distribuídos entre dois transmissores, como representa a figura a seguir:

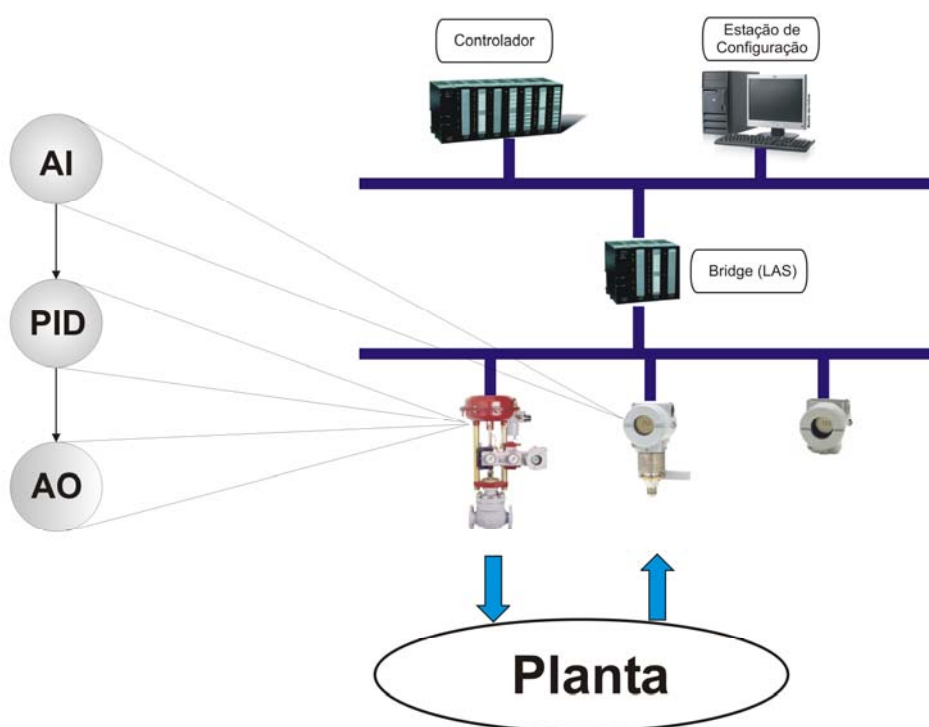


Figura 4 - Exemplo de uma malha de controle AI-PID-AO no protocolo FF

2.1.3. Controladores PID

Nos dias atuais, mesmo com os desenvolvimentos tecnológicos aplicados aos processos industriais, ainda se observa que os controles Proporcional-Integral-Derivativo (PID) oferecem soluções simples e muito eficientes para a grande maioria dos sistemas de controle industrial (ASTRÖM; HÄGGLUND, 2001). Com o avanço das tecnologias digitais, e também com a grande experiência em controladores PID, são oferecidas extensas listas de soluções para controladores nesta área. Porém, contraditoriamente, várias pesquisas mostram que grande parte das malhas de controle apresenta um pobre desempenho (YU, 2006).

Três diferentes estruturas do algoritmo de controlador PID são mais comumente encontradas na literatura: a forma padrão ou ISA (*The Instrumentation Systems and Automation Society*), a forma paralela e a forma série (ASTRÖM; HÄGGLUND, 1995). Abaixo são mostradas as fórmulas de cada equação no domínio de Laplace:

$$G_c = \frac{Y(s)}{U(s)} = K_c \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (\text{PID - ISA}) \quad (1)$$

$$G_c = \frac{Y(s)}{U(s)} = K_c + \frac{1}{T_i s} + T_d s \quad (\text{PID -Paralelo}) \quad (2)$$

$$G_c = \frac{Y(s)}{U(s)} = K_c \left(\frac{T_i s + 1}{T_i s} \right) + (T_d s + 1) \quad (\text{PID- Série}) \quad (3)$$

Sendo K_c o ganho proporcional, T_i a constante de tempo integral, T_d a constante de tempo derivativa.

As funcionalidades dos três termos são mostradas abaixo:

- Termo Proporcional – provê uma ação proporcional ao sinal do erro.
- Termo Integral – reduz os erros de regime através das compensações das baixas frequências por um integrador.
- Termo Derivativo – aumenta a resposta transiente através da compensação das baixas frequências por um diferenciador.

A maioria dos sistemas industriais utiliza microprocessadores para processar os cálculos do controle seja ele centralizado em um controlador como nos equipamentos PROFIBUS, ou distribuído nos equipamentos de campo, como no FF. Desta forma, os equipamentos digitais amostram as variáveis de controle e calculam as saídas do controlador de uma forma discreta. Assim, o controle é realizado em intervalos regulares com um intervalo de amostragem T_0 .

Devido às diferenças entre os parâmetros dos algoritmos ISA, série e paralelo, e também das diferenças que poderão haver entre os vários algoritmos, é necessário saber explicitamente em qual algoritmo é baseado o controlador.

Atualmente existem diversos métodos de sintonia na literatura que serão mostrados adiante. Alguns destes métodos serão detalhados no capítulo 3, pois serão utilizados neste trabalho.

2.1.4. O bloco funcional PID

Nesta seção será mostrado o bloco funcional PID padronizado pela especificação FF (FOUNDATION, 2003c) e que garante um nível mínimo de funcionalidades entre os fabricantes de equipamentos.

O bloco PID FF é o bloco de controle usado principalmente para sistemas lineares. O algoritmo do PID é do modelo ISA, como mostrado pela equação (1). Quanto à atuação do controlador, ele é classificado como PID, ou seja, o ganho proporcional é aplicado a todos os termos do PID, o Proporcional e o Integral atuam sobre o erro, e o Derivativo atua sobre o valor da PV. Portanto, mudanças no SP não causarão impacto na saída devido à ação do termo derivativo, quando o bloco estiver configurado para operar no modo automático.

O bloco PID sempre trabalha internamente com porcentagem da escala de entrada e converte esta porcentagem em unidade de engenharia da saída. Estas escalas são configuradas pelo usuário. Os parâmetros GAIN, RESET e RATE são as constantes de sintonia para os termos Proporcional, Integral e Derivativo, respectivamente.

É possível escolher a ação de controle direta ou reversa através dos parâmetros de configuração do bloco. Quando a ação é direta, o erro será obtido subtraindo o *setpoint* (SP) da variável do processo (PV). Para a ação reversa, o erro é calculado subtraindo a PV do SP. O bloco PID possui também o controle antecipatório (*FeedForward*). Neste caso é fornecida uma variável externa, a qual é proporcional a alguns distúrbios na malha de controle e que vai auxiliar na antecipação do controle. Outras características do PID padrão são: saída rastreada (*output tracking*), utilizada quando ocorre um evento externo ao bloco colocando o controle em segurança, e também salto (*bypass*) da entrada para a saída, caso não seja necessário calcular a saída. O bloco suporta, adicionalmente, cascata de PIDs, filtros de entrada e tratamento de alarmes. Um esquemático das características do bloco é representado na figura a seguir:

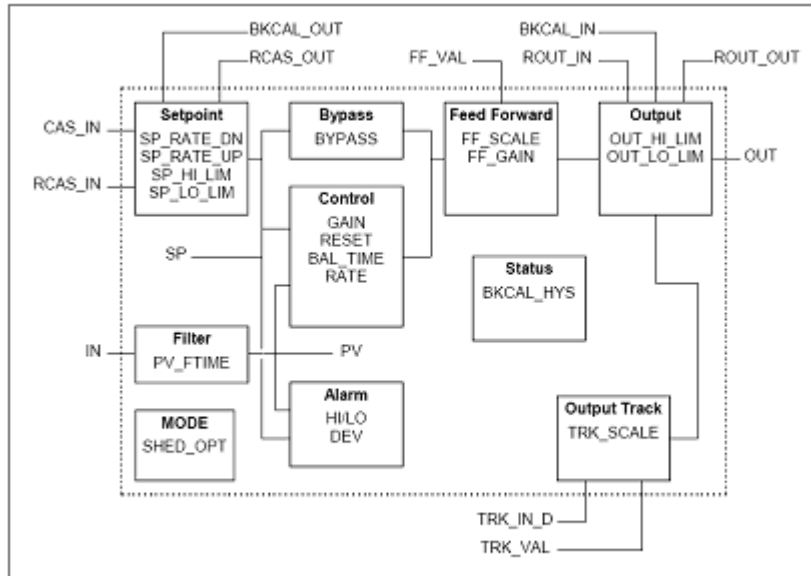


Figura 5 - Esquemático do Bloco PID padrão FF segundo especificação (FOUNDATION, 2003c)

2.2. Sistemas de gestão de processos

Na gestão de processos, os *softwares* SCADA são os sistemas responsáveis por aquisição de dados, tratamento e atuação dos dados do nível de chão de fábrica. Aplicações SCADA costumam ser compostas de algumas centenas a milhares de pontos das diversas partes de controle do processo (TORRES; HORI, 2005).

Quanto à estrutura de comunicação baseada no modelo OSI (*Open Systems Interconnection*) desenvolvido pela *International Standards Organizations* (ISO), na camada de enlace predominam as redes baseadas no protocolo IEEE 802.3, conhecida como *Ethernet* (IEEE, 2001).

A *Ethernet* é uma rede que usa a tecnologia CSMA/CD (*Carrier Sense Multiple Access with Collision Detected*) que resolve colisões de dados em uma forma não determinística, desta forma ela não é considerada uma rede de tempo-real (HANSSEN; JANSEN, 2003).

Ainda neste contexto existem outras redes locais consideradas de tempo-real como IEEE 802.5 baseada na passagem de símbolo (*token-ring*) e *Fiber Distributed Data Interface* (FDDI), que é uma evolução da anterior, porém em um ambiente mais amplo considerando várias redes locais (PLEINEVAUX, 1995), todas elas consideradas

determinísticas, porém consideradas muito caras. Atualmente, a *Ethernet* é ainda a mais utilizada.

Na camada de transporte do modelo OSI predominam em ambiente industrial os serviços de transporte UDP (*User Datagram Protocol*) e o TCP (*Transmission Control Protocol*). O UDP não é voltado à conexão e desta forma não garante a chegada do pacote ao destino enquanto o TCP, que é orientado à conexão, possui esta garantia. A tecnologia HSE (*High Speed Ethernet*) trabalha sobre o protocolo UDP, enquanto outras tecnologias como Modbus TCP/IP, Ethernet IP, ProfiNet trabalham sobre TCP/IP.

Visando melhorar o desempenho das redes e dos softwares SCADA os protocolos fieldbuses contam com mecanismos específicos e padrões. Um desses mecanismos, por exemplo, é o serviço de *Trends e Alerts* do protocolo HSE. Os *trends* consistem de um conjunto de amostras de uma determinada variável do bloco em *macrocycles* sequenciais com o respectivo *timestamp* do equipamento (FOUNDATION, 2003a,2003b).

A arquitetura SCADA encontrada mais frequentemente no ambiente industrial com tecnologia *fieldbus* pode ser dividida em três categorias principais por funcionalidades de *software*: *softwares* de configuração de dispositivos e malhas de controle, *softwares* de supervisão e operação do processo e *softwares* de gerenciamento de ativos (PANTONI, 2006). A figura 6 mostra um exemplo desta arquitetura.

Os *softwares* de configuração ou configuradores de dispositivos e malhas de controle são usados para programação da estratégia de controle e na iniciação do mesmo no campo. Os *softwares* de supervisão ou supervisórios são utilizados na operação da planta. Geralmente são dotados de parte gráfica que facilita a visualização e operação da planta pelos operadores. Por último, os *softwares* de gerenciamento de ativo (ou de *Asset Management*) têm como função gerenciar a planta com objetivo de evitar paradas do processo. O gerenciamento de ativos engloba hoje uma vasta área de *softwares* de supervisão, entre eles monitoramento e calibração de instrumentação, gerenciamento de alarmes, monitoramento de redes e computadores, e supervisão e sintonia de malhas de controle. Estes *softwares* têm apresentado, recentemente, uma grande importância na automação industrial, pois estes gerenciamentos possibilitam a obtenção de melhores desempenhos e verificação de possíveis problemas que estejam ocorrendo na planta (TORRES, 2005)

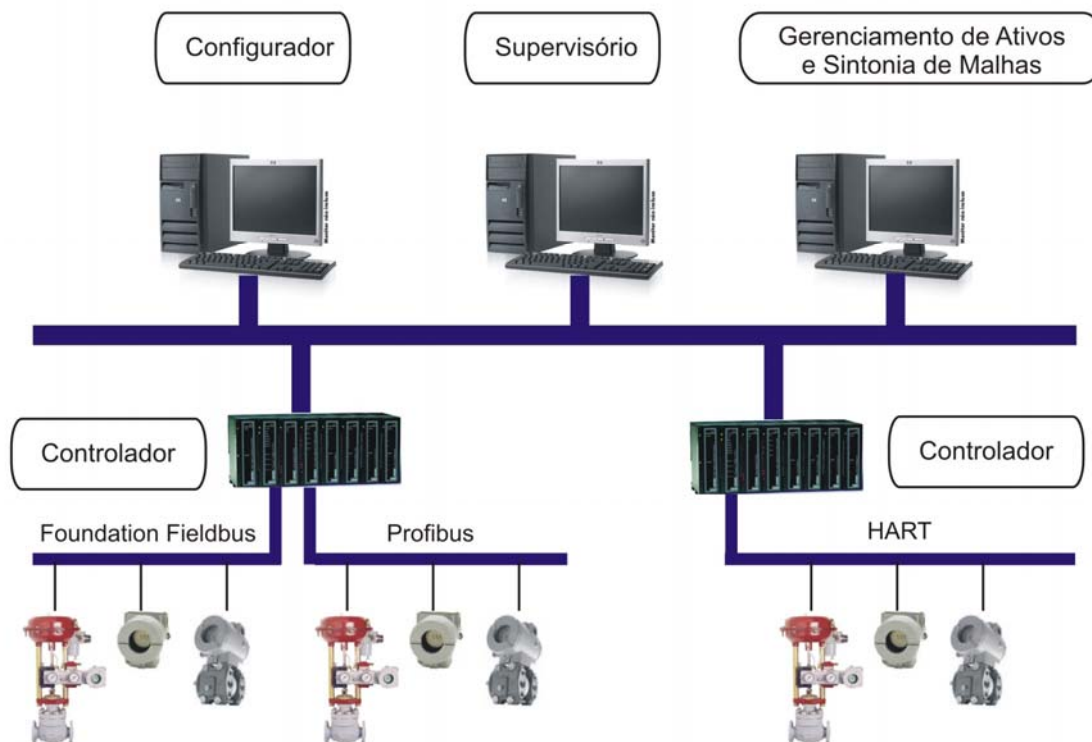


Figura 6 – Arranjo típico do sistema de automação industrial em uma planta

Ainda neste contexto, cada vez mais são requeridos sistemas abertos nas soluções, implicando na padronização dos sistemas. O maior apelo dos sistemas abertos é a liberdade de escolha de soluções proporcionada ao cliente e ao usuário final (PANTONI, 2006).

Com a disseminação de diversos protocolos de campo e a necessidade de comunicação entre vários sistemas em ambientes heterogêneos e de forma distribuída, fez-se a necessidade de padronizar também a forma de interface entre os componentes dos *softwares* SCADA. Desta forma, surgiram tecnologias orientadas a objetos distribuídos como o padrão OPC (*OLE for Process Control*) e o padrão DDE (*Dynamic Data Exchange*) baseados nas tecnologias COM/DCOM do Windows e os padrões independentes de plataforma como CORBA (*Common Object Request Broker Architecture*) (ANDRADE; MACEDO, 2005). Embora todas estas tecnologias atendam aos requisitos de sistemas distribuídos e orientação a objeto, o mais difundido no ambiente industrial atualmente é o padrão OPC (PATTLE; RAMISCH, 1997; KAPSALIS et al., 2002).

A tecnologia DDE foi a primeira a ser introduzida pela Microsoft baseada na arquitetura Windows. A função primordial da DDE é permitir que aplicações Windows compartilhem dados.

A evolução da tecnologia DDE são as tecnologias COM (*Component Object Model*) e DCOM (*Distributed Component Object Model*) da Microsoft. Elas permitem a criação de

componentes de *software* distribuídos em computadores interligados em rede. Esta tecnologia é usada para a composição de partes de aplicativos, que provê a disponibilidade de serviços para qualquer outra aplicação desenvolvida numa linguagem que suporta COM, como C++, Visual Basic, entre outras. A diferença destes arquivos para os executáveis convencionais é que os componentes COM são construídos com uma interface padrão, o que garante compatibilidade binária entre os diversos componentes. O DCOM é uma extensão do COM para a comunicação entre objetos em sistemas distribuídos (JANKE, 2000).

A evolução da tecnologia DCOM são as plataformas de desenvolvimento dotNET e dotNET Remoting (dotNET e dotNet Remoting são denominações difundidas para as estruturas de comunicação do inglês *Common Language Infrastructure* e *Common Language Runtime*, respectivamente) desenvolvidas pela Microsoft para sistemas abertos que são independentes de plataforma (MICROSOFT, 2007).

Já a tecnologia CORBA (GROUP, 2007) foi criada pelo *Object Management Group* (OMG). Esta tecnologia estabelece e simplifica a troca de dados entre sistemas distribuídos heterogêneos promovendo interoperabilidade entre plataformas distintas de diferentes fabricantes. A CORBA utiliza uma linguagem de interface chamada IDL (*Interface Definition Language*), que é puramente declarativa, e, portanto, independente da linguagem de programação utilizada para acessá-la.

Neste trabalho foi escolhida a tecnologia orientada a objetos distribuídos OPC sob a plataforma dotNet pela facilidade de integração em sistemas distribuídos, por ele oferecer um acesso transparente aos equipamentos de automação e por ser atualmente o padrão “de facto” utilizado em sistemas de automação. Um outro motivo que resultou nesta escolha é fato do projeto usar blocos funcionais em sistemas com tecnologia *fieldbus* e o OPC hoje ser uma metodologia de acesso unificada para a classe de sistemas *fieldbus*.

2.2.1. Tecnologia OPC

A supervisão e operação de plantas vêm utilizando, desde o final da década de 90, as facilidades promovidas pelo padrão OPC (OPC FOUNDATION, 2006) que é baseado na tecnologia COM/DCOM da Microsoft.

A tecnologia OPC é baseada na comunicação entre cliente e servidor. Os clientes são desenvolvidos pelos fabricantes dos *softwares* SCADA e os servidores são fornecidos pelos fabricantes dos equipamentos. Com isto, basta o fabricante da interface homem máquina (IHM) desenvolver uma única vez o cliente, que esta passa a ter acesso a uma infinidade de servidores existentes para cada equipamento diferente existente no mercado.

(BLACKWELL, 1996). A figura seguinte ilustra as diferenças de aplicações utilizando *drivers* proprietários nos aplicativos e com padrão OPC.

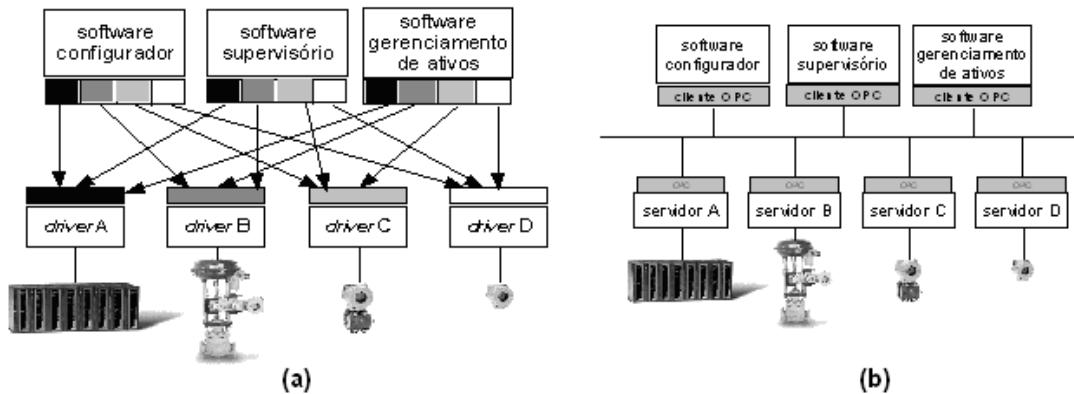


Figura 7 – Comunicação de SCADA e equipamentos de campo: (a) através de drivers de dispositivo e (b) através de OPC

A atual arquitetura OPC contempla diferentes conjuntos de especificações para cada domínio de aplicação. Para a supervisão dos dados de uma planta, inicialmente, foi especificada a norma OPC DA (*Data Access*) que define todas as interfaces necessárias para acesso e atuação em dados do processo. Outras normas foram criadas: OPC A&E (*Alarms & Events*), OPC HDA (*Historical Data Access*), *OPC Security*. Todos estes padrões seguem a tecnologia COM e DCOM que são baseadas na plataforma Windows (ZHENG; NAKAGAWA, 2002).

2.2.2. Sintonia de malhas de controle

A sintonia de malhas de controle tem um papel importante no desempenho do sistema de processo, pois ela é determinante na estabilidade do sistema de controle bem como as conseqüências desta estabilidade, como redução de perdas de matéria-prima e também a melhoria da qualidade do produto final.

A sintonia de um controlador PID consiste na determinação dos parâmetros Proporcional, Integral e Derivativo que melhor se adéquem ao sistema, tal que a robustez e os critérios de desempenho desejado sejam obtidos. A etapa de sintonia visa assegurar que os critérios de desempenho estabelecidos para a malha de controle sejam alcançados (ASTRÖM; HÄGGLUND, 1995).

As formas de sintonia podem ser divididas em: sintonia adaptativa e sintonia sob demanda. A sintonia adaptativa é realizada automaticamente pelo sistema de controle através do reconhecimento das variações da planta. O sistema, por sua vez, constantemente

monitora as mudanças normais da PV em relação ao SP durante o processo e, desta forma, realiza a identificação em malha fechada sem causar distúrbio algum ao processo. A sintonia sob demanda é realizada através da excitação da planta sob intervenção do usuário. Esta excitação permite o cálculo para obtenção dos parâmetros do controlador. A sintonia sob demanda pode ser realizada diretamente no controlador ou através de equipamentos externos (ANG et al., 2005).

Quando o sintonizador é externo, a sintonia será realizada nos controladores dos diferentes fabricantes. Para isso, é necessária a obtenção de uma série de informações importantes do sistema como: tipo de PID (padrão, paralelo ou série), taxa de amostragem, constantes de tempo de filtros, unidades dos diferentes parâmetros do controlador, banda proporcional, repetições por minutos, (por segundos) (ASTRÖM; HÄGGLUND, 1995).

De acordo com Santos (2006), os procedimentos de sintonia são realizados seguindo duas etapas básicas: a identificação do processo e a determinação dos parâmetros do controlador. Abaixo são descritas cada uma destas etapas.

Identificação de processo

Yu (2006) destaca que a identificação dos sistemas é uma das partes mais importantes no processo de sintonia de malhas de controle. E Miranda (2005) contribui com a premissa de que mais da metade do esforço gasto em sintonia está na fase de identificação. Assim, observa-se que o sucesso de uma boa sintonia está relacionado a uma boa identificação do modelo.

Uma das discussões sobre a identificação dos processos é como operar a malha de controle durante a identificação: de forma aberta ou fechada. A identificação em malha aberta é realizada com o controlador em operação no modo manual, ou seja, o controlador é colocado fora de operação normal durante a identificação. Já na identificação em malha fechada, o controlador é configurado para operar no modo automático.

De acordo com estudos de Miranda (2005), no passado optava-se pela identificação em malha aberta devido à facilidade no desenvolvimento e também pela segurança da malha de controle. A identificação em malha fechada era algo não recomendável, sendo usado somente quando o seu uso não fosse contornável, como em sistemas instáveis em malha aberta.

Van Den Hof (2004) aborda as diferenças de comportamento do sistema em malha aberta e fechada. Nota-se que existe um maior interesse em tentar recuperar a dinâmica da

planta com o controlador em malha fechada, pois, desta forma, o sistema fornecerá dados mais relevantes na identificação do modelo.

De acordo com a informação obtida, os métodos de identificação podem ser classificados com abordagem no domínio da frequência e no domínio do tempo (AGUIRRE, 2004).

Na abordagem do domínio da frequência, a excitação do sistema é realizada através de um relé com intuito de gerar uma oscilação constante. Assim, são obtidas as margens de ganho e fase e largura de banda. Eles são comuns em auto-sintonizadores embarcados no controlador (CAON, 1999). O método mais abordado na literatura é da excitação a relé com histerese que é abordado na seqüência dos métodos de sintonia.

Na abordagem do domínio do tempo, a obtenção da resposta transiente se dá através da excitação do sistema usando um sinal no formato degrau, um impulso ou uma seqüência pseudo-randômica binária (PRBS) na entrada do sistema. Então são obtidas as constantes de tempo, ganho estático e o tempo morto do sistema em malha aberta ou fechada. Apesar da simplicidade da excitação a degrau, este tipo de identificação não é adequado para sistemas que apresentam elevadas não-linearidades (YU, 2006).

Método de sintonia

Desde a introdução da sintonia proposta por Ziegler-Nichols (ZN) em 1942, a sintonia de controladores passou por um extenso estudo nos últimos 60 anos. O'Dwyer (2003) pesquisou os diferentes métodos de sintonia existentes na literatura desde ZN e registrou 245 regras de sintonia, onde 104 são para controladores PI e 141 para controladores PID. Yu (2006) discute que a escolha do método correto de sintonia deve considerar as particularidades do processo. A correta sintonia dependerá do conhecimento das características deste processo como o tipo, ordem dos sistemas, não-linearidades, de incertezas, como saber se o sistema tem um longo tempo morto, tentar identificar possíveis problemas de ruído e sugerir filtragem.

Ziegler e Nichols (1942) foram pioneiros no desenvolvimento de métodos de sintonia de controladores PID e propuseram dois métodos. Um dos métodos é baseado no sistema em resposta a degrau em malha aberta obtendo o sistema através de uma determinação gráfica da planta. O outro se baseia no ganho crítico K_u e no período crítico P_u a partir de uma oscilação sustentável, chamada de oscilação cíclica. Os dois métodos são baseados no decaimento de $1/4$, que produz respostas oscilatórias e largos *overshoots*.

Embora estes métodos sejam considerados antigos e dos problemas apresentados, ainda são largamente aplicados na indústria (ASTRÖM; HÄGGLUND, 1995).

Muitas versões modificadas dos métodos de ZN são propostas na literatura: Chien et al. (1952) propuseram um método baseado em ZN para dar melhor amortecimento em sistemas de malha fechada. O método propõe dois critérios: de resposta mais rápida possível sem *overshoot* e com *overshoot* de 20%. No método de Cohen e Coon os parâmetros são função da razão θ/τ (relação entre tempo morto e constante de tempo). Quanto menor o K_p (ganho do processo), maior poderá ser o K_c (ganho do controlador). Uma regra mais conservadora de sintonia é proposta por Tyreus e Luyben que se apóia em um sistema integrador com tempo morto. O método tem grande êxito em processos com constante de tempo dominante e sistemas multivariáveis (YU, 2006). Este mesmo autor sugere as regras de Ciancone-Marlin para processos com tempo morto dominante. Neste caso, o método é obtido examinando processos com puro tempo de atraso e se obtém os fatores de sintonia convertendo as constantes do processo de sintonia em K_u e P_u como em ZN. Neste método, são obtidas uma ação proporcional mais conservativa e uma ação integral mais agressiva do que em ZN.

Aström e Hägglund (1984) estudaram um teste de auto-sintonia de relé para gerar uma oscilação sustentável como alternativa da técnica de oscilação cíclica de ZN. Neste teste, o controlador é temporariamente substituído por um relé com histerese. Como resultados são determinados o ganho crítico e frequência crítica. Seborg et al. (2004) constata que sujeitar processos lentos a 2 ou 4 ciclos de oscilação, conforme requeridos pelo método, pode não ser possível. Este é um método muito utilizado principalmente para sistemas de auto-sintonia em controladores. Em razão de sua larga aplicação, foram sugeridas várias modificações no método para sistemas não-lineares, malha aberta instável, e sistemas com múltiplas entradas e múltiplas saídas (YU, 1999; HANG et al. 2002).

Os métodos de alocação de pólos são usados principalmente em sistemas de baixa ordem. Uma aproximação possível é a utilização de um sistema de segunda ordem e, desta forma, especificar uma taxa de amortecimento e frequência natural para o sistema. Esta especificação pode, então, ser preenchida pela alocação dos dois pólos do sistema para uma posição que determine o melhor desempenho da malha (OGATA, 2003).

Outro conjunto de métodos de sintonia são os métodos baseados em modelo interno. Podem ser citados os de síntese direta (SD), de controle por modelo interno (IMC) e método de *Lambda-Tuning*. Estes métodos são baseados na adoção de modelos do processo que conduzem a expressões analíticas para a sintonia de controladores. Os modelos SD e IMC

apresentam sintonias muito próximas, mas o IMC tem a vantagem de permitir incertezas no modelo. *Lambda-Tuning* é utilizado nos casos onde existem grandes atrasos de transporte ou quando o atraso é variante no tempo (SEBORG et al. ,2004).

Há ainda na literatura outros métodos baseados em técnicas de otimização. Os métodos de critérios de integrais de erro são os mais comuns e utilizam conceitos simples de otimização de uma função integral do erro. Aström e Hägglund, (1995) abordam que o método é adequado para controladores PID e que ele tem a vantagem de capturar diferentes aspectos dos problemas de desenvolvimento. Porém, o mesmo autor salienta que é necessária atenção na etapa de determinar os critérios e limites, senão o critério escolhido pode ter um valor ótimo, mas o controle continuar instável devido ao critério não ser eficaz para este controle. Zhuang e Atherton (1993) propuseram fórmulas de sintonia para o modelo FOPDT que mostraram melhores resultados quando comparadas a outras técnicas para as aplicações propostas. A contribuição de Visioli (2001) se faz na utilização dos métodos de critério de desempenho integral para sintonia de PID em modelos integrais e instáveis.

Estudos recentes de Ang et.al. (2005) apresentam as ferramentas de sintonia existentes no mercado de automação industrial. A maioria destes *softwares* é baseada na tecnologia OPC em plataforma Windows, o que os torna integráveis com a maioria dos equipamentos de sistemas industriais como tecnologias *fieldbus*. Alguns exemplos destes *softwares* são o INTUNE (CONTROLISOFT, 2007) e o ExpertTune (EXPERTTUNE, 2007). Outros *softwares* de sintonia existentes são *softwares* de sintonia proprietários baseados nos equipamentos do próprio fabricante, como DeltaV-Tune (EMERSON, 2007), PID-Tool (ALTUS, 2007), PID-Analyser (ATOS, 2007), entre outros. Para o *software* proprietário, as ferramentas servirão para sintonizar somente os controladores dos respectivos fabricantes.

Segundo a pesquisa de Ang et al. (2005), os métodos de sintonia mais utilizados são o IMC e o *Lambda-Tuning*. Também se destacam métodos baseados em otimização, tais como a integral do erro absoluto ponderado pelo tempo (ITAE), integral do erro quadrático ponderado pelo tempo (ITSE), e posicionamento dos pólos. Outra observação do mesmo autor é que a grande maioria dos *softwares* de sintonia somente trabalha com controlador PID. Algumas exceções são *softwares* que possuem sintonizadores para controladores baseados em sistemas Fuzzy e controlador de modelo preditivo (MPC). Outro aspecto observado é que embora existam alguns métodos proprietários de sintonia, a maioria dos

softwares utiliza os principais métodos encontrados na literatura científica. Os principais *softwares* de sintonia para sistemas industriais são apresentados na tabela a seguir.

Nome do produto	Baseado em modelo	Suporta operações <i>online</i>	Comentários
AdvaControlLoopTuner	-	Sim	Seleciona rápido, normal ou desempenho amortecido de malha fechada. Utiliza método de posicionamento de pólos.
IMCTune	Sim	Não	Utiliza método IMC.
Model ID & PID	Sim	-	Utiliza método IMC.
Robust PID Tuning	-	Não	Seleciona entre IMC, <i>Lambda-Tuning</i> com malha aberta ou fechada.
INTUNE	Sim	Sim	Utiliza IMC avançado.
Control Station	Sim	Não	Seleciona regulação ou seguidor de desempenho usando <i>Lambda-Tuning</i> .
DeltaV Tune	Sim	Sim	Seleciona desempenho de conservador para agressivo usando ZN modificado, margens de ganho e de fase, <i>Lambda-Tuning</i> , IMC e preditor de Smith.
Entech Toolkit Tuner	Sim	Sim	Utiliza <i>Lambda-Tuning</i> avançado.
PidTune	Sim	Não	Utiliza IMC.
Expertune	Sim	Sim	Seleciona regulação ou desempenho rastreado utilizando sintonia de $\frac{1}{4}$ ou 10% amplitude amortecimento, ou <i>Lambda-Tuning</i> .
TunePlus	Sim	Sim	Utiliza <i>Lambda-Tuning</i> e IMC.
TuneUP	Sim	Sim	Utiliza <i>Lambda-Tuning</i> e métodos de otimização.
TuneWizard	Sim	Sim	IMC ou <i>Lambda-Tuning</i>
RSTune	Sim	Sim	Utiliza Expertune.
Tune a Fish	Sim	Sim	Utiliza Expertune.
PIDEasy	Sim	Sim	Utiliza métodos de otimização proprietários.
GRAPHIDOR	Sim	Não	Gera gráfico 3D usando P, I utilizando método de minimizando de erro.
Profit PID	Sim	Sim	Utiliza algoritmo de otimização min-max proprietário.

Nome do produto	Baseado em modelo	Suporta operações <i>online</i>	Comentários
VisSimOptimizePro	-	Sim	Utiliza algoritmo de gradiente reduzido e generalizado.
PIDS	Não	Não	Seleciona o desempenho baseado em ITAE, ITSE, ISE ou IAE.
Ctrlab	Não	Não	Seleciona o desempenho baseado em ISE, ITSE, IST ² E ou margem de ganho e de fase.

Tabela 1 – Principais softwares de sintonia de PID comercial (ANG et AL., 2005)(1)

(1) Na tabela 1 a coluna “Baseado em modelo” refere-se aos *softwares* que identificam o sistema em malha aberta ou fechada e comparam a algum modelo matemático. A coluna “Suporta operações *online*” indica *softwares* que suportam operações *online* como sintonia *online*, amostragem de dados *online*, entre outros. O “-” significa que a informação não está disponível.

2.3. Sistemas SCADA remotos

Na supervisão remota, os sistemas SCADA são clientes ligados remotamente ao sistema de controle de chão de fábrica. Tipicamente as centrais de controle e os servidores de chão de fábrica estão localizados localmente dentro da planta industrial, enquanto que existem estações remotas espalhadas geograficamente uma das outras e que acessam os dados dos servidores. A figura a seguir mostra um exemplo de uma instalação típica de uma rede em ambiente industrial.

Como visto anteriormente para as redes locais, as conexões remotas entre os clientes e servidores também são baseados principalmente no meio físico *ethernet*, ligadas remotamente via *internet* através de estação servidora.

Com o desenvolvimento de aplicações de tempo real e *multicast* (i.e. transmissão de uma fonte para vários destinos ou de várias fontes para vários destinos), foi necessário desenvolver mecanismos para que as redes de comunicação pudessem prestar um serviço com uma qualidade de serviço diferenciada, uma vez que essas aplicações exigem uma qualidade mínima em termos de parâmetros temporais (como atraso e *jitter*) e capacidade efetiva de transmissão (como largura de banda). Vários protocolos para as diferentes camadas de rede surgiram para tentar melhorar o determinismo e qualidade de serviço, entre eles o RSVP (*Reservation Protocol*) que cuida apenas da reserva de recursos ao longo da rota entre os nós da rede. O protocolo RTP (*Real-time Transport Protocol*) provê serviços de sincronização, multiplexagem, e de segurança para transferência de dados, estes dois

principalmente voltados para tratamento de imagem e voz usando a *internet* (HANSSEN; JANSEN, 2003).

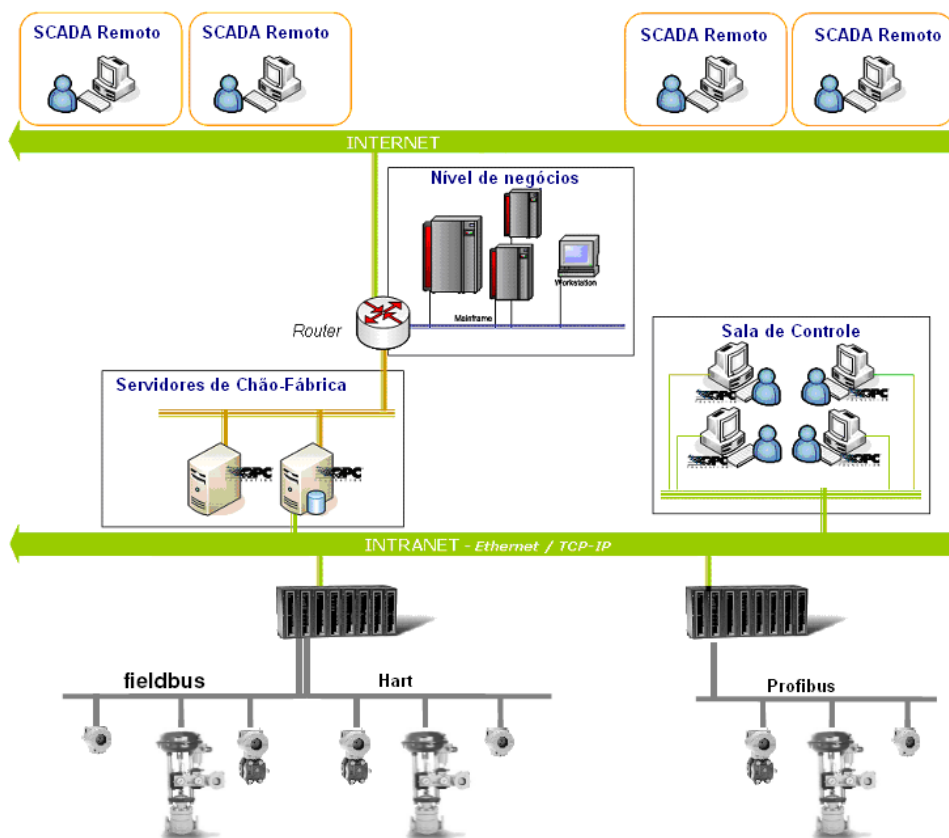


Figura 8 – Exemplo de uma aplicação industrial usando comunicação remota

Para as camadas de aplicação do modelo OSI, existem várias tecnologias de acesso aos dados do processo industrial, como por exemplo: ASP (*Active Server Pages*), utilizado em conjunto com os objetos ActiveX, e PHP (*Hypertext Preprocessor*), acessando bases de dados do processo que estão disponibilizadas em servidores SQL (*Structured Query Language*) (ZEILMANN, 2002). Para o acesso a aplicações distribuídas pode ser utilizado OPC com DCOM, como mencionado anteriormente. Também pode ser utilizada tecnologia de padrão aberto como XML ou JSON (*JavaScript Object Notation*) que são muito difundidas atualmente na comunicação na *internet*.

A OPC Foundation vem desenvolvendo desde 2003 um novo padrão OPC, baseado na tecnologia XML, chamada OPC UA (*Unified Architecture*). O OPC UA tem como objetivo a integração entre as várias especificações OPC existentes (DA, AE, HDA, DX, etc) em um único banco de dados facilitando o tratamento para os aplicativos (IEB, 2006). Além disso, o OPC UA propõe o suporte ao requisito de portabilidade, suportando ser

integrado em qualquer plataforma. Porém, esta tecnologia ainda se encontra em fase de aprovação e não existem equipamentos comercialmente lançados com este padrão.

As novas especificações do OPC UA mostram o caminho das tecnologias abertas como XML e JSON como grandes tendências em interatividade de sistemas industriais através da *internet* (TORRISI, 2007b).

2.3.1. XML e JSON

A XML é uma tecnologia criada inicialmente com o propósito de trocar informações via *internet* ou em sistemas distribuídos. Sua especificação 1.0 foi criada em 1998, adquirindo grande popularidade entre os profissionais da área de tecnologia da informação (THIRUVATHUKAL, 2004). XML é uma metalinguagem, isto é, uma linguagem usada para descrever outras linguagens. Existem atualmente várias linguagens derivadas da XML, pode-se citar a MathML, usada na matemática, e a SOAP (*Simple Object Access Protocol*), usada para troca de informações em sistemas distribuídos. XML é baseada em linguagens de marcadores, que seriam separadores de conteúdo (onde < indica o início e > indica o fim de uma seção). Uma linguagem de marcadores mundialmente conhecida é a HTML (*Hypertext Markup Language*).

JSON é um formato de troca de dados computacionais como o XML. JSON é baseado em um subconjunto da linguagem de programação JavaScript, Standard ECMA-262 (JSON, 2007). JSON é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C, incluindo C++, C#, Java, JavaScript, Perl, Python e muitas outras. Ele suporta apenas duas estruturas: “objetos” e “arrays” e também 4 tipos de dados: *strings*, números, booleanos e nulos. Estas propriedades fazem com que JSON seja um formato ideal de troca de dados.

Uma das vantagens do JSON sobre XML, como um formato para troca de dados neste contexto, é a facilidade verificada para escrever um analisador JSON. Na prática, os argumentos a respeito da facilidade de desenvolvimento e desempenho do analisador são raramente revelados devido aos interesses de segurança e à crescente integração de processamento XML nos navegadores modernos da *internet*. Por esta razão, JSON é tipicamente usado em ambientes onde o tamanho do fluxo de dados entre o cliente e o servidor é de grande importância, onde a fonte dos dados pode ser explicitamente confiável, e onde a perda dos recursos de processamento no lado cliente para manipulação de dados ou geração da interface não é uma considerada (TORRISI, 2007).

2.3.2. SOA e serviços *web*

Inicialmente a *internet* somente era composta de páginas, chamadas *web sites*. Estas páginas eram construídas usando HTML, armazenados em servidores e acessados pelo usuário final através de programas *browsers*. A evolução desta tecnologia são as arquiteturas orientadas a serviços (SOA – *Service Oriented Architecture*). (NEWCOMER; LOMOW, 2004).

A tecnologia SOA é estruturada para permitir que existam integrações de diferentes programas em diferentes plataformas possibilitando a construção de programas complexos e reusáveis (PHAM; GEHLEN, 2005).

O principal representante da arquitetura SOA são os serviços *web* (*web services*) que utilizam XML baseada em protocolo e descrições de dados. WSDL (*Web Services Description Language*) é uma linguagem baseada em XML que descreve os serviços, especifica o acesso e quais as operações ou métodos disponíveis nos serviços *web*.

De forma geral, a SOA define três regras, como mostrado na figura a seguir: um requisitor de serviço (R), um provedor de serviço (P) e um corretor de serviço (B). Um agente de *software* comunicando com outro agente de *software* pode executar uma ou mais regras de serviços.

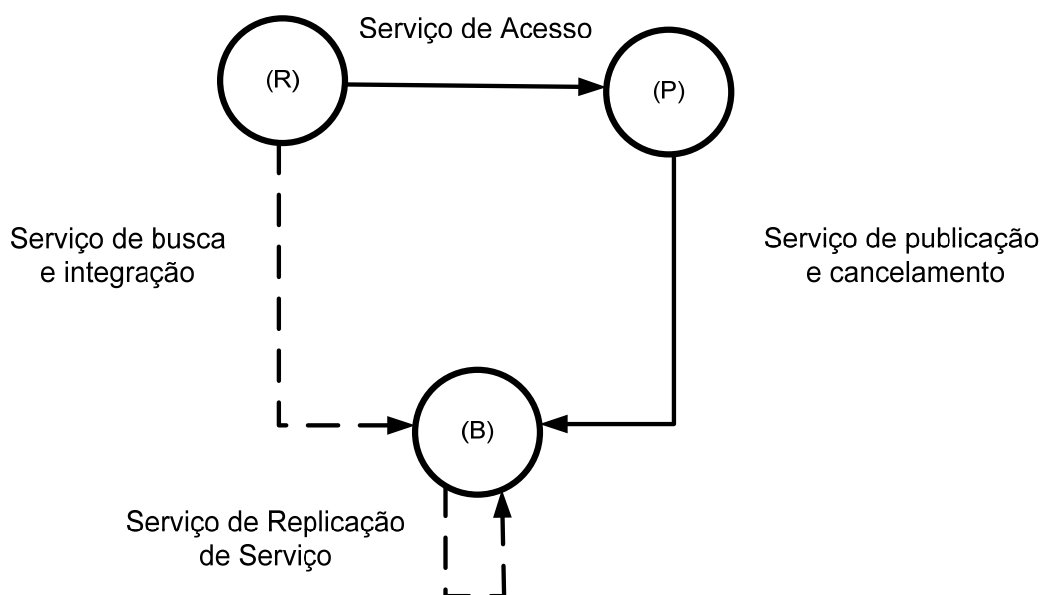


Figura 9 – Diagrama geral de uma arquitetura orientada a serviço (SOA)

Usando um documento XML criado na forma de uma mensagem, um programa envia uma requisição para um servidor *web* através da rede, e opcionalmente, pode receber uma resposta, também na forma de um documento XML. Na figura acima, o requisitor (R)

usa o corretor de serviço (B) para procurar por serviços e os integra acessando a descrição do serviço. Esta descrição inclui toda a informação necessária para acessar o serviço e é usada para gerar um objeto *proxy*. Por outro lado, o provedor (P) publica seu serviço para um registrador de serviço (corretor de serviço). Quando existir mais de um corretor de serviço no ambiente deverá ser replicado os registros do serviço. Desta forma, a tecnologia de serviços *web* provê uma forma padrão de conexões das diversas camadas de integração do programa.

Por utilizar protocolos padrões, os serviços *web* permitem que clientes de serviços requisitem provedores de serviços em plataformas computacionais diferentes. Uma outra vantagem é que os serviços *web* podem trafegar sobre o HTTP (*HyperText Transfer Protocol*), um protocolo simples e largamente difundido e que geralmente é liberado nos *firewalls* corporativos, permitindo que serviços sejam requisitados através da *internet* (KAPSALIS et al., 2002).

No contexto de aplicações de controle de processos as aplicações utilizam os serviços padrões descritas anteriormente. Pelo fato de não ser dedicado as aplicações específicas de controle e supervisão pode aparecer problemas mostrados a seguir.

2.3.3. Problemas atuais de supervisão e controle baseado na *internet*

A *internet* e os serviços *web* difundidos atualmente oferecem alguns obstáculos para o uso com propósitos de sistemas de controle industriais.

- *Atraso na comunicação* - A *internet* sofre de diversos tipos de atrasos durante todo o caminho da origem ao destino dos dados. Os principais atrasos são o atraso de processamento, o atraso de fila, o atraso de transmissão e o atraso de propagação para cada nó da rede. O atraso de processamento refere-se ao processamento interno de um *software* para examinar a mensagem e determinar para onde enviá-la ou verificar se existem erros na mensagem. O atraso de fila acontece no enfileiramento da mensagem esperando o momento da transmissão. O atraso de transmissão se refere ao tempo do dado chegar ao equipamento e, então, ser transmitido na rede. Por fim, o atraso de propagação se refere ao tempo de propagação da mensagem na linha. Segundo Han et al. (2001) o tempo de atraso da *internet* T_a no instante k pode ser descrito por:

$$T_a(k) = \sum_{i=0}^n \left[\frac{l_i}{C} + v_i^R + \frac{Q}{r_i} + v_i^L(k) \right] = d_N + d_L(k) \quad (4)$$

Onde, l_i é a distância do i -ésimo link da rede, C é a velocidade da luz, v_i^R é a velocidade do i -ésimo roteador, Q é a quantidade de dados, r_i é a largura de banda do i -ésimo link e $T_a(k)$ é o atraso causado pela carga do i -ésimo nó da rede.

Separando os termos dependentes e independentes do tempo tem-se d_N a parte independente do tempo e d_L a parte dependente.

A contribuição de cada componente do atraso pode variar significativamente. Por exemplo, o tempo de propagação pode ser desprezível na comunicação entre dois roteadores dentro do mesmo laboratório, contudo poderá variar de forma relevante para equipamentos interligados por um enlace de satélite e ser o termo dominante no tempo total de atraso (KUROSE e ROSS, 2006).

De acordo com um estudo feito por Luo e Chen (2000), o desempenho associado ao tempo de atraso e à perda de dados mostra uma variação temporal e espacial grande. O atraso médio das mensagens aumenta linearmente com o aumento do tráfego de acordo com Boggs et al. (1988).

- *o não determinismo da rede* – a rede *internet* é formada por várias sub-redes e vários roteadores entre a estação de origem e a estação de destino. A função destes roteadores é selecionar a rota mais apropriada para o tráfego de mensagens entre estes dois pontos. O algoritmo de roteamento varia de acordo com as variações de estabilidade de *hardware* e *software* de toda a rede. A decisão da melhor rota a ser utilizada deverá ser tomada para cada pacote de dados recebidos. Desta forma, não se garante o determinismo da rede (KUROSE; ROSS, 2006).

Contudo muitas técnicas têm sido desenvolvidas para suportar tráfegos de tempo real em particular para *ethernet*. Entre estas técnicas destacam-se garantias de largura de banda colocando restrições no tráfego entre os nós da rede, como nos trabalhos de Molle e Kleinrock (1985) onde é proposto um virtual CSMA, e no trabalho de Venkatramani e Chiueh (1994), propondo uma técnica *token-ring* sobre a *ethernet*. Outra forma é atuando nos *switches* e roteadores como no trabalho de Loeser e Haertig (2004) que propõe a utilização em conjunto de *switches* inteligentes e gerenciamento de tráfego. Por fim utilizando técnicas de gerenciamento de colisões sobre a *ethernet* (WANG, 2000).

Yang (2002) relata que a inclusão da *internet* nos níveis de sistemas de controle industriais seria inviável, devido à necessidade do determinismo na rede, pois a *internet* é altamente não-determinística com atrasos substanciais.

- *a segurança dos dados da rede* - é fundamental para os dados trafegados que os sistemas de controle distribuído atendam aos requisitos de comunicação segura. Neste caso, é necessário que sejam atendidas as seguintes propriedades de segurança: confidencialidade, autenticação e integridade das mensagens. Na confidencialidade é esperado que somente o remetente e o destinatário da conexão devam entender o conteúdo da mensagem. Na autenticação é necessário que tanto a origem quanto o destino confirmem a identidade da outra parte envolvida na comunicação. Na integridade é desejado assegurar que o conteúdo da comunicação não seja alterado durante a transmissão (KUROSE; ROSS, 2006).
- *Latência de serviços web* - Apesar da grande vantagem da alta interoperabilidade, pois todas as entidades dos SOA usam linguagens comuns para descrição de serviço, mensagens e registros de serviços, a utilização do SOA traz problemas de latência e de espaço de memória necessários para utilização de serviços *web*, onde estas características são críticas dependendo da aplicação, segundo Pham e Gehlen (2005). Para aplicações industriais onde são necessárias comunicações síncronas e assíncronas podem aparecer efeitos de *jitter* (variação do atraso entre os pacotes sucessivos de dados) elevados devido ao processamento interno (TORRISI, 2007a).

Na figura a seguir são mostrados os diversos passos para troca de dados usando serviços *web* desde a chegada do pacote para o WSDL até a entrega do dado para a camada de aplicação. A camada de aplicação representa a fronteira entre o cliente OPC e o servidor OPC localizado localmente ou remotamente.

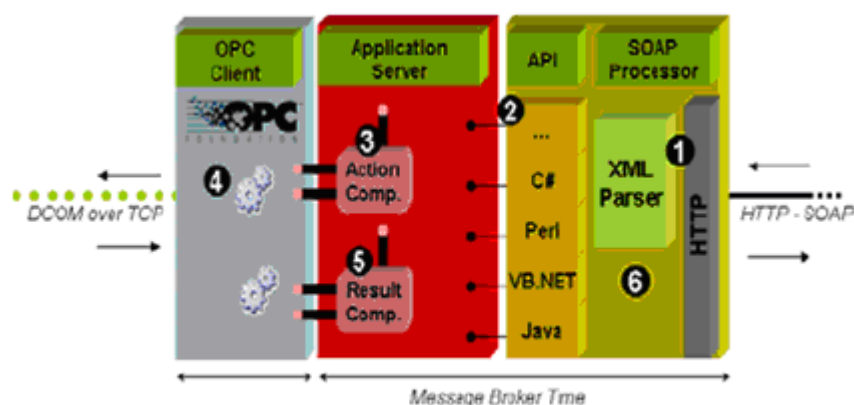


Figura 10 – Exemplo do caminho do dado dentro do servidor web (Torrison, 2007a)

Um pedido de requisição é feito pelo cliente de *internet* remoto através de uma chamada para o servidor *web*. Este receberá a requisição e transferirá para o processador Http-Soap do servidor (este processamento é mostrado na figura no passo 1). Então, é invocada a correspondente API (passo 2).que repassa para o servidor de aplicação (*Application Server*) a correspondente função requisitada (passo 3). Por último, o servidor de aplicação faz o pedido da mensagem para o cliente OPC tratar no protocolo OPC. Após isso esta mensagem é repassada para um servidor OPC. O servidor OPC fará a requisição do dado para um equipamento de campo que responderá e, desta forma, o ciclo é inverso e passará por todo o processo até a camada Http novamente (passo 6).

2.3.4. Tecnologia *CyberOPC*

Conforme explicação anterior, a tecnologia OPC é atualmente o padrão de comunicação mais utilizado na gestão de processos. Para oferecer soluções mais complexas com comunicação remota são combinadas com o OPC as tecnologias de *internet* como serviços *web*, XML e SOAP (ZHENG; NAKAGAWA, 2002). Porém, segundo Torrisi (2007b) esta solução não foi desenvolvida para atender aos requisitos de segurança e desempenho que são necessários para o ambiente industrial. Desta forma, o mesmo autor propôs uma nova forma de comunicação remota sem utilização de serviços *web* chamada *CyberOPC*.

A proposta do *CyberOPC* é desenvolver um novo protocolo de transporte de dados do processo com as seguintes características:

- Diminuição dos atrasos de transportes para dados críticos em tempo;
- Garantir a segurança do canal de comunicação usado;
- Garantir a integridade e confidencialidade das mensagens transmitidas.

Para obter a máxima interoperabilidade com as já existentes tecnologias de chão de fábrica, foram utilizadas tecnologias de padrão aberto, como o OPC. A comunicação do *CyberOPC* prevê o uso de uma estação *gateway* chamada *CyberOPC gateway* que processa as mensagens enviadas para o OPC através da rede pública e vice-versa.

Devido à simplicidade de comandos do *CyberOPC*, o “*Parser*” que possui as regras para reconhecer estes comandos é mais simples que qualquer *parser* XML para mensagens SOAP. Desta forma, os comandos OPC são executados rapidamente e, no caso de uma requisição periódica, é possível aumentar o tempo de resposta usando uma *cache* OPC dedicada compartilhada pelo cliente OPC e o HTTP *Broker*. Uma rápida *cache* de dados

OPC pode ser escrita assincronamente pelo cliente OPC para toda requisição periódica de dados e pelo cliente remoto da *internet*, como mostrado na figura a seguir.

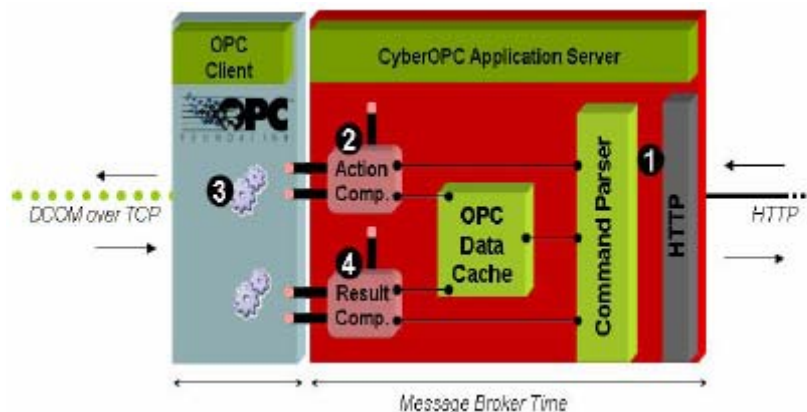


Figura 11 – Exemplo do caminho do dado dentro do CyberOPC (Torrise, 2007a)

Uma requisição do cliente é recebida no *gateway* (passo 1) que agora não tem o bloco processador SOAP. A introdução da *cache* OPC reduz fortemente o tempo utilizado com a chamada do cliente OPC. Testes realizados por Torrise (2007b) mostraram uma redução significativa na otimização no tempo de mensagem comparado com o *gateway* baseado em serviços *web*. Os passos 2, 3 e 4 representam a interação entre a biblioteca OPC e a camada do servidor de aplicação do *CyberOPC*.

Como o sistema proposto inclui a utilização desta tecnologia de comunicação do *CyberOPC*, este será explicado em maiores detalhes no capítulo de desenvolvimento do projeto.

2.4. Pesquisas atuais de supervisão e controle sobre *internet*

A *World Wide Web* tem fornecido oportunidades para o desenvolvimento e análise de sistemas de controle através da *internet*, segundo estudos de Yu et al. (2004). Diversos trabalhos propõem o uso da *internet* em sistemas de controle em arquiteturas distintas. As arquiteturas de acesso remoto podem ser realizadas em diferentes níveis na hierarquia de controle: nível do processo, nível supervisor e nível de otimização global do sistema (YANG et al., 2003).

No nível do processo, a proposta é incluir o controle remoto ao processo, segundo os trabalhos de Overstreet e Tzes (1999) e Yang et al. (2007). Neste caso, a estrutura de controle discreto convencional precisa ser alterada para atender aos tempos variáveis da

internet. Luo e Chen (2000) realizaram uma análise de um controlador remoto via *internet*. Uma das dificuldades observada do controle via *internet* é o atraso da rede. Foi verificado que os tempos de leitura e escrita da rede *internet* aumentam com a distância, bem como dependentes dos números de nós e da ocupação da rede.

No nível supervisorio, a preocupação é com segurança e qualidade de serviço. Os trabalhos de Kunes e Sauter (2001) estão baseados no protocolo SNMP sob sistemas de tecnologia *fieldbus*. Esta arquitetura trabalha bem para operações de leitura e escrita e notificações assíncronas como alarmes. Porém, trabalha sobre protocolo UDP onde não é permitido o tráfego na maioria dos *firewalls*. Outra desvantagem do SNMP é possuir níveis baixos de segurança.

Yang et al. (2003) propõe um controle remoto no nível de supervisão para serviços que são independentes do atraso da *internet*, que seriam restritos a serviços acíclicos como mudança de SP e parâmetros de sintonia de um bloco PID. No trabalho de Yang et al. (2003) é apresentado um controle de parâmetros de supervisão virtual. Este trabalho mostra que este controle somente seria invocado quando a alteração de parâmetros como SP e parâmetros de sintonia do PID fossem solicitados e, então, seria enviado o dado para o controle. Neste contexto, é permitido múltiplo acesso concorrentemente através da resolução de possíveis conflitos. E também a segurança do processo como um todo é garantida já que é possível prover redundância e diagnóstico de falhas na comunicação remota. Ainda no nível supervisorio, outro enfoque seria fazer a identificação e sintonia remota.

Qin e Wang (2007) estudaram o controle de admissão para um servidor de *internet* onde é controlada a admissão ou a rejeição de requisições para o sistema. Também um método para variação do parâmetro linear (LPV) para a identificação e controle de um servidor *web* é proposto. Desta forma, a aproximação LPV sintoniza o modelo especificando as condições de carga da *internet*, permitindo que o sistema se adapte às variações de carga e as condições de operação.

As empresas atualmente oferecem algumas soluções de controladores (CLPs) com servidores *web* embutidos nos equipamentos, contudo estas soluções apresentam algumas limitações quando aplicadas a plantas complexas (CALVO et al., 2006). Por exemplo, no trabalho de Batur et al. (2000) é mostrada uma arquitetura de supervisão e sintonia remota utilizando um Controlador SLC 500 da empresa Allen Bradley. A proposta do sistema é utilizar as variáveis de medição com os respectivos tempos de amostragem para garantir maior determinismo na rede. Também é descrito um mecanismo de controle de acesso. A desvantagem do sistema é que ele é composto por uma solução totalmente proprietária,

dependente de todas as bases de *softwares* da empresa para conseguir a supervisão e sintonia do controlador.

Yang et al. (2007) apresenta uma arquitetura de manutenção de controle de processos baseada na *internet*. As características do sistema industrial estudadas incluem índices de desempenho, detecção de falhas e acertos no desempenho do controle degradado. A proposta é fazer a monitoração do índice de desempenho do sistema localmente e caso seja notada alguma variação no índice, será feita uma identificação do sistema e, então, enviado um pedido de análise e sintonia para as estações remotas. Na arquitetura proposta, são divididos os trabalhos considerados “pesados” no processamento local, como: cálculo do índice de desempenho e identificação do modelo. E os trabalhos “leves” são enviados para a análise remota, como resultados dos testes de desempenho e modelo do processo. Assim, a análise dos dados seria realizada por especialistas que fariam uma proposição de sintonia.

Várias instituições têm realizado pesquisas e fornecido aplicações de controle e de ensino à distância de sistemas de controle através da *internet*. Estes trabalhos dividem-se basicamente em dois níveis de interação: o conceito de laboratório virtual que agrega uma estrutura física desenvolvida e a sua posterior disponibilização na *internet*; e cursos de ensino à distância que também oferecem um elevado grau de interatividade, vindo estes possibilitar, em alguns casos, a realização de simulações de fenômenos físicos. Estes laboratórios virtuais permitem ao usuário sintonizar plantas de controle remotamente, seja através da planta simulada, ou através de uma planta real. Entre eles podem ser citados Aktan et al. (1996), Ko et al. (2005), Zeilmann et al. (2003) e Mossin (2007).

3. Identificação e sintonia de processos industriais

Atualmente o controlador PID ainda é o mais utilizado nas malhas de controle industriais. Cerca de 90% dos controladores existentes são PI ou PID. Estes se tornaram o padrão na prática da engenharia de controle, tendo sido implementados sob diferentes formas, desde aplicações individuais até integrantes de esquemas de controle distribuído. Dentre as várias características apresentadas, destacam-se a sua simplicidade, baixo custo e robustez (ASTRÖM, 2001). Observa-se que a sintonia inadequada dos parâmetros dos controladores PID industriais se constitui no principal problema de eficácia dos controles de processos.

A sintonia de controladores, geralmente, é realizada em quatro fases. Primeiramente é feita a identificação do modelo da planta. Posteriormente, ocorre a validação do modelo identificado. Então é realizada uma sintonia baseada no modelo. Por fim, ocorre a simulação do modelo do sistema comparando a sintonia atual. A figura abaixo representa uma arquitetura do processo de sintonia típica de um processo industrial.

Na figura 12 o SP é a referência do processo. A saída do controlador é a variável manipulada (MV) representada na figura como u que através do acionamento atua no processo. A resposta do processo (PV) é representada por y onde pode ser aplicado um filtro para o ruído de sinal. O módulo de identificação coleta os dados de y e u (após a aplicação do filtro) e propõe um modelo da planta para a sintonia da malha. Após a validação da nova sintonia os novos parâmetros do controlador são efetivados.

Na primeira fase, para a obtenção do modelo da planta em questão são usadas técnicas de identificação de sistemas. A identificação de sistemas é tratada, muitas vezes, como um problema de otimização onde se busca estimar, com a melhor precisão possível, modelos matemáticos representativos (CUNHA, 2004). A seleção destes modelos e os ajustes de seus parâmetros são influenciados por muitos fatores como: conhecimento a priori do sistema, propriedades do modelo, presença de ruídos e critérios a serem minimizados na identificação.

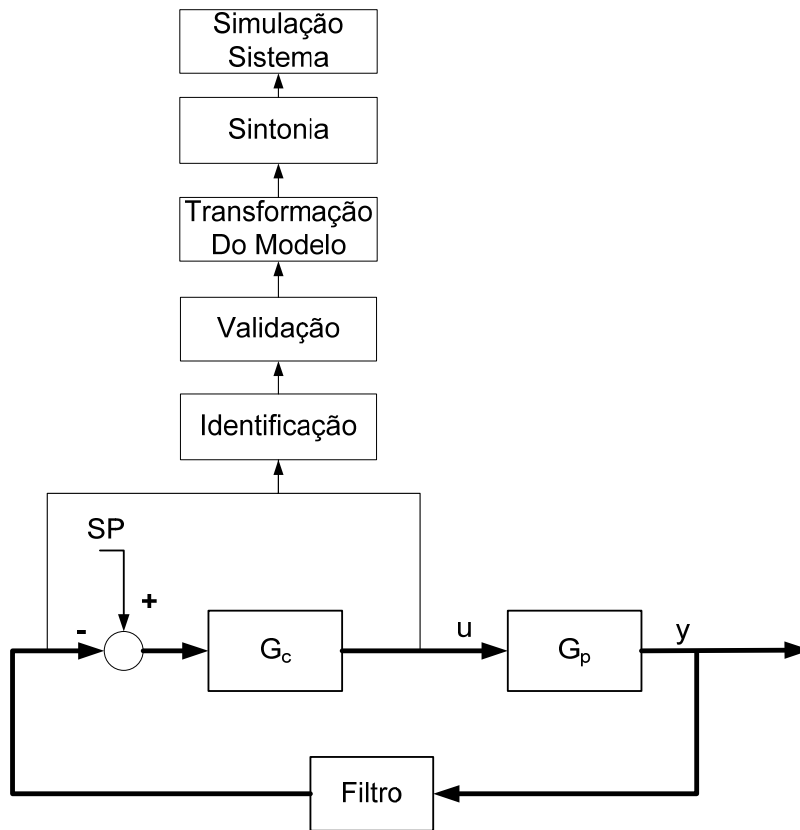


Figura 12 – Arquitetura de sintonia típica de processos industriais

Na segunda fase, realiza-se a sintonia do controlador baseado no modelo do sistema estimado. Vários métodos de sintonia são propostos na literatura desde o clássico método de Ziegler-Nichols (ASTRÖM; HÄGGLUND, 1995), até métodos avançados usando controle evolutivo com Algoritmos Genéticos e *Simulated Annealing*. Alguns métodos se aplicam muito bem a sistemas embarcados, que são simples e não necessitam de muito processamento, enquanto outros métodos são mais adequados a sistemas de supervisão SCADA (*Supervisory Control and Data Acquisition*), que necessitam de máquinas com maior poder de processamento (CAON, 1999).

Por fim, é feita a validação da sintonia utilizando como parâmetro um índice de desempenho obtido através de simulações do processo para somente, então, efetivar os novos parâmetros no controlador de fato.

O propósito deste trabalho é apresentar métodos de sintonia práticos que atendam os sistemas mais comuns no ambiente industrial, e que sirvam para o propósito principal do trabalho que está na sintonia remota de controladores. Somente serão abordados os métodos relacionados ao sistema SCADA e de sistema FOPDT, por ser o objetivo do trabalho em questão.

Para tal, inicialmente serão apresentadas as técnicas de identificação de sistemas, entre elas o método dos mínimos quadrados recursivo. Para a sintonia do controlador será mostrado um breve resumo dos modelos existentes e dado enfoque ao modelo de critérios integrais e IMC que serão os métodos abordados neste trabalho.

3.1. Identificação de sistemas

A identificação de sistemas realizada de forma experimental consiste na coleta dos sinais de entrada $u(k)$ e sinais de saída real $y(k)$ do sistema a ser identificado onde k representa o parâmetro de tempo discreto como mostrado na figura 12. Então, é estimado um modelo matemático baseado nos dados coletados, que representará o comportamento do sistema. O resultado da modelagem será uma função de transferência contínua ou discreta (LJUNG, 1999).

Para isso, diferentes procedimentos existem para geração destes sinais de entrada, coleta e armazenamento dos sinais de saída e estimação do modelo. Cada um destes procedimentos se aplica de forma mais satisfatória na identificação de um ou outro sistema e a decisão do melhor caminho dependerá do conhecimento a priori do sistema a ser abordado. Na seqüência será feita uma breve descrição destes procedimentos existentes.

No contexto de coleta e armazenamento de dados, Cunha (2004) classifica a modelagem de identificação de sistemas em:

- *Identificação offline*: excita-se o processo com sinais de teste apropriados e armazenam-se as medidas de entrada e saída do ensaio de identificação para estimação posterior dos parâmetros do modelo. Neste caso, necessita-se de coleta de dados e memória para armazenamento de dados, pois o cálculo dos parâmetros será realizado após o ensaio. Dentre as técnicas de modelagem *offline* podem ser destacadas: resposta em frequência, resposta temporal e identificação paramétrica.

- *Identificação online*: diferencia-se do tipo anterior pelo fato de ser um procedimento iterativo em que o cálculo dos parâmetros é feito simultaneamente ao ensaio de identificação. Não há necessidade de coleta de dados como no método anterior e métodos recursivos são empregados para atualização dos parâmetros da função de transferência, a cada período de amostragem da identificação. A identificação segue até a convergência dos parâmetros estimados. Para as técnicas de modelagem *online* são destaque: a identificação paramétrica usando mínimos quadrados recursivo, variáveis instrumentais, redes neurais artificiais entre outras.

No contexto da disposição das malhas de controle surge a questão sobre qual a melhor forma de se identificar um sistema: se a malha de controle deve permanecer fechada ou se deve ser aberta. No passado, a identificação em malha fechada era vista como algo que deveria ser evitado em virtude dos problemas de instabilidade e segurança da planta. Atualmente, os estudos de controle robusto para modelos mais avançados como MPCs trouxeram novas abordagens e vantagens entre os sistemas de malha fechada frente à malha aberta.

Miranda (2005) aponta as seguintes considerações entre as opções de malha aberta e malha fechada:

- Testes industriais em malha aberta deixam a operação exposta a distúrbios ao processo que podem levar a produtos fora de especificação. Testes em malha fechada são mais fáceis de serem realizados, já que a ação manual pode ser praticamente eliminada.
- A ação de realimentação tem a propriedade de reduzir as variações do processo e de permitir que sinais de maior amplitude sejam inseridos no processo sem aumentar a potência de saída, o que torna o sistema em malha fechada preferível em sistemas com restrições na saída.
- Testes em malha aberta podem despertar não-linearidades, o que pode levar a modelos não adequados para a região normal de operação.

No contexto do modelo dinâmico do sistema, é possível analisar o sistema através das relações entre as entradas e saídas durante uma resposta transiente. Isto se aplica a sistemas lineares invariantes no tempo (ASTRÖM; HÄGGLUND, 1995). A análise da resposta transiente caracterizada em termos da resposta a um sinal de entrada. De acordo com a resposta transiente os sistemas podem ser classificados em sistemas estáveis e sistemas integradores. Cada um destes sistemas terá características distintas de sintonia.

No sistema estável o processo se estabiliza mesmo sem um controlador. A saída sempre estabiliza em um novo valor após uma mudança na entrada. Segundo Aström e Hägglund (1995) a grande maioria dos sistemas industriais é estável. Exemplos deste tipo são sistemas de temperatura, de vazão, pH e pressão. Ele é geralmente representado por um modelo de primeira ordem mais tempo morto FOPDT:

$$G_p(s) = \frac{K_p}{\tau s + 1} e^{-\theta s} \quad (5)$$

Onde, ganho estático (k_p) – determina a sensibilidade do processo a uma mudança na entrada. Constante de tempo (τ) – Mede a velocidade em que a PV atingirá seu valor final. Tempo morto (θ) – tempo necessário para a PV começar a mudar, devido a uma mudança na MV.

No sistema integrador como mostrado em (6), o processo aumenta ou diminui o seu valor indefinidamente, após uma variação transiente. Exemplos desse tipo de sistema podem ser: controle de posição e controle de nível. No controle de nível, por exemplo, aumentando a vazão de entrada, o nível subirá constantemente até transbordar desde que a vazão de saída seja constante.

$$G_p(s) = \frac{K_p}{s(\tau s + 1)} e^{-\theta s} \quad (6)$$

Aström e Hägglund, (1995) ainda relatam que uma grande parte dos processos com controladores PID pode ser representada pelo modelo de três parâmetros (FOPDT), como em (5). O modelo FOPDT é bastante estudado na literatura como nos trabalhos de Zhuang e Atherton (1993); Hang et al. (1994), Zhi e Jingling, (1997), Ang et al. (2005), entre outros.

Neste trabalho, será abordada a identificação de sistemas no modelo de FOPDT que representa uma grande porcentagem de malhas industriais e uma sólida literatura sobre estes sistemas. Também será feita a identificação em malha fechada por ser uma tendência em sintonia em não perturbação do sistema principalmente na identificação *online*.

3.1.1. Etapas de identificação

Aguirre (2004) divide o processo de identificação em quatro etapas:

- Testes dinâmicos e as coletas de dados.
- Escolha do modelo matemático que representará o sistema.
- Estimação do modelo.
- Validação da estimação.

Caso não se obtenha um resultado esperado é feita nova estimação até que o modelo seja validado. Abaixo é detalhada cada uma destas etapas.

Testes dinâmicos e aquisição de dados

Nesta fase são obtidos os dados do processo de identificação. Muitas vezes estes dados são oriundos da operação normal do sistema. Outras vezes, é necessário excitar o sistema para obter as respostas dinâmicas. Alguns pontos importantes nesta etapa são a

determinação no sinal de excitação (forma e tamanho do sinal de entrada gerado) e a taxa de amostragem.

Como visto no capítulo 2, o sinal de excitação pode ser de resposta em frequência ou resposta a degrau. Nesta seção será escolhida a excitação por resposta a degrau, pois a abordagem é no domínio do tempo. Este caso consiste em aplicar uma variação do tipo degrau na entrada do processo armazenando-se os dados da saída conseqüente. Além do sinal do tipo degrau podem ser aplicados, também, outros sinais determinísticos como pulsos, rampas e PRBS (sinais binários pseudo-randômicos). Esta curva de reação obtida é submetida a técnicas gráficas ou numéricas para estimação do modelo do processo. Como o sinal degrau é um sinal com pobre composição frequencial, este método se aplica para processos de ordens inferiores (primeira e segunda ordem) (ASTROM; HAGLUNG, 1995).

Na identificação em malha aberta as perturbações geralmente costumam ser introduzidas na variável manipulada (MV), ou seja, na saída do controlador com este configurado para operar no modo manual. Já a identificação em malha fechada, a perturbação é introduzida no SP do controlador ou através de mudanças na saída do processo devido à mudança de carga. Vantagem esta que a identificação malha fechada tem sobre malha aberta, pois o processo pode ser mantido em controle durante todo o tempo do experimento. Porém, este procedimento pode não garantir um experimento suficientemente informativo, pois o controlador pode não movimentar a MV tanto quanto desejável. Sendo assim, é necessário verificar previamente alguns parâmetros do controlador como os limites operacionais das variáveis manipuladas e das variáveis controladas (MIRANDA, 2005).

Outro ponto importante na identificação consiste na determinação da taxa de amostragem. Para que um sinal amostrado retenha algumas das características fundamentais do sinal original, é necessário que o tempo de amostragem seja suficientemente curto. Na prática, a frequência de amostragem é escolhida entre 5 a 10 vezes maior do que a maior frequência de interesse contida nos dados (AGUIRRE, 2004). Para aplicações de controle remotas, a taxa de amostragem pode ser variável, e confronta com a estrutura de controle discreto convencional que pressupõe taxas de amostragem fixas (YANG et al., 2005).

Para o trabalho em questão, a tecnologia OPC não traz uma atualização de dados rápida sendo, em média, da ordem de segundos. Isto reduz o escopo do trabalho a sistemas mais lentos com constantes de tempo e tempo morto da ordem de 5 a 10 vezes o tempo de aquisição do OPC. Porém, muitos sistemas industriais se enquadram neste cenário como controle de temperatura de processos químicos que são sistemas lentos.

Ainda no contexto da coleta de dados, em qualquer experimento real haverá ruído. Segundo AGUIRRE (2004), em muitas situações o ruído pode ser desprezado quando a relação sinal ruído for bastante grande. Uma forma comum de reduzir os efeitos do ruído no sinal de entrada é utilizando a média dos dados através de filtro passa baixa da forma:

$$Y_{Filter} = \frac{1}{(T_{Filter} s + 1)} Y \quad (7)$$

Sendo Y_{Filter} o sinal de saída após aplicação de um filtro; T_{Filter} é o tempo do filtro de primeira ordem.

Na tecnologia *fieldbus* os blocos de entrada analógica e o bloco PID padrão possuem filtro de primeira ordem como mostrado em (6). Neste trabalho será utilizado o filtro de primeira ordem da tecnologia *fieldbus* para atenuação do ruído.

Escolha do modelo matemático

Nesta etapa é escolhida a representação matemática que será utilizada para representação do sistema real. Dentre estas, podem ser citados os modelos lineares polinomiais ARX (modelo auto-regressivo com entradas externas) e ARMAX (modelo auto-regressivo com média móvel e entradas exógenas), funções de transferência no tempo contínuo, entre outros. Para realizar esta escolha, é necessário ter um conhecimento prévio do sistema a ser identificado (AGUIRRE, 2004). Um exemplo de modelo linear polinomial (ARX) é descrito pela equação diferença:

$$\frac{y(k)}{u(k)} = \frac{b_1 u(k-1) + \dots + b_{nb} u(k-nb)}{a_1 y(k-1) + \dots + a_{na} y(k-na)} \quad (8)$$

Este modelo pode ser representado da seguinte forma:

$$y(k) = \sum_{i=1}^n \varphi_i^T(k) \xi_i + e(k) \quad (9)$$

e,

$$\begin{aligned} \varphi &= [y(k-1), \dots, y(k-na), u(k-1), \dots, u(k-nb)] \\ \xi &= [a_1, \dots, a_{na}, b_1, \dots, b_{nb}] \end{aligned} \quad (10)$$

Sendo,

$u(k)$ e $y(k)$ são os sinais de entrada e saída do sistema,

a_i e b_i são os parâmetros do sistema a ser identificado,

na e nb - representam a quantidade de regressores que estão associadas à ordem da equação,

φ - vetor de regressores contendo os valores de $u(k)$ e $y(k)$,

ξ - o vetor de parâmetros estimados de acordo com a estrutura escolhida.

$e(k)$ - o erro de predição, ou seja, a diferença entre a resposta estimada e resposta real e k o número de amostras.

Identificação paramétrica

Na fase de estimação paramétrica são estimados os parâmetros da estrutura matemática escolhida que representará o comportamento do sistema dinâmico a ser identificado. Para um sistema de uma entrada e uma saída SISO (*Single Input Single Output*) o algoritmo dos mínimos quadrados (MQ) é o método mais clássico para determinar os parâmetros ξ a cada amostra $y(k)$ e $u(k)$ no modelo definido em (9). O MQ é a base para muitas outras técnicas de identificação (LJUNG, 1999).

O método dos mínimos quadrados possui duas variantes de formulação: o caso não recursivo (MQ) e recursivo (MQR) que podem ser representados pela estimação *offline* e *online*. Abaixo segue a descrição dos dois métodos.

Inicialmente, considere a equação (9) na forma da equação diferença mostrada em (8). Considerando que seja aplicada uma entrada $u(k)$ variante no tempo desde o tempo zero ($k=1$) até o tempo $t(k=N)$, e considerando $n = na = nb$ a ordem do sistema a ser identificado, podem-se agrupar os dados das leituras em um vetor de saídas Y e uma matriz Ψ com os valores passados da saída e da entrada. Tem-se, então, um sistema matricial:

$$Y = \Psi\Phi + E \quad (11)$$

Sendo Y é o vetor de saídas com dimensão $N \times 1$ com N igual ao número de amostras coletadas. Ψ o vetor com as saídas e entradas passadas e tem dimensão $N \times 2n$. O vetor Φ com dimensão $N \times 1$ tem os coeficientes desejados para a identificação. Então para um sistema de ordem $n = na = nb$, tem-se:

$$y(k) = a_1 y(k-1) + \dots + a_{na} y(k-na) + b_1 u(k-1) + \dots + b_{nb} u(k-nb) \quad (12)$$

e,

$$Y = \begin{bmatrix} y(n) \\ y(n+1) \\ \dots \\ y(N) \end{bmatrix} \quad \Psi = \begin{bmatrix} y(n-1) & \dots & y(0) & u(n-1) & \dots & u(0) \\ y(n) & \dots & y(1) & u(n) & \dots & u(1) \\ \dots & & & & & \\ y(N-1) & \dots & y(N-n) & u(N-1) \dots u(N-n) \end{bmatrix} \quad (13)$$

$$\Phi = \begin{bmatrix} a_1 \\ \dots \\ a_{na} \\ b_1 \\ \dots \\ b_{nb} \end{bmatrix} \quad E = \begin{bmatrix} e(n) \\ e(n+1) \\ \dots \\ e(N) \end{bmatrix}$$

O que se deseja calcular é o vetor Φ sendo que se tem Y e Ψ . Tendo assim um problema de otimização que procura determinar Φ de forma que se minimize o somatório do quadrado de $e(k)$. Definindo-se a função J como erro quadrático médio, tem-se:

$$J = \frac{1}{2} \sum_{k=2}^N e(k)^2 = \frac{1}{2} E^T E \quad (14)$$

Então, calcula-se a derivada primeira da função J obtendo:

$$\Phi = (\Psi^T \Psi)^{-1} \Psi^T Y \quad (15)$$

A operação de (15) é realizada uma vez ao final da coleta dos dados e da montagem das matrizes Ψ e Y , caracterizando assim o processamento *offline*.

Já no método dos mínimos quadrados recursivo (MQR), os parâmetros são calculados recursivamente no tempo. Isto significa que quando houver uma estimativa de $\hat{\xi}(k-1)$ baseada em dados no instante $(k-1)$, então, $\hat{\xi}(k)$ será calculada a partir de $\hat{\xi}(k-1)$. Deste modo, a resposta atual do algoritmo será uma previsão baseada em parâmetros processados pela planta, com o sistema em funcionamento, caracterizando o processamento *online* (CUNHA, 2004).

Para o detalhamento do método, recorre-se à consideração do modelo definido em (9). A estimativa do vetor de parâmetros $\hat{\xi}$ pelo critério MQR pode ser calculada como no método MQ citado anteriormente:

$$\begin{aligned} \hat{\xi}(k) &= [\phi(k)^T \phi(k)]^{-1} \phi(k)^T Y \\ \hat{\xi}(k) &= \left[\sum_{i=1}^k \phi(i) \phi(i)^T \right]^{-1} \sum_{i=1}^k \phi(i) y(i) \end{aligned} \quad (16)$$

e,

$$\varphi(k) = \begin{bmatrix} \varphi(1)_{1 \times n}^T \\ \varphi(2)_{1 \times n}^T \\ \dots \\ \varphi(k)_{1 \times n}^T \end{bmatrix} \quad \phi(k+1) = \begin{bmatrix} \varphi(1)_{1 \times n}^T \\ \dots \\ \varphi(k+1)_{1 \times n}^T \end{bmatrix} = \begin{bmatrix} \phi(k) \\ \varphi(k+1)^T \end{bmatrix} \quad (17)$$

Sendo n é igual ao número de parâmetros do modelo a ser identificado.

Entretanto, deve-se encontrar uma maneira de atualizar a matriz $\hat{\xi}$ sem que seja necessário o cálculo da inversa de $\phi^T \phi$ a cada iteração do método. Para isso, introduz-se a matriz de covariância P :

$$P(k) = [\phi^T \phi]^{-1} = \left[\sum_{i=1}^k \varphi(i) \varphi(i)^T \right]^{-1} \quad (18)$$

e,

$$P^{-1}(k) = P^{-1}(k-1) + \varphi(k) \varphi^T(k) \quad (19)$$

Então,

$$\begin{aligned} \hat{\xi}(k) &= P(k) \left[\sum_{i=1}^{k-1} \varphi(i) y(i) \right] + \varphi(k) \varphi(k)^T \\ \hat{\xi}(k) &= P(k) \left[P(k-1)^{-1} \hat{\xi}(k-1) + \varphi(k) \varphi(k)^T \right] \\ \hat{\xi}(k) &= \hat{\xi}(k-1) + P(k) \varphi(k) \left[y(k) - \varphi(k)^T \hat{\xi}(k-1) \right] \end{aligned} \quad (20)$$

Obtêm-se, assim, as equações fundamentais que compõem o método recursivo:

$$\begin{aligned} \varepsilon(k)_{1 \times 1} &= y(k) - \varphi(k)^T \hat{\xi}(k-1) \\ K(k)_{n \times 1} &= P(k) \varphi(k) \\ \hat{\xi}(k)_{n \times 1} &= \hat{\xi}(k-1) + K(k) \varepsilon(k) \end{aligned} \quad (21)$$

Sendo $\varepsilon(k)$ o erro de predição, que é a diferença entre a saída medida e a saída do modelo predito baseada nos parâmetros do instante $(k-1)$. O vetor K é chamado ganho de Kalman. A matriz P é denominada matriz de covariância. A atualização de P e K é realizado conforme a seguinte equação (LJUNG, 1999):

$$\begin{aligned} P(k) &= P(k-1) - \frac{P(k-1) \varphi(k) \varphi^T(k) P(k-1)}{1 + \varphi^T(k) P(k-1) \varphi(k)} \\ K(k) &= \frac{P(k-1) \varphi(k)}{1 + \varphi^T(k) P(k-1) \varphi(k)} \end{aligned} \quad (22)$$

O erro de previsão torna-se zero quando $\hat{\xi} \rightarrow \xi$. A medida da qualidade do estimador pode ser inferida a partir da magnitude dos elementos da diagonal principal da matriz P , denominada como matriz de covariância do estimador.

Validação do modelo

Nesta fase é verificado se realmente o sistema estimado possui as características de interesse do sistema original. Esta fase é muito subjetiva e dependerá dos critérios de validação para as aplicações desejadas para o modelo.

Cunha (2004) cita como formas de validação do modelo o teste cruzado onde são utilizadas duas séries diferentes de dados, e estimam-se parâmetros para o primeiro conjunto. Então são calculados os valores estimados da saída do segundo conjunto baseado na estimação prévia feita pelo primeiro.

Uma forma de validação dos modelos é utilizando os critérios de Akaike, *Final Prediction Error* (FPE) e *Akaike's Information Criterion* (AIC). Os critérios de Akaike são utilizados para determinação da qualidade do modelo através de simulações com diferentes conjuntos de dados e diferentes modelos. De acordo com a teoria de Akaike, o melhor modelo é aquele que tem menor FPE (LJUNG, 1999).

Outra forma é através do índice de desempenho do erro absoluto (ITAE) (OGATA,2003). Neste caso, o menor índice significa o modelo mais próximo ao original. Outro índice de validação é percentagem da variação da saída (FIT) que representa a qualidade do modelo de 0 a 100%. O melhor modelo é aquele que tem o maior FIT (LJUNG, 2008). Valores de FIT baixos indicam pobre identificação. A equação abaixo mostra a fórmula de cálculo do FIT.

$$FIT = \left(1 - \frac{\sum_{k=1}^N \text{Norm} \left(y(k) - \hat{y}(k) \right)}{\sum_{k=1}^N \text{Norm} \left(y(k) - \bar{y} \right)} \right) 100 \quad (23)$$

e

$$\text{Norm}(V) = \sum_{k=1}^N \sqrt{\text{ABS}(V)^2}$$

Onde, $y(k)$ é a saída real do processo no instante k , $\hat{y}(k)$ é a saída estimada, obtida pela aplicação dos parâmetros obtidos, e \bar{y} é a média das N amostras de y ao longo da identificação.

3.2. Redução do modelo estimado

Muitas vezes para conseguir a identificação de sinais com ruído utilizando os modelos paramétricos, é necessário utilizar um modelo de mais alta ordem na identificação para obter uma boa aproximação do modelo. E após a identificação do modelo em alta ordem, os pólos e zeros utilizados para descrever o ruído do sinal poderão ser cancelados (LJUNG, 1999). Uma das formas de redução da ordem do modelo é utilizar somente os pólos dominantes.

O método utilizado para determinação dos pólos dominantes neste trabalho foi através do truncamento dos pólos, que para sistemas FOPDT estáveis, os pólos e zeros dominantes estão próximos à origem do lugar geométrico das raízes para o modelo contínuo. Para o modelo discreto ele corresponde à região próxima ao eixo positivo do círculo de raio um. Por exemplo, considere um sistema com função de transferência abaixo:

$$G_p = \frac{2}{100s+1} e^{(-50s)} \quad (24)$$

Considerando o sistema malha fechada com um controlador PI ISA com os parâmetros: $K_c = 0.9$ e $T_i = 100$. A equação do sistema na forma discreta para ($T_0 = 1[seg]$) resulta nos seguintes pólos e zeros:

$$\begin{aligned} \text{Pólos} &: [1.0700, (0.9490 \pm 0.0674i)] \\ \text{Zeros} &: [1.0516, 0.9899] \end{aligned} \quad (25)$$

Para uma identificação do sistema através do ARX de 10^a ordem ($na = 10, nb = 10, nk = 1$), obtém-se uma função de transferência com os seguintes pólos e zeros ($T_0 = 1[seg]$):

$$\begin{aligned} \text{Pólos} &: [0.9897, (0.9887 \pm 0.0241i), (0.2921 \pm 0.8135i), (-0.2575 \pm 0.8004i), \\ &(-0.6718 \pm 0.4862i), (-0.6718 \pm 0.4862i), -0.8257] \\ \text{Zeros} &: [1.0405, 1, (0.2922 \pm 0.8135i), (-0.2575 \pm 0.8005i), (-0.8259), \\ &(-0.6719 \pm 0.4864i)] \end{aligned} \quad (26)$$

Reduzindo a equação (26) para um sistema de ordem três serão considerados somente os pólos e zeros próximos a um. Portanto, a equação do sistema reduzido seria (26) para ($T_0 = 1[seg]$). Comparando (25) e (27) verifica-se que os resultados são muito próximos.

$$\begin{aligned} \text{Pólos} &: [0.9897, (0.9887 \pm 0.0241i)] \\ \text{Zeros} &: [1.0405, 1] \end{aligned} \quad (27)$$

3.3. Transformação de modelo matemático

Para se obter a sintonia do modelo estimado utilizando os métodos baseados em modelo, é necessário obter a função de transferência do sistema. O que foi feito até então na identificação foi a obtenção da equação diferença a partir dos dados coletados. Desta forma, ainda será necessária a conversão do modelo ARX para o modelo da função de transferência de FOPDT mostrado em (5).

Para isso, é proposta uma formulação matemática que resolva este problema para um sistema FOPDT, malha aberta ou malha fechada e controlador PID para mudança de carga (FERNANDES; BRANDÃO, 2008). Os dois métodos são descritos a seguir.

Considere o sistema em malha fechada típica mostrada na figura 13. No caso do estudo em questão tem-se prévio conhecimento das informações do controlador.

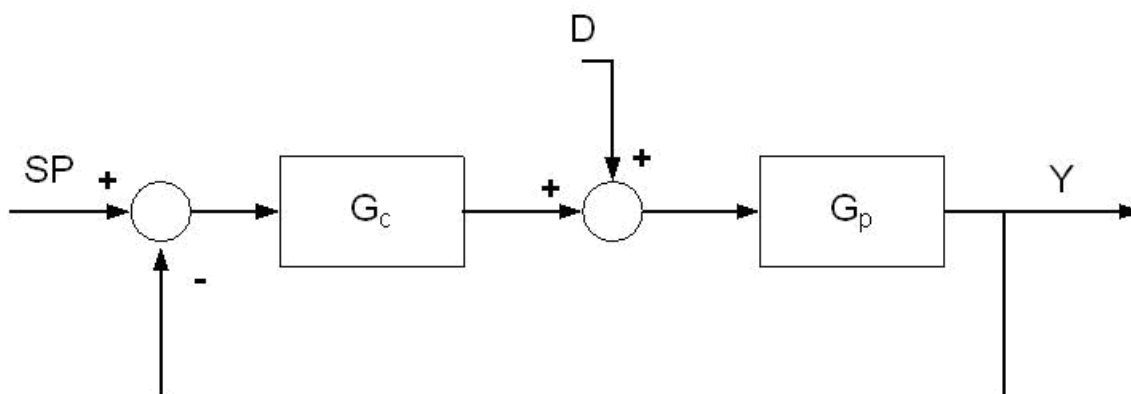


Figura 13 – Controle por realimentação clássica

A especificação de sintonia de um controlador pode ser classificada em variações devido à mudança de setpoint (SP) e distúrbios da variação de carga (D). O controle deve atuar no processo para atenuar estes distúrbios. Abaixo são definidas estas duas especificações:

- *Variações devido à mudança de SP* – Especificações devido à mudança de SP podem incluir requerimentos no tempo de subida, tempo de amortecimento, taxa de decaimento, *overshoot*, e variação no valor da saída em regime permanente.

- *Distúrbios devido à variação de carga* – Estes são os distúrbios que ocorrem no processo após ele estar em regime estável com SP constante. Distúrbios de carga são geralmente de baixas frequências.

O sistema real pode ser descrito pelas equações em malha fechada devido à mudança de SP (27) e mudança de carga (28):

$$\frac{Y}{SP} = \frac{G_c G_p}{1 + G_c G_p} \quad (28)$$

$$\frac{Y}{D} = \frac{G_p}{1 + G_c G_p} \quad (29)$$

Onde, G_c representa a função de transferência do controlador, G_p a função de transferência do processo, Y a saída do sistema, SP a referência do sistema e D um distúrbio gerado devido à mudança de carga.

Considerando um sistema de FOPDT como em (5). Como o tempo morto θ não é conhecido, ele pode ser estimado por uma aproximação de Paddé (SEBORG et al.,2004) da forma:

$$e^{-\theta s} \approx \frac{(1 - 0.5\theta s)}{(1 + 0.5\theta s)} \quad (30)$$

A aproximação do tempo morto tem maior imprecisão quando o tempo morto é relativamente grande em relação à constante de tempo do processo.

Considere um controlador PI padrão ISA, como descrito em (30):

$$G_c(s) = K_c \left(1 + \frac{1}{T_i s} \right) \quad (31)$$

Onde, K_c é o ganho proporcional e T_i a constante de tempo integral do controlador.

Para este trabalho considera-se possível obter os parâmetros K_c e T_i do controlador. Uma aplicação para este tipo de modelo seria a otimização de malhas de controle que já estejam em operação.

As equações do sistema para mudança de carga são obtidas substituindo (30), (31), (5) em (29), obtendo a seguinte equação no domínio contínuo:

$$\frac{Y}{D}(s) = \frac{(bs_1)s^2 + (bs_2)s}{(as_1)s^3 + (as_2)s^2 + (as_3)s + (as_4)} \quad (32)$$

Sendo,

$$\begin{aligned}
bs_1 &= (-0.5K_p T_i \theta) \\
bs_2 &= (K_p T_i) \\
as_1 &= (0.5\tau T_i \theta) \\
as_2 &= (\tau T_i + 0.5T_i \theta - 0.5K_p K_c T_i \theta) \\
as_3 &= (T_i + K_p K_c T_i - 0.5K_p K_c \theta) \\
as_4 &= (K_p K_c)
\end{aligned} \tag{33}$$

Para conversão do tempo contínuo para o discreto, utilizou-se a aproximação por integração retangular (*backward difference*) (SEBORG et al.,2004). A correspondente descrição na variável z é obtida como:

$$s = \frac{(z-1)}{T_0 z} \tag{34}$$

Onde, T_0 representa a taxa de amostragem do sistema.

Fazendo a conversão do tempo contínuo para o discreto, obtém-se:

$$\frac{Y}{D}(z) = \frac{(bz_1)z^2 + (bz_2)z + (bz_3)}{(az_1)z^3 + (az_2)z^2 + (az_3)z + (az_4)} \tag{35}$$

Sendo,

$$\begin{aligned}
az_1 &= (1/2)p_1 + p_2 + (1/2)p_3 - (1/2)p_4 + p_5 + p_6 - (1/2)p_7 + p_8 \\
az_2 &= (-3/2)p_1 - 2p_2 - p_3 + p_4 - p_5 - p_6 + (1/2)p_7 \\
az_3 &= (3/2)p_1 + p_2 + (1/2)p_3 - (1/2)p_4 \\
az_4 &= (-1/2)p_1 \\
bz_1 &= (-1/2)p_9 + (p_6/K_c) \\
bz_2 &= p_9 - (p_6/K_c) \\
bz_3 &= (-1/2)p_9
\end{aligned} \tag{36}$$

Onde, para (36) tem-se:

$$\begin{aligned}
p_1 &= (\tau T_i \theta) \\
p_2 &= (\tau T_i T_0) \\
p_3 &= (T_i T_0 \theta) \\
p_4 &= (K_p K_c T_i T_0 \theta) \\
p_5 &= (T_i T_0^2)
\end{aligned} \tag{37}$$

$$\begin{aligned}
p_6 &= (K_p K_c T_i T_0^2) \\
p_7 &= (K_p K_c \theta T_0^2) \\
p_8 &= (K_p K_c T_0^3) \\
p_9 &= (K_p T_i T_0 \theta) \\
p_{10} &= p_2 + (1/2)p_3 - (1/2)p_4 \\
p_{11} &= p_5 + p_6 - (1/2)p_7
\end{aligned}$$

A equação do modelo ARX estimado de terceira ordem ($na = nb = 3$) é da forma:

$$\frac{B}{A}(q) = \frac{B_1 q^{-1} + B_2 q^{-2} + B_3 q^{-3}}{1 + A_2 q^{-1} + A_3 q^{-2} + A_4 q^{-3}} \quad (38)$$

Que é equivalente ao sistema no domínio “z” da forma:

$$\frac{B}{A}(z) = \frac{B_1 z^2 + B_2 z + B_3}{z^3 + A_2 z^2 + A_3 z + A_4} \quad (39)$$

Igualando (35) e (39) e fazendo alguns ajustes, obtém-se:

$$B_1 = \frac{bz_1}{az_1}; A_2 = \frac{az_2}{az_1}; A_4 = \frac{az_4}{az_1} \quad (40)$$

E aplicando (39) e (35) em (40), resulta no seguinte sistema composto por três equações:

$$\begin{cases}
-(bz_3) - (bz_2) - B_1(az_3) + (B_1 p_1) - (B_1 p_{11}) - (B_1 p_8) = 0 \\
(-3/2 + A_2)p_1 - 2p_2 - p_3 + p_4 - A_2 p_8 - (A_2 + 1)p_{11} - A_2(az_3) = 0 \\
(-1/2 + A_4)p_1 - A_4(az_3) - (A_4 p_8) - (A_4 p_{11}) = 0
\end{cases} \quad (41)$$

Resolvendo o sistema de equações, as seguintes soluções são possíveis para o sistema em malha aberta:

$$\begin{cases}
K_p = \frac{c_{32}\theta^2 + c_{33}\theta + c_{34}}{c_{35}\theta^2 + c_{36}\theta + c_{37}} \\
\tau = \frac{c_{12} + c_{28}K_p - c_{26}K_p\theta + (1/2)c_{10}\theta}{c_9\theta - c_{10}} \\
c_{44}\theta^4 + c_{45}\theta^3 + c_{46}\theta^2 + c_{47}\theta + c_{48} = 0
\end{cases} \quad (42)$$

Sendo,

$$\begin{aligned}
c_9 &= \frac{(-A_4 - 1)T_i}{2} \\
c_{10} &= A_4 T_i T_0 \\
c_{12} &= A_4 T_i T_0^2 \\
c_{16} &= \frac{(-A_2 - 3)T_i}{2} \\
c_{17} &= (2 + A_2)T_i T_0 \\
c_{18} &= (1 + 0.5A_2)T_i T_0 \\
c_{20} &= (A_2 + 1)T_i T_0^2 \\
c_{21} &= B_1 T_i \\
c_{22} &= B_1 T_i T_0 \\
c_{23} &= B_1 T_i T_0^2 \\
c_{24} &= [-B_1 + (1/K_c)]K_c T_i T_0^2 \\
c_{25} &= B_1 K_c T_0^3 \\
c_{26} &= \frac{A_4 K_c T_0 (T_i + T_0)}{2} \\
c_{27} &= A_4 K_c T_0^2 (T_i + T_0) \\
c_{28} &= \frac{K_c T_0 (2T_i + A_2 T_i + A_2 T_0 + T_0)}{2} \\
c_{29} &= K_c T_0^2 (A_2 T_i + T_i + A_2 T_0) \\
c_{30} &= \frac{B_1 K_c T_0 (T_i + T_0) - T_i T_0}{2} \\
c_{31} &= c_{24} - c_{25} \\
c_{32} &= -0.5c_{16}c_{10} + c_{18}c_9 \\
c_{33} &= -c_{16}c_{12} + 0.5c_{17}c_{10} - c_{18}c_{10} + c_{20}c_9 \\
c_{34} &= c_{17}c_{12} - c_{20}c_{10} \\
c_{35} &= c_{28}c_9 - c_{16}c_{26} \\
c_{36} &= c_{17}c_{26} + c_{16}c_{27} - c_{28}c_{10} - c_{29}c_9 \\
c_{37} &= c_{29}c_{10} - c_{17}c_{27} \\
c_{38} &= c_{30}c_9 + 0.5c_{21}c_{26} \\
c_{39} &= c_{31}c_9 - 0.5c_{21}c_{27} + c_{22}c_{26} - c_{30}c_{10} \\
c_{40} &= -c_{22}c_{27} - c_{31}c_{10} \\
c_{41} &= 0.25c_{21}c_{10} + 0.5c_{22}c_9 \\
c_{42} &= c_{23}c_9 + (c_{22}c_{10} + c_{21}c_{12} - c_{22}c_{10})/2
\end{aligned} \tag{43}$$

$$C_{43} = C_{22}C_{12} - C_{23}C_{10}$$

$$C_{44} = C_{32}C_{38} - C_{41}C_{35}$$

$$C_{46} = C_{32}C_{40} + C_{33}C_{39} + C_{34}C_{38} - C_{41}C_{37} - C_{42}C_{36} - C_{43}C_{35}$$

$$C_{47} = C_{33}C_{40} + C_{34}C_{39} - C_{42}C_{37} - C_{43}C_{36}$$

$$C_{48} = C_{34}C_{40} - C_{43}C_{37}$$

Para obter a solução do problema da equação (41) é necessário resolver um polinômio de quarta ordem onde existem quatro soluções possíveis para o sistema. Uma forma de resolver este problema, utilizando recursos computacionais, é através da aplicação do algoritmo de Jenkins-Traub (JENKINGS; TRAUB, 1970).

Para a determinação da solução que mais se ajusta ao sistema é comparado o modelo obtido na identificação com cada solução obtida de (41). Para a comparação foi utilizado o critério integral do erro absoluto ponderado pelo tempo (ITAE). Ou seja, para cada solução obtida em (41) é aplicado um degrau unitário e comparado com o modelo estimado em (11) calculando o índice de erro. A solução com o menor erro será a solução escolhida.

Utilizando o mesmo conceito podemos obter as equações de um sistema malha aberta. Considere o sistema mostrado na figura 13. O sistema malha aberta é realizado colocando o controlador em manual.

O sistema real pode ser descrito pelas equações em malha aberta da forma

$$\frac{Y}{D} = G_p \quad (44)$$

Onde, G_p a função de transferência do processo, Y a saída do sistema, e D o distúrbio de excitação do sistema.

Para o sistema de primeira ordem mostrado em (5) e a aproximação de Paddé obtém-se a seguinte equação no domínio contínuo:

$$\frac{Y}{D}(s) = \frac{(-0.5K_p\theta)s + K_p}{(0.5\tau\theta)s^2 + (0.5\theta + \tau)s + 1} \quad (45)$$

Fazendo a conversão do tempo contínuo para o discreto, obtém-se:

$$\frac{Y}{D}(z) = \frac{(bz_1)z + (bz_2)}{(az_1)z^2 + (az_2)z + (az_3)} \quad (46)$$

Sendo

$$\begin{aligned}
az_1 &= 0.5\tau\theta + 0.5\theta T_0 + \tau T_0 + T_0^2 \\
az_2 &= -\tau\theta - 0.5\theta T_0 - \tau T_0 \\
az_3 &= 0.5\tau\theta \\
bz_1 &= K_p T_0^2 - 0.5K_p\theta T_0 \\
bz_2 &= 0.5K_p\theta T_0
\end{aligned} \tag{47}$$

A equação obtida do modelo ARX estimado de segunda ordem ($na = nb = 2$) é da forma:

$$\frac{B}{A}(q) = \frac{B_1q^{-1} + B_2q^{-2}}{1 + A_2q^{-1} + A_3q^{-2}} \tag{48}$$

Que é equivalente ao sistema no domínio discreto da forma:

$$\frac{B}{A}(z) = \frac{B_1z + B_2}{z^2 + A_1z + A_2} \tag{49}$$

Igualando (49) e (46), obtém-se:

$$B_1 = \frac{bz_1}{az_1} \quad ; \quad A_2 = \frac{az_2}{az_1} \tag{50}$$

Por fim, aplicando (47) e (49) em (50) e fazendo alguns ajustes são obtidos os valores de K_p , τ e θ para a sintonia do processo da forma:

$$\begin{cases}
K_p = \frac{(B_1 + B_2)[(0.5\tau\theta) + (0.5\theta T_0) + (\tau T_0) + (T_0^2)]}{T_0^2} \\
\tau = \frac{-[(0.5A_2\theta T_0) + (A_2T_0^2) + (0.5\theta T_0)]}{[\theta + T_0 + (0.5A_2\theta) + (A_2T_0)]} \\
[(-0.25A_2T_0) - (0.25T_0) - (0.25A_3T_0)]\theta^2 + \\
+ [(-0.5A_2T_0^2) - (A_3T_0^2)]\theta - (A_3T_0^3) = 0
\end{cases} \tag{51}$$

Como descrito anteriormente para o modelo malha fechada, para obter a solução do problema da equação (51) é necessário resolver um polinômio de segunda ordem onde existem duas soluções possíveis para o sistema. Para a determinação da solução que mais se ajusta ao sistema é comparado o modelo obtido na identificação com cada solução obtida do sistema em malha aberta. Para a comparação foi utilizado o critério integral do erro absoluto ponderado pelo tempo (ITAE).

3.4. Sintonia de controladores PID

Como visto no capítulo 2, vários métodos de sintonia são apresentados na literatura. Nesta seção serão mostrados dois dos métodos mais utilizados na literatura pesquisada e também em programas comerciais de sintonia de sistemas industriais baseados em sistemas de baixa ordem. Os métodos são o controle por modelo interno (IMC) (FRUEHAUF et al., 1994; RIVERA et al., 1986) e os métodos baseados em critério de desempenho de erro integral (ZHUANG; ATHERTON, 1993; VISIOLI, 2001; LIPTAK, 1995). Como vantagens destes métodos destacam-se baixo *overshoot* e bom tempo de acomodação.

3.4.1. Método IMC (*Internal Model Control*)

O nome controle por modelo interno vem do fato do controlador conter internamente um modelo do processo. O método é baseado no modelo representado na figura 14.

O modelo do processo \tilde{G}_p e a saída do controlador P são usados para calcular a resposta \tilde{Y} . A resposta do modelo é subtraída da atual resposta Y e a diferença $(Y - \tilde{Y})$ é usada como um sinal de entrada do controlador \tilde{G}_c . Em geral, $\tilde{Y} \neq Y$ devido aos erros de modelagem ($\tilde{G}_p \neq G_p$), e incertezas não conhecidas que não são computados no modelo. Abaixo segue a definição do modelo de acordo com Segorg et al., (2004).

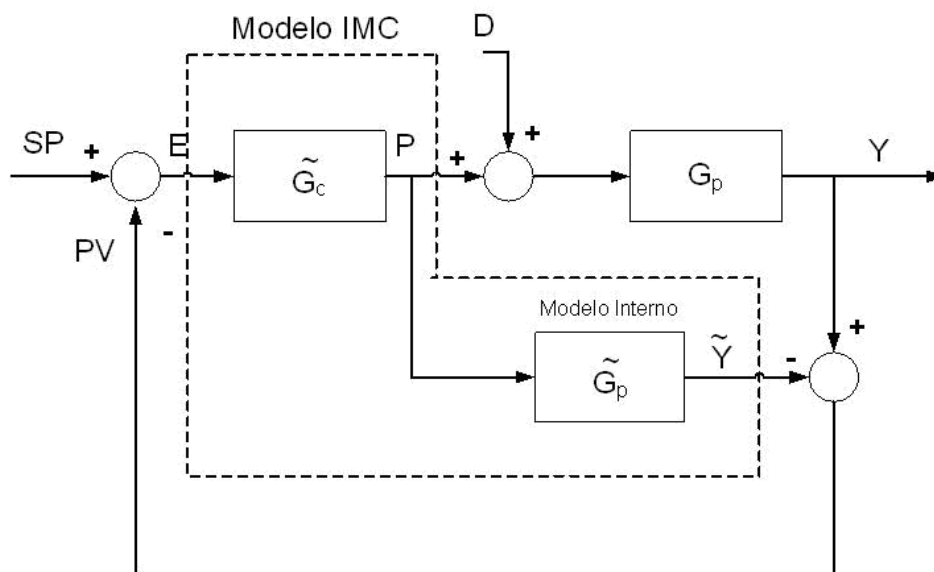


Figura 14 – Diagrama de blocos de um sistema de controle usando IMC

Considerando um processo com realimentação formado por um controlador G_c e o processo G_p como mostrado na figura 14, a equação em malha fechada do sistema considerando uma mudança de SP é definida em (27). Se G_c e \tilde{G}_c satisfazem a relação (52). Desta forma, qualquer modelo \tilde{G}_c é equivalente a um controlador de realimentação padrão G_c .

$$G_c = \frac{\tilde{G}_c}{(1 - \tilde{G}_c \tilde{G}_p)} \quad (52)$$

O desenvolvimento do controlador IMC consiste nos seguintes passos:

1. Assumir um modelo de processo \tilde{G}_p .
2. Fatorar o modelo em duas partes: uma parte inversível \tilde{G}_p^- e uma parte não inversível \tilde{G}_p^+ . A parte não inversível contém o tempo morto e o modelo ficará da forma:

$$\tilde{G}_p = \tilde{G}_p^- \tilde{G}_p^+ \quad (53)$$

3. O controlador é especificado como:

$$\tilde{G}_p = \frac{1}{\tilde{G}_p^-} f \quad (54)$$

Onde f é um filtro passa-baixa. Ele tipicamente tem a forma: (RIVERA et. Al. 1986):

$$f = \frac{1}{(\lambda s + 1)^n} \quad (55)$$

Onde λ é o parâmetro de sintonia especificado pelo usuário e n é um inteiro positivo. A escolha usual de $n=1$.

Aström e Hägglund (1995) citam que modelos de alta ordem que geram controladores de alta ordem, podem não ser viáveis na prática. Porém, para modelos de primeira e segunda ordem é possível obter bons controladores PI ou PID baseados no modelo IMC.

Para determinar um controlador IMC para um sistema FOPDT como em (5) e aproximando o atraso como em (29), obtém-se:

$$\tilde{G}_p(s) = \frac{K_p (1 - (\theta/2)s)}{(1 + (\theta/2)s)(\tau s + 1)} \quad (56)$$

Fatorando a equação (56) da forma de (54) e notando que \tilde{G}_p^+ precisa ter um ganho em regime como requerido no procedimento do controlador IMC:

$$\tilde{G}_p^+ = 1 - (\theta/2)s \quad (57)$$

$$\tilde{G}_p^- = \frac{K_p}{(1 + (\theta/2)s)(\tau s + 1)} \quad (58)$$

Substituindo (58) e (55) em (54) e considerando $n=1$, tem-se:

$$\tilde{G}_c(s) = \frac{(1 + (\theta/2)s)(\tau s + 1)}{K_p(\lambda s + 1)} \quad (59)$$

E a equação equivalente do controlador pode ser obtida substituindo (59) em (52):

$$G_c(s) = \frac{(1 + (\theta/2)s)(\tau s + 1)}{K_p(\lambda + (\theta/2)s)} \quad (60)$$

E arranjando para um controlador PID como em (1) obtém-se:

$$\begin{aligned} K_c &= \frac{1}{K_p} \frac{2(\tau/\theta) + 1}{2(\lambda/\theta) + 1} \\ T_i &= \theta/2 + \tau \\ T_d &= \frac{\tau}{2 * (\tau/\theta) + 1} \end{aligned} \quad (61)$$

3.4.2. Critérios de otimização baseados em índices de desempenho integrais

Ogata (2003) define que nos projetos de sistemas de controle, o sistema deve satisfazer as especificações de desempenho determinadas a priori.

Um índice de desempenho é um valor numérico positivo ou nulo que serve como indicativo da qualidade do desempenho do sistema. O sistema de controle é dito ótimo se os parâmetros são escolhidos de forma a minimizar ou maximizar o índice de desempenho escolhido, dependendo da situação.

Há muitos critérios com os quais se podem obter um sistema ótimo. Os mais amplamente usados são os critérios de erro nos quais o índice de desempenho é representado por integrais de uma função ponderada do desvio da saída do sistema em relação à entrada. Os valores das integrais devem ser obtidos como funções dos parâmetros do sistema, de

forma que o sistema ótimo seja obtido ajustando-se os parâmetros para fornecer, em geral, o menor valor para a integral.

Considerando-se a entrada e saída de um sistema de controle dada por $SP(t)$ e $Y(t)$, respectivamente, o erro é dado como:

$$e(t) = SP(t) - y(t) \quad (62)$$

Caso a condição $\lim_{t \rightarrow \infty} e(t) = 0$ não seja verdadeira, os índices de desempenho tenderão ao infinito. No caso do limite acima não tender a zero, pode-se definir:

$$e(t) = y(\infty) - y(t) \quad (63)$$

Com esta definição de erro, os índices de desempenho resultarão em números finitos, se o sistema for estável.

Vários índices de desempenho têm sido propostos na literatura onde os principais são IAE, ITAE, ISE, ITSE (OGATA,2003). Um exemplo típico do cálculo do índice de desempenho está representado na figura 15.

A tabela 2 apresenta os principais índices de desempenho e suas características, segundo Ogata (2003).

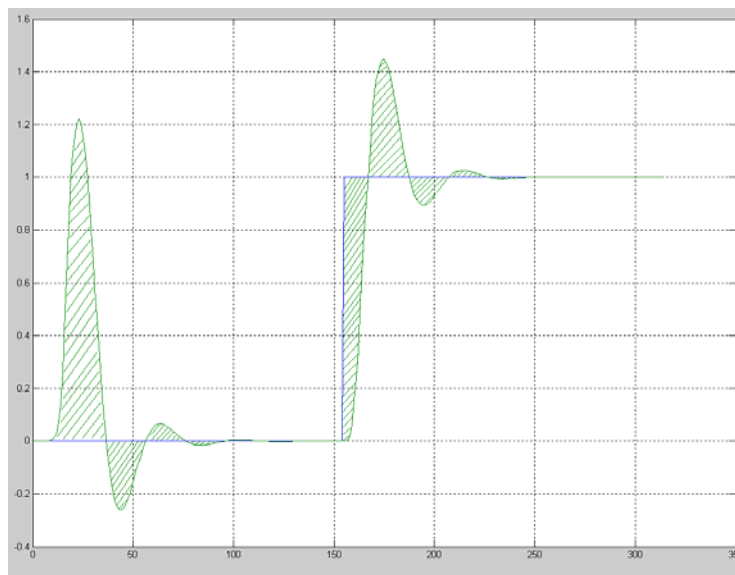


Figura 15 – Interpretação gráfica do IAE. A área hachurada é o valor de IAE

Critério	Fórmula	Descrição
ISE – integral do erro quadrático	$\int_0^{\infty} e(t)^2 dt$	Este critério proporciona grande peso para erros grandes e pequeno peso para erros pequenos. No entanto, ele não é muito seletivo, pois, para variações dos parâmetros do sistema escolhido a variação apresentada pela integral é pequena. Um sistema projetado que utilize este critério tende a apresentar uma rápida diminuição em um erro inicial grande. Portanto, a resposta é rápida e oscilatória. Desta forma, o sistema tem baixa estabilidade relativa.
ITSE - integral do erro quadrático ponderado pelo tempo	$\int_0^{\infty} t e(t)^2 dt$	Neste critério, para uma resposta a degrau unitário, um erro grande inicial é ponderado com peso baixo, enquanto os erros que ocorrem quando o sistema já está em regime são bastante penalizados. Este critério apresenta melhor seletividade que o critério ISE.
IAE – integral do erro absoluto	$\int_0^{\infty} e(t) dt$	Sistemas altamente subamortecidos ou sobreamortecidos apresentam um amortecimento razoável e uma característica de resposta transitória satisfatória.
ITAE - integral do erro absoluto ponderado pelo tempo	$\int_0^{\infty} t e(t) dt$	Um erro inicial grande em uma resposta a degrau unitário é ponderada com peso pequeno, e erros que ocorrem no tempo posterior são bastante penalizados. Este critério tem pequenas oscilações no sobre-sinal e as oscilações são bem amortecidas. Apresenta uma relativa melhoria sobre o critério IAE.

Tabela 2 – Principais índices de desempenho

3.4.3. Sintonia PID baseada em critérios de erros integrais

O método de otimização de critérios integrais também é tema de vários trabalhos de pesquisa na literatura. A maioria deles é baseada em sistemas FODPT, como descrito por (1). Os métodos de critérios integrais utilizam conceitos simples de otimização de uma função integral como mostrado nas equações da tabela 2 (ASTRÖM; HÄGGLUND, 1995).

A relação entre a sintonia do controlador e os critérios integrais é baseada na relação θ/τ (razão entre tempo morto e constante de tempo do sistema) e expressa pela relação de sintonia dada na equação a seguir:

$$X = m \left(\frac{\theta}{\tau} \right)^n \quad (64)$$

Sendo X um parâmetro do controlador (como Proporcional, Integral ou Derivativo) e m e n constantes.

Sintonia para ISE e ITSE

Em Zhuang e Atherton (1993) são apresentadas as fórmulas de sintonia para sistema FOPDT baseadas nos critérios de desempenho de integral usando ISE e ITSE. Abaixo são mostradas estas fórmulas para mudança de SP e para mudança de carga.

Para determinação dos parâmetros de sintonia devido à mudança de SP foram realizadas repetidas otimizações para diferentes valores para a razão θ/τ usando uma aproximação de Paddé para obter um PID ideal. A partir dos resultados, foram traçados gráficos das relações entre $K_p K_c$, τ/T_i , T_d/τ e θ/τ . Usando a técnica de mínimos quadrados, obtiveram-se as seguintes fórmulas baseadas nos resultados gráficos.

$$K_c = \frac{m_1}{K_p} \left(\frac{\theta}{\tau} \right)^{n_1} \quad (65)$$

$$T_i = \frac{\tau}{m_2 + n_2 (\theta/\tau)} \quad (66)$$

$$T_d = m_3 \tau \left(\frac{\theta}{\tau} \right)^{n_3} \quad (67)$$

Os valores dos coeficientes m e n na fórmula para controladores PI e PID com θ/τ na faixa de 0,1 a 1,0 e 1,1 a 2,0 são listados nas tabelas a seguir.

Faixa θ/τ	0.1 - 1.0		1.1 - 2.0	
	ISE	ITSE	ISE	ITSE
m_1	0.980	0.712	1.072	0.786
n_1	-0.892	-0.921	-0.560	-0.559
m_2	0.690	0.968	0.648	0.883
n_2	-0.155	-0.247	-0.114	-0.158

Tabela 3 – Sintonia PI para mudança de SP

Faixa θ/τ	0.1 - 1.0		1.1 - 2.0	
Critério	ISE	ITSE	ISE	ITSE
m_1	1.048	1.053	1.295	1.120
n_1	-0.897	-0.930	-0.619	-0.625
m_2	1.195	0.736	0.661	0.720
n_2	-0.368	-0.126	0.110	0.114
m_3	0.489	0.349	0.378	0.350
n_3	0.888	0.907	0.756	0.811

Tabela 4 – Sintonia PID para mudança de SP

As fórmulas de sintonia do controlador PID dos índices de desempenho ISE e ITSE para distúrbio devido à mudança de carga foram estimados da mesma forma que a entrada degrau. Desta forma, obtém-se:

$$K_c = \frac{m_1}{K_p} \left(\frac{\theta}{\tau} \right)^{n_1} \quad (68)$$

$$\frac{1}{T_i} = \frac{m_2}{\tau} \left(\frac{\theta}{\tau} \right)^{n_2} \quad (69)$$

$$T_d = m_3 \tau \left(\frac{\theta}{\tau} \right)^{n_3} \quad (70)$$

Os coeficientes m e n dos métodos ISE e ITSE para variação de carga são listados nas tabelas 5.

Faixa θ/τ	0.1 - 1.0		1.1 - 2.0	
Critério	ISE	ITSE	ISE	ITSE
m_1	1.279	1.015	1.346	1.065
n_1	-0.945	-0.957	-0.675	-0.673
m_2	0.535	0.667	0.552	0.687
n_2	-0.586	-0.552	-0.438	-0.427

Tabela 5 – Sintonia PI para variação de carga

Faixa θ/τ	0.1 - 1.0		1.1 - 2.0	
	ISE	ITSE	ISE	ITSE
m_1	1.473	1.468	1.524	1.515
n_1	-0.970	-0.970	-0.735	-0.730
m_2	1.115	0.942	1.130	0.957
n_2	-0.753	-0.725	-0.641	-0.598
m_3	0.550	0.443	0.552	0.444
n_3	0.948	0.939	0.851	0.847

Tabela 6 – Sintonia PID para variação de carga

Sintonia para IAE e ITAE

Liptak (1995) apresenta as fórmulas dos critérios integrais IAE e ITAE. Para a estimação dos parâmetros é considerado um modelo FOPDT como em (1). As fórmulas de sintonia são baseadas na relação mostrada em (64) e existem diferentes valores para variação de SP e variação de carga, sendo $X = K_p K_c$ para modo proporcional, $X = \tau/T_i$ para modo integral e $X = \theta/\tau$ para o modo derivativo

A tabela a seguir apresenta diferentes valores de sintonia para variação de SP e variação de carga para os parâmetros m e n da fórmula (64).

Critério	Controlador – distúrbio	Termo Proporcional		Termo Integral		Termo Derivativo	
		m	n	m	n	m	n
		IAE	PI - Mudança de carga	0,984	-0,986	0,608	-0,707
PI- Mudança de sp	0,758		-0,861	1,02	-0,323		
PID - Mudança de carga	1,435		-0,921	0,878	-0,749	0,482	1,137
PID - Mudança de sp	1,086		-0,869	0,740	-0,130	0,348	0,914
ITAE	PI - Mudança de carga	0,859	-0,977	0,674	-0,680		
	PI - Mudança de sp	0,586	-0,916	1,03	-0,165		
	PID - Mudança de carga	1,357	-0,947	0,842	-0,738	0,381	0,995
	PID - Mudança de sp	0,965	-0,855	0,796	-0,147	0,308	0,929

Tabela 7 – Relações de sintonia para variação de carga

Como exemplo para o cálculo de um controlador PI usando método ITAE segundo a tabela 7 para uma variação de carga seria obtido às fórmulas 71 e 72.

$$K_c = \frac{0,586 \left(\frac{\theta}{\tau} \right)^{-0,916}}{K_p} \quad (71)$$

$$T_i = \frac{\tau}{1,03 * \left(\frac{\theta}{\tau}\right)^{-0,165}} \quad (72)$$

Da mesma forma, para uma mudança de SP seriam obtidas as equações 73 e 74.

$$K_c = \frac{0,859 \left(\frac{\theta}{\tau}\right)^{-0,977}}{K_p} \quad (73)$$

$$T_i = \frac{\tau}{0,674 * \left(\frac{\theta}{\tau}\right)^{-0,680}} \quad (74)$$

4. Proposta de arquitetura para o Tele-Sintonizador de plantas industriais através da *internet*

De acordo com Zeilmann (2002), um *software* de supervisão e aquisição remoto deve possuir uma estrutura genérica de aquisição de dados através da *internet* para tentar atender a grande maioria de sistemas de automação industrial. Para que tal estrutura seja atendida, ainda segundo o autor, as seguintes características são desejáveis:

- Acesso remoto aos dados do sistema de automação industrial por clientes;
- Desempenho da aquisição de dados da rede, especificado na taxa de atualização dos dados;
- Garantia de segurança no canal de comunicação contra acessos não autorizados;
- Garantia da integridade e confiabilidade das mensagens transmitidas;
- Interface aberta de comunicação entre componentes de *software*, como requisito de escalabilidade do sistema;
- Independência dos dispositivos e protocolos de campo em operação;
- Independência de plataforma do PC do cliente remoto;
- Independência de plataforma do servidor do sistema de automação industrial.

Posto tais requisitos operacionais, será proposta, a seguir, uma estrutura genérica para a aplicação de tele-sintonia de malhas de controle regulatórias em equipamentos industriais que atenda aos requisitos considerados. Um dos pontos principais a ser observado será a flexibilidade do sistema, ou seja, a capacidade de adequação de qualquer módulo ou componente dentro da estrutura.

4.1. Arquitetura genérica de um Tele-Sintonizador

A arquitetura proposta para o Tele-Sintonizador é baseada na interconexão de módulos contidos em três contextos diferentes: a planta industrial, o servidor e o cliente, conforme figura 16. Ela é baseada no modelo de cooperação do tipo cliente-servidor

composta por módulos distintos interligados com o propósito de disponibilizar variáveis de processo e de configuração da planta industrial para o cliente remoto.

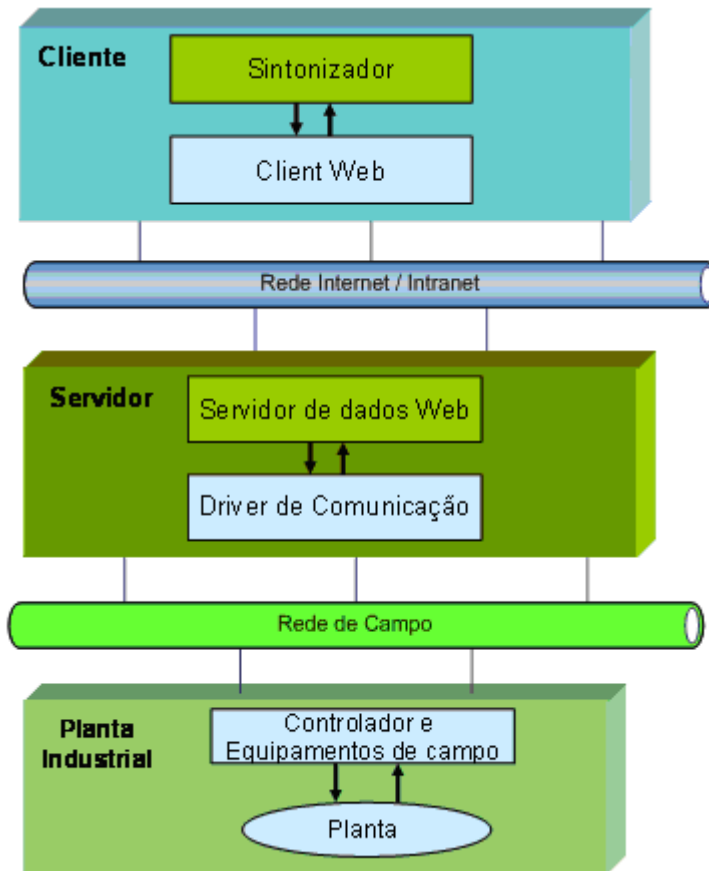


Figura 16 – Arquitetura do Tele-Sintonizador

A seguir são descritos os três contextos definidos na estrutura proposta e também são apresentadas as vantagens e desvantagens dos módulos propostos a fim de se definir uma estratégia de acesso à planta industrial que seja ideal.

4.1.1. Planta Industrial

Conforme destacado no capítulo 2, existem diversos protocolos de comunicação para equipamentos que atendem às aplicações específicas nos ambientes industriais atuais, como, por exemplo, controle de processos e controle da manufatura. Os meios físicos de comunicação entre estes equipamentos também diferem uns dos outros, seja em topologias possíveis, tipos de cabos, presença ou não de alimentação sobreposta à comunicação, adequação aos requisitos para uso em áreas classificadas, entre outros aspectos.

A arquitetura proposta neste trabalho prevê um canal de comunicação entre o controlador de campo e o *device driver* para aquisição de dados, este último instalado em

um computador. Como, por hipótese, o controlador ou os equipamentos de campo são genéricos, o protocolo de comunicação entre estes equipamentos e seu *device driver* é definido em função do equipamento ou do sistema utilizado. Pode-se, neste caso, utilizar um protocolo de comunicação proprietário ou aberto. Entretanto será necessário que o *device driver* possua, por sua vez, uma interface de *software* aberta e que seja facilmente integrável em qualquer componente de *software*.

Cabe aqui citar que, como o propósito do sintonizador se refere aos controladores do tipo PID, é necessário que o controlador ou os equipamentos de campo possuam um controlador do tipo PID em operação. Neste trabalho, optou-se pela validação baseada em controladores com algoritmo PID do tipo ISA definido em (1) por ser bastante difundido no ambiente industrial. A tecnologia utilizada foi a *Foundation Fieldbus* por possuir bloco PID padrão como descrito no capítulo 2. A rede de campo da figura 16 foi utilizada o meio físico ethernet. Entretanto, estas opções não excluem futuras adequações a outros algoritmos de controladores PID e outras tecnologias

4.1.2. Servidor

O servidor da rede é responsável pela interface entre a planta industrial e os clientes remotos. Como citado anteriormente, a arquitetura do sistema pressupõe o uso de um computador para realizar as tarefas do servidor.

O servidor é composto por vários módulos de comunicação, como apresentado na figura 16: o *device driver* de comunicação com controlador e os equipamentos de campo, o Sistema de Aquisição de Dados, e, também, o servidor de dados *web* para os clientes remotos.

O *device driver* é responsável pela aquisição dos dados do sistema de automação industrial e de sua posterior disponibilidade para outros aplicativos, possui, portanto, o conhecimento do protocolo (aberto ou proprietário) dos equipamentos de campo. Tais dados do sistema de automação serão tratados por aplicações em níveis mais elevados do sistema, como, por exemplo, o servidor de dados ou uma aplicação de supervisão e controle qualquer. Como apresentado no capítulo 2, existem diversos padrões para a aquisição de dados da planta industrial, como o padrão OPC e CORBA.

As vantagens do CORBA se apresentam em função deste padrão ser independente de plataforma, enquanto que o OPC possui vantagens devido à grande quantidade de *softwares* disponíveis com este padrão, ou seja, é atualmente o padrão “de facto” no setor industrial.

Pesquisas recentes têm como objetivo torná-lo independente de plataforma como é o caso do OPC UA visto no capítulo 2. Através do padrão OPC, a integração entre um SCADA e os equipamentos da planta industrial será transparente do ponto de vista do usuário. Portanto, neste trabalho foi utilizado a tecnologia OPC para comunicação entre o servidor e a planta industrial conforme indicado na figura 16.

O servidor de dados *web* da figura 16, é responsável pelo tratamento das requisições remotas dos vários clientes de *internet* conectados ao servidor. Conforme apresentado no capítulo 2, a comunicação remota entre o servidor *web* e seus clientes pode usar protocolos proprietários, serviços *web* e demais protocolos particulares sobre o HTTP.

Como destacado anteriormente, devido à generalização da comunicação dos servidores *web* em um âmbito mais amplo do que o de aplicações industriais com características de tempo real, a quantidade e complexidade de interfaces e camadas tornaram-se um empecilho para sistemas com requisitos de tempo “*soft real time*” pela *internet*, cumulativamente ao não determinismo nativo do protocolo TCP/IP. O *CyberOPC* é baseado em tecnologias padrões e abertas. Tem como premissa a minimização de camadas de *software* e de interfaces para uma melhor e mais ágil utilização da rede por se tratar de uma tecnologia com filosofia simples voltada para aplicações industriais com requisitos de “*soft real time*”. Uma das características deste protocolo é o uso de uma *cache* interna, que possibilita um maior desempenho no processamento de mensagens.

Neste projeto foi utilizado o padrão *CyberOPC* em conjunto com o padrão OPC para a aquisição de dados do servidor com os controladores ou equipamentos de campo. A figura 17 expõe os diagramas simplificados de arquitetura em blocos das duas estruturas propostas para o *CyberOPC gateway* que reside do lado do servidor.

Como mostrado na figura 17, o *CyberOPC gateway* é composto por uma parte comum chamada de servidor de aplicação (*Application Server*) que é responsável por todo o tratamento dos dados do servidor com os clientes da *internet*. Ele também realiza a aquisição de dados do barramento de campo através do *driver* de comunicação.

No primeiro nível do servidor de aplicação existe a camada do protocolo HTTP sobre SSL, também chamado HTTPS. Este representa o padrão para tratar da identificação e da autenticação de dois *hosts*, como, por exemplo, um cliente *CyberOPC* e servidor *CyberOPC*. Após a identificação da conexão, é realizada a decodificação dos comandos pelo módulo “*Parser JSON*”. Lembrando que no *CyberOPC* todos os dados de processos são codificados em formato JSON pela rede. Opcionalmente, pode ser usada uma *cache* de dados para todas as requisições de dados periódicos, o que otimiza o processo de aquisição

de dados. Então, o servidor de aplicação faz a requisição do pedido para a interface de acesso do barramento de campo.

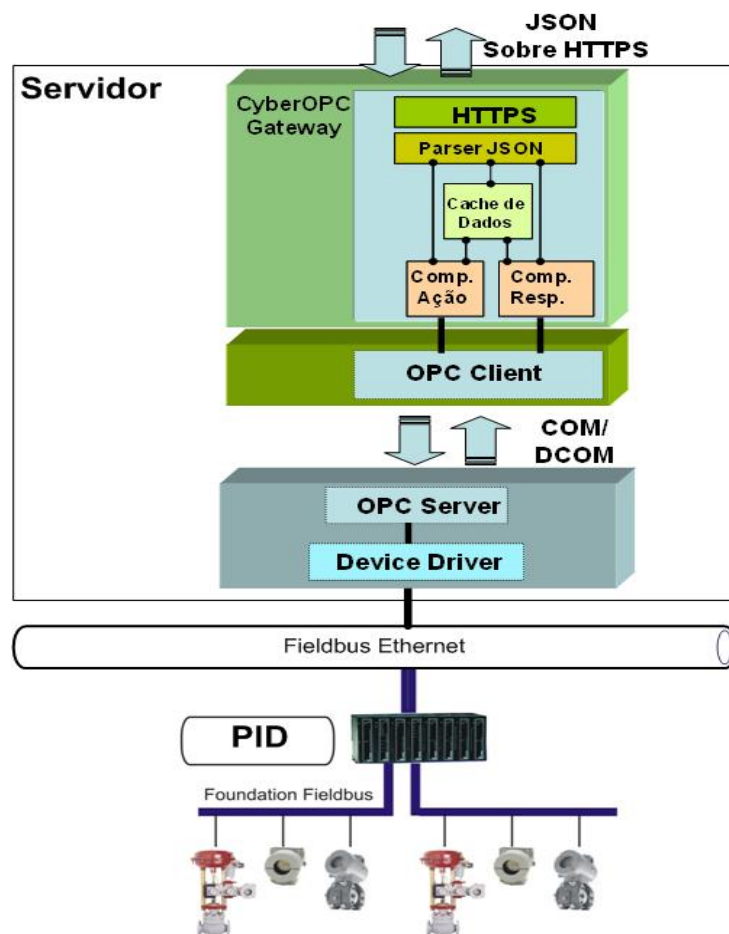


Figura 17 – Arquitetura do CyberOPC gateway no servidor de dados remoto da planta industrial

Para a interface de acesso entre o barramento de campo e o *CyberOPC* pode ser utilizado o OPC ou um *driver* nativo, como mostrado na figura acima. No caso do OPC, o cliente OPC se comunica com qualquer servidor OPC permitindo uma maior flexibilidade de integração com diferentes protocolos. Neste caso, o servidor OPC pode estar residente na mesma estação do servidor ou distribuído remotamente.

Ainda no contexto do servidor, é válido lembrar que todo o projeto e desenvolvimento da tecnologia *CyberOPC* foram realizados em um trabalho anterior do grupo de Automação Industrial da EESC e estão detalhados em Torrisi (2007). O tele-sintonizador proposto neste trabalho utiliza o *CyberOPC* como um cliente de uma aplicação. Para este último, tanto o projeto quando o desenvolvimento do protótipo será detalhado nas próximas seções.

4.1.3. O cliente remoto

No contexto da arquitetura proposta, o cliente representa a unidade remota de supervisão e sintonia conforme figura 16. O requisito do cliente é ser um cliente padrão OPC que possibilite a comunicação com as mais diversas redes de equipamentos. Neste trabalho foi utilizada uma arquitetura cliente OPC e cliente *CyberOPC*, o que facilita a implementação da comunicação tanto em ambiente local quanto remoto. Um exemplo da arquitetura completa do tele-sintonizador utilizando CyberOPC é apresentado na figura 18.

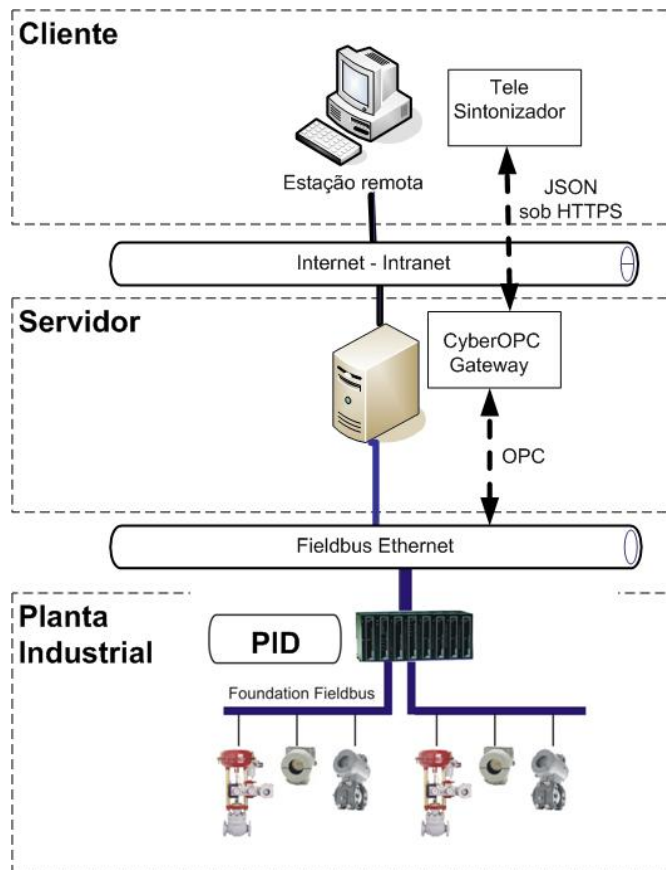


Figura 18 – Arquitetura do tele-sintonizador comunicando com a planta industrial remotamente através do CyberOPC gateway

Sistema de supervisão e sintonia

A sintonia de sistemas de controle é tipicamente composta pelas fases de aquisição de dados da planta, identificação do sistema, validação do modelo, eventualmente uma simulação da dinâmica da planta sintonizada para efeitos de verificação, sintonia da malha e efetivação dos dados na planta (CUNHA, 2004).

O tele-sintonizador aqui proposto é composto por quatro módulos operacionais principais: módulo de aquisição de dados, módulo de identificação do modelo ARX, o módulo de transformação do modelo ARX para malha aberta e o módulo de sintonia. A figura 19 ilustra as relações entre tais módulos.

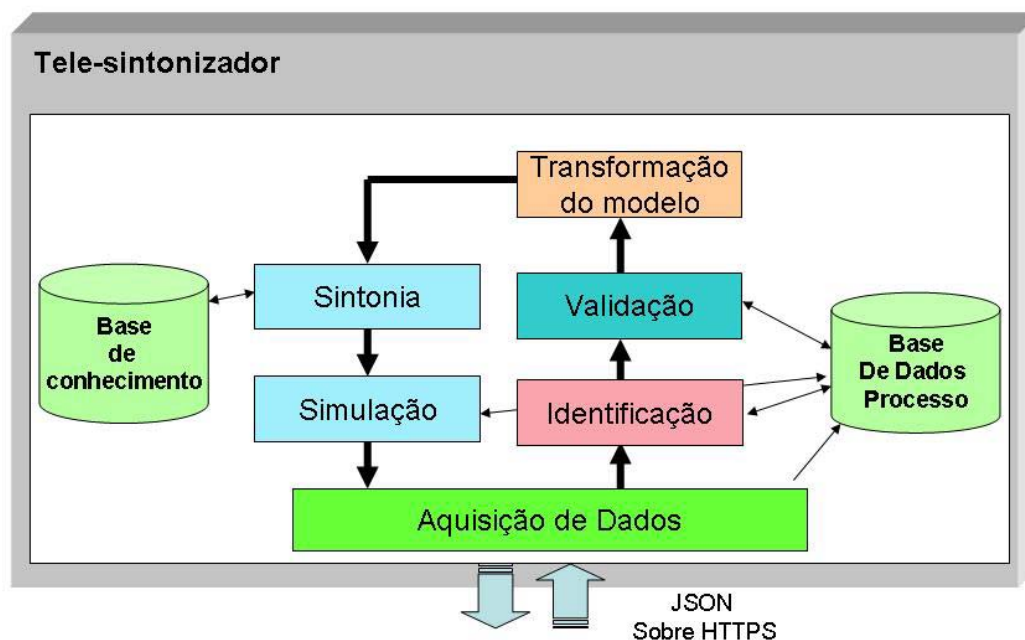


Figura 19 – Esquemático do Tele-Sintonizador cliente CyberOPC

O módulo de aquisição de dados é formado por um cliente OPC ou *CyberOPC* de acordo com as especificações do padrão OPC (OPC FOUNDATION, 2006) ou das especificações *CyberOPC* (TORRISI, 2007c). Desta forma, o componente de interface tem a mesma filosofia de acesso ao dado, composta de registro da biblioteca OPC, da adição de grupos e itens à base de dados e comunicação acíclica por evento, onde o cliente é notificado quando ocorre um novo evento de dados emitido pelo servidor.

Para o módulo de identificação do sistema, responsável pela determinação da função de transferência do sistema, foi utilizado o modelo ARX proposto no capítulo 3 que possui bons resultados para sistemas lineares de primeira e segunda ordem, além de ser bastante difundido na literatura consultada (CUNHA, 2004).

Como este projeto visa à validação da arquitetura *online* de identificação e sintonia de uma forma íntegra e confiável, a proposta principal do identificador é receber dados do processo e processar automaticamente a identificação através da identificação *online*. Porém, também é importante a identificação *offline*, onde se processa inicialmente uma coleta de dados e registro desta base de dados para posterior identificação e sintonia, que também foi desenvolvida neste projeto.

Para realização da identificação *online* remota são pressupostos atrasos de comunicação e também falhas de envio. Para evitar estes casos, este trabalho propõe a solução mostrada a seguir.

Primeiramente toda amostra coletada pelo *CyberOPC* tem registrado o horário (*timestamp*) do momento da aquisição do dado pelo *gateway*.

Segundo, como o modelo ARX precisa de uma amostragem contínua e o *CyberOPC* envia o dado de forma otimizada, por evento no momento da mudança do dado (*on data change*), é necessário reconstruir o sinal do processo a uma taxa de amostragem constante.

Para resolver este caso foi incluído o módulo de “*pre-identification*” que é responsável por receber o dado da fila do módulo de aquisição e amostrar para a fila de dados de identificação com uma taxa de amostragem constante. Para a conexão entre dois pontos da amostra é utilizada uma interpolação de primeira ordem. Um exemplo desta arquitetura é mostrado na figura 20.

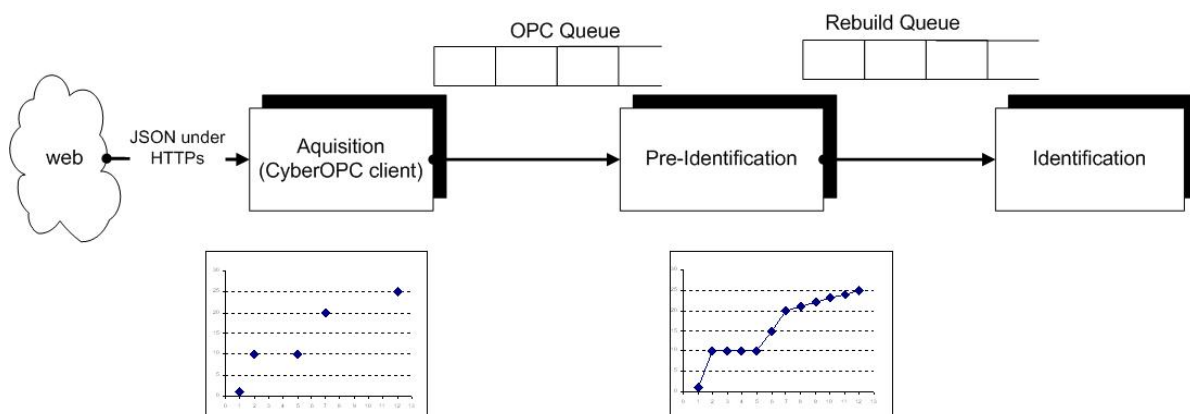


Figura 20 – Esquema dos tratamentos dos dados recebidos pelo OPC e CyberOPC

Para a validação do modelo identificado é utilizado o método do Teste Cruzado apresentado por Cunha (2004). Neste método é comparada a resposta gerada pelo modelo identificado e a resposta do sistema real, para um mesmo sinal de entrada. Durante a validação é, então, calculado o erro quadrático médio e o índice de percentagem da variação da saída como medida de desempenho e de validade do método.

O módulo de sintonia é responsável pela aplicação do método de sintonia no modelo determinado pelo módulo de identificação. Dos diversos métodos de sintonia difundidos pela literatura, foram considerados para este projeto os métodos que atendem aos requisitos de bom desempenho em sistemas de primeira ordem. Como não se trata aqui de um trabalho de validação de métodos de sintonia, será utilizado na experimentação prática do sistema

proposto apenas o método de otimização ITSE, ITAE e IMC devido à sua boa resposta em sistemas FOPDT. Contudo, o desenvolvimento de outros métodos poderá ser agregada de forma factível na base de conhecimento do sintonizador bastando atender aos requisitos dos sistemas de identificação.

A figura 21 apresenta um fluxograma do processo de aquisição, identificação e sintonia proposto.

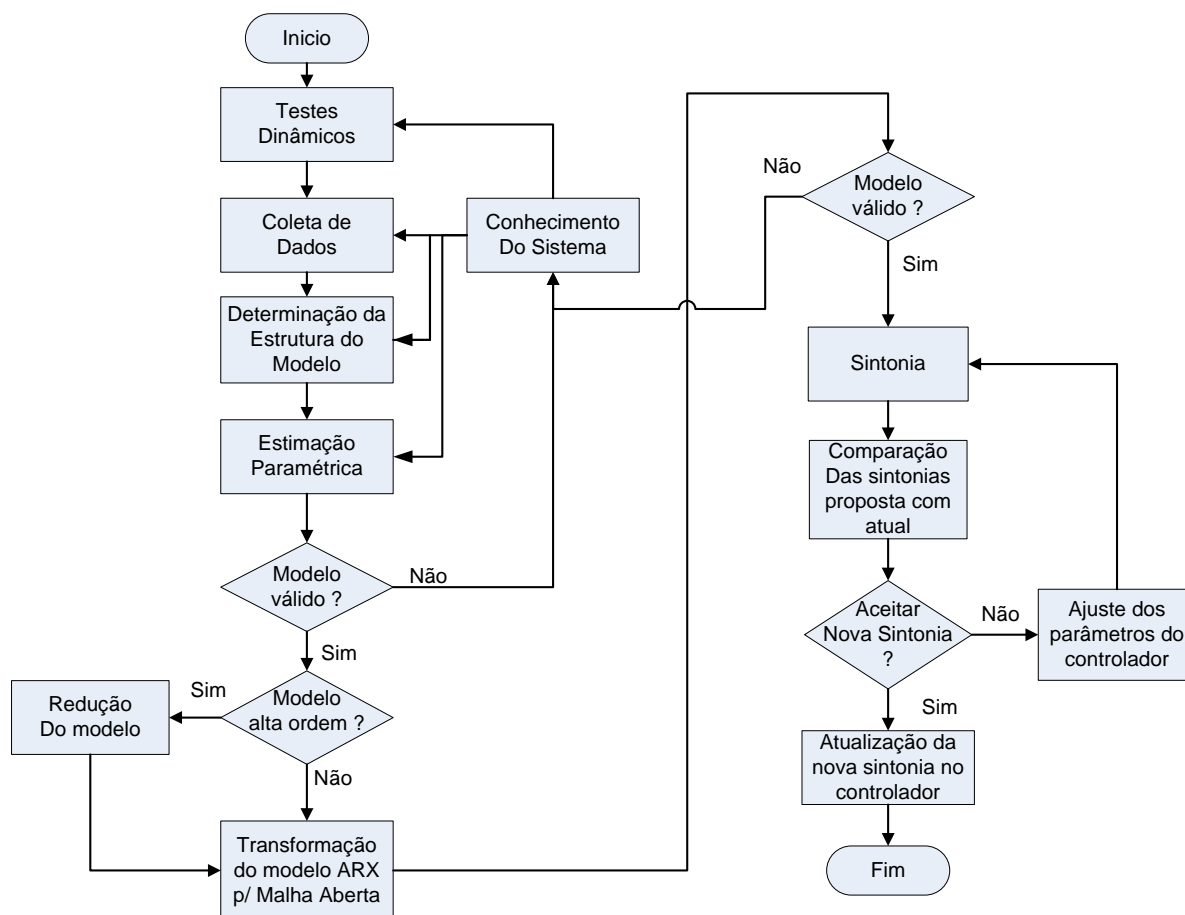


Figura 21 – Fluxograma do Tele-Sintonizador

5. Descrição dos testes e resultados

Nesta seção são apresentados todos os testes realizados utilizando simulações de sistemas com diferentes características. O propósito principal é a validação da arquitetura proposta e dos cálculos utilizados para identificação e sintonia de modelos de sistemas industriais de primeira ordem com tempo morto.

Para validação da arquitetura proposta, foram feitos testes de identificação e sintonia em diferentes cenários, de forma *online* e *offline* utilizando o MATLAB e o tele-sintonizador. Os testes foram feitos tanto em malha aberta quanto em malha fechada.

Nas próximas seções será detalhada a metodologia utilizada e os resultados dos testes realizados.

5.1. Metodologia dos testes

Os testes cobertos neste trabalho utilizam três fontes de dados: MATLAB/SIMULINK, simulador de ambiente *fieldbus* FBSIMU e equipamentos comerciais *Foundation Fieldbus*.

Inicialmente foram feitos testes com o MATLAB/ SIMULINK. O MATLAB foi utilizado tanto para simulações quanto para identificação e sintonia dos sistemas. Para isso, foram criadas rotinas no MATLAB usando os mesmos algoritmos desenvolvidos no tele-sintonizador. Desta forma, o algoritmo foi comparado com a correspondente função do MATLAB. Um exemplo desta comparação é o cálculo do algoritmo dos mínimos quadrados recursivo (MQR) comparado com a função ARX do MATLAB. Na figura 22 é mostrado o diagrama utilizado no MATLAB/ SIMULINK para gerar os dados do sistema simulado.

Uma segunda forma de simulação foi o FBSIMU. O FBSIMU é um sistema de simulação de redes desenvolvido na EESC-USP por Brandão (2005). Com ele é possível simular uma malha de controle *fieldbus* com função de transferência do sistema de até quinta ordem. Também é possível comunicar com o simulador de forma *online* através do OPC ou de forma *offline* através da gravação dos dados simulados em arquivo. Neste projeto foi utilizado o FBSIMU como principal simulador de dados para a validação do projeto.

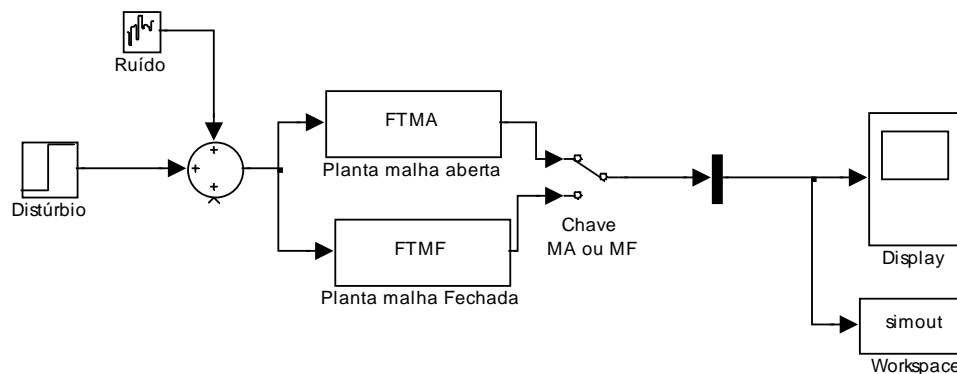


Figura 22 – Modelo do sistema FOPDT com PI simulado no MATLAB/SIMULINK

Para a utilização do OPC lendo as variáveis do FBSIMU foi necessária a implementação de um *driver* OPC que realizasse a leitura das variáveis do FBSIMU através da utilização de variáveis compartilhadas pelo LabVIEW (*Shared Variables*). Este *driver* foi desenvolvido dentro do grupo do Laboratório de Automação Industrial por Torrisi (2007c). Desta forma, foi possível a comunicação do cliente OPC com o FBSIMU como mostrado na figura 23.

Para os testes realizados foram sempre utilizados uma configuração padrão com taxa de controle de um segundo e taxa de atualização *CyberOPC* também de um segundo que são valores comuns em ambiente industrial para comunicações utilizando tecnologia OPC. A configuração tanto do FBSIMU quanto do ambiente de comunicação remota são mostradas no Apêndice I.

Por fim, foram realizadas simulações com o controlador *fieldbus* comercial DF51 da empresa SMAR Equipamentos Industriais, o que possibilitou a validação do tele-sintonizador em sistemas *Foundation Fieldbus* com equipamentos comerciais. Na figura 24 é apresentada a arquitetura do ambiente de teste com o DF51.

A validação dos testes no ambiente com o DF51 mostrou que o tele-sintonizador pode se comunicar com diferentes equipamentos nos padrões OPC e *CyberOPC*. Porém, os resultados de validação apresentados neste trabalho somente se restringem à ferramenta FBSIMU. No Apêndice II são mostrados os detalhes da configuração do ambiente utilizando DF51.

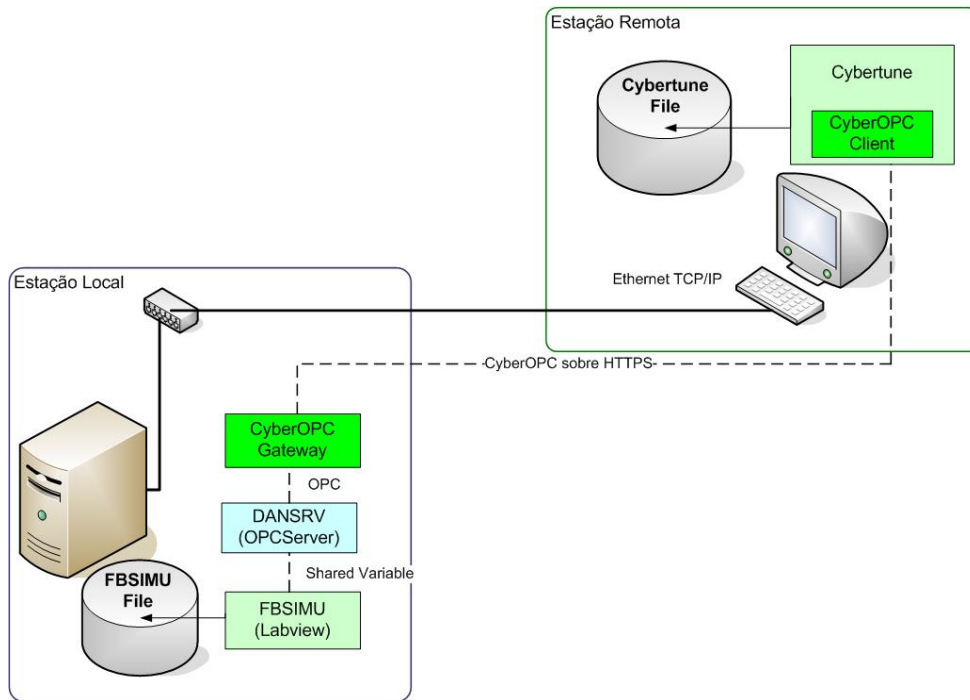


Figura 23 – Arquitetura do sistema de teste usando FBSIMU e o tele-sintonizador remoto

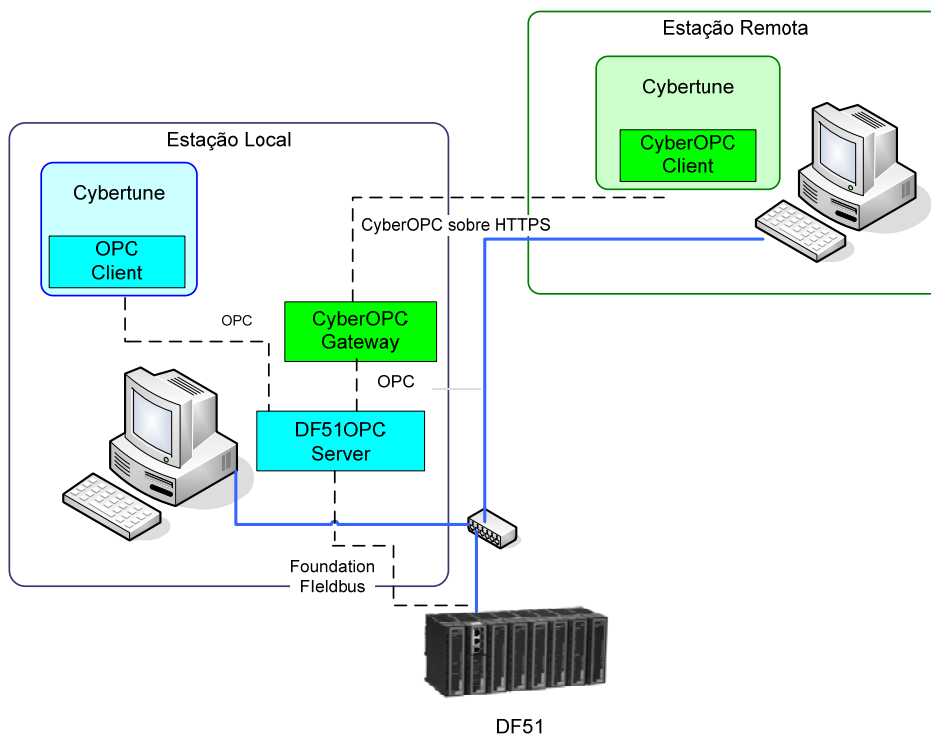


Figura 24 - Arquitetura do sistema de teste usando DF51 e o tele-sintonizador

Quanto à forma de comparação e validação entre os diferentes resultados da identificação e sintonia para os diferentes sistemas foram estudados os índices AIC, FPE e FIT, mostrados na seção 3.1. Porém, optou-se por utilizar somente o índice FIT como

critério de validação por ser um índice que atende as expectativas de comparação desejada e também por exigir um menor esforço computacional para implementação no tele-sintonizador.

Na figura a seguir é ilustrado um gráfico de comparação entre os três índices. Os dados utilizados no gráfico são resultados dos seis sistemas testados em malha fechada descritos na tabela 10 e que serão detalhados dentro deste capítulo.

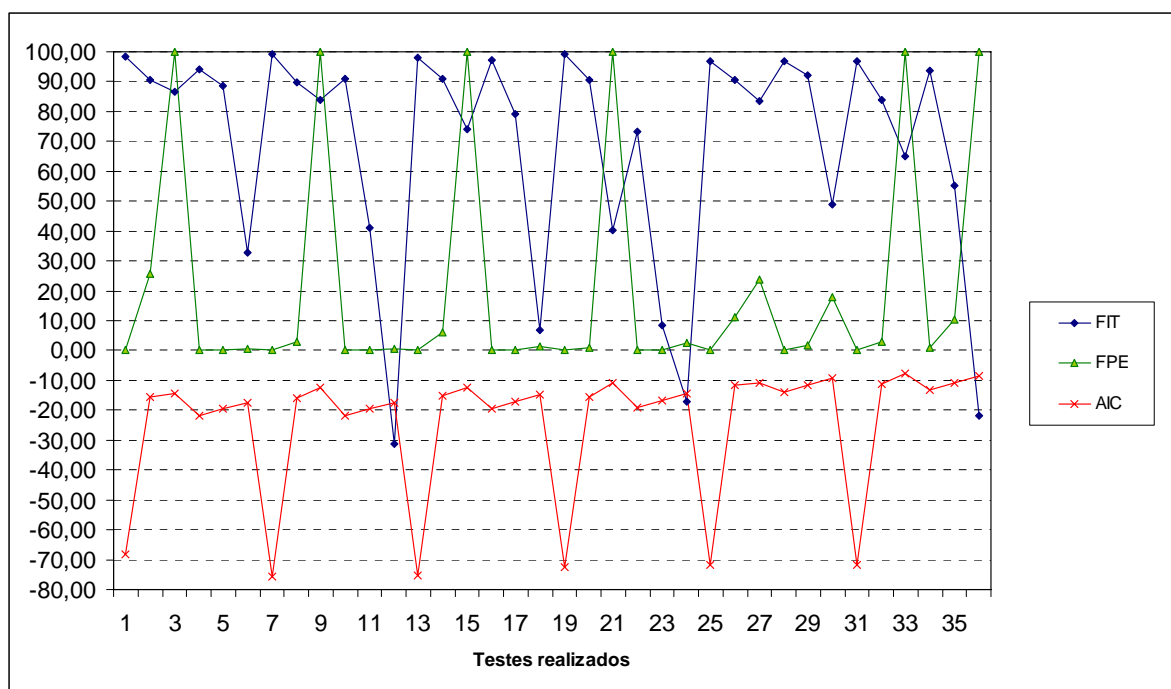


Figura 25 - Gráfico comparativo entre os índices de desempenho FIT, FPE e AIC

Na figura 25 os índices FIT e o AIC são mostrados com os valores reais obtidos, enquanto que o FPE foi normalizado de 0 a 100 somente por questão de equalização de escala com os outros dois índices. O critério de avaliação do FIT é pelo maior, ou seja, os maiores valores são os melhores resultados, enquanto que para os índices FPE e AIC os menores valores são os melhores resultados.

A análise do gráfico mostra que uma boa identificação acontece quando o valor do FIT é próximo de 100, enquanto que o FPE é próximo de zero e o AIC é próximo de -100. Isto pode ser observado para os testes 1, 7, 13, 19, 21, 25 e 31. Para uma identificação considerada ruim, então se obtém valores baixos de FIT, e conseqüentemente valores altos de AIC e FPE. Fato observado nos testes 6, 12, 18, 24 e 36. Isto mostra que os três índices poderiam ser utilizados na comparação dos resultados.

5.1.1. Descrição dos sistemas

Nesta seção são descritos os sistemas escolhidos para validação do projeto proposto. Foram escolhidos sistemas com diferentes características com a relação θ/τ variando de 0.1 a 2.3. A relação θ/τ é utilizada como forma de comparação entre os diferentes tipos de sistema FOPDT como verificados nos trabalhos de Zhuang e Atherton (1993) e Liptak (1995).

Na tabela 8 são mostradas todas as equações dos sistemas utilizadas para a validação do projeto. No total são seis diferentes sistemas, sendo que para cada um são apresentadas as equações em malha aberta e malha fechada no domínio contínuo.

	Sistema	θ/τ	Função de transferência malha aberta	Função de transferência malha fechada
1	$G_p = \frac{3}{150s+1} e^{-20s}$	0.13	$\frac{-30s+3}{1500s^2+160s+1}$	$\frac{-1200s^2+120s}{60000s^3+3700s^2+242.5s+6.75}$ $K_c = 2.25; T_i = 40$
2	$G_p = \frac{2}{100s+1} e^{-50s}$	0.50	$\frac{-50s+2}{2500s^2+125s+1}$	$\frac{-5000s^2+200s}{250000s^3+8000s^2+235s+1.8}$ $K_c = 0.9; T_i = 100$
3	$G_p = \frac{1.5}{22s+1} e^{-25s}$	1.14	$\frac{-18.75s+1.5}{275s^2+34.5s+1}$	$\frac{-937.5s^2+75s}{13750s^3+1230s^2+79.7s+0.792}$ $K_c = 0.53; T_i = 50$
4	$G_p = \frac{4.5}{49s+1} e^{-75s}$	1.53	$\frac{-168.8s+4.5}{1838s^2+86.5s+1}$	$\frac{-2.53E4s^2+675s}{2.75E5s^3+9668s^2+216.1s+0.588}$ $K_c = 0.13; T_i = 150$
5	$G_p = \frac{9}{7.5s+1} e^{-1425s}$	1.90	$\frac{-64.13s+9}{53.44s^2+14.63s+1}$	$\frac{-1828s^2+256.5s}{1523s^3+320.6s^2+38.63s+0.474}$ $K_c = 0.05; T_i = 28.50$
6	$G_p = \frac{25}{15s+1} e^{-35s}$	2.33	$\frac{-437.5s+25}{262.5s^2+32.5s+1}$	$\frac{-30625s^2+1750s}{18375s^3+1803s^2+90.25s+0.3857}$ $K_c = 0.0154; T_i = 70$

Tabela 8 - Equações dos sistemas utilizados para validação do trabalho

Na tabela 8, a coluna “Função de transferência malha aberta” representa o sistema com o tempo morto segundo a aproximação de Paddé, conforme equação (29). A coluna “Função de transferência malha fechada” representa o sistema malha fechada com variação de carga como mostrado em (28). Os valores de K_c e T_i correspondem aos parâmetros de pré-sintonia do controlador PI conforme o método de sintonia de ZN (LIPTAK, 1995). A seguir, é mostrada a resposta de cada sistema a uma variação degrau em malha aberta (figura 26) e a uma variação de carga em malha fechada (figura 27).

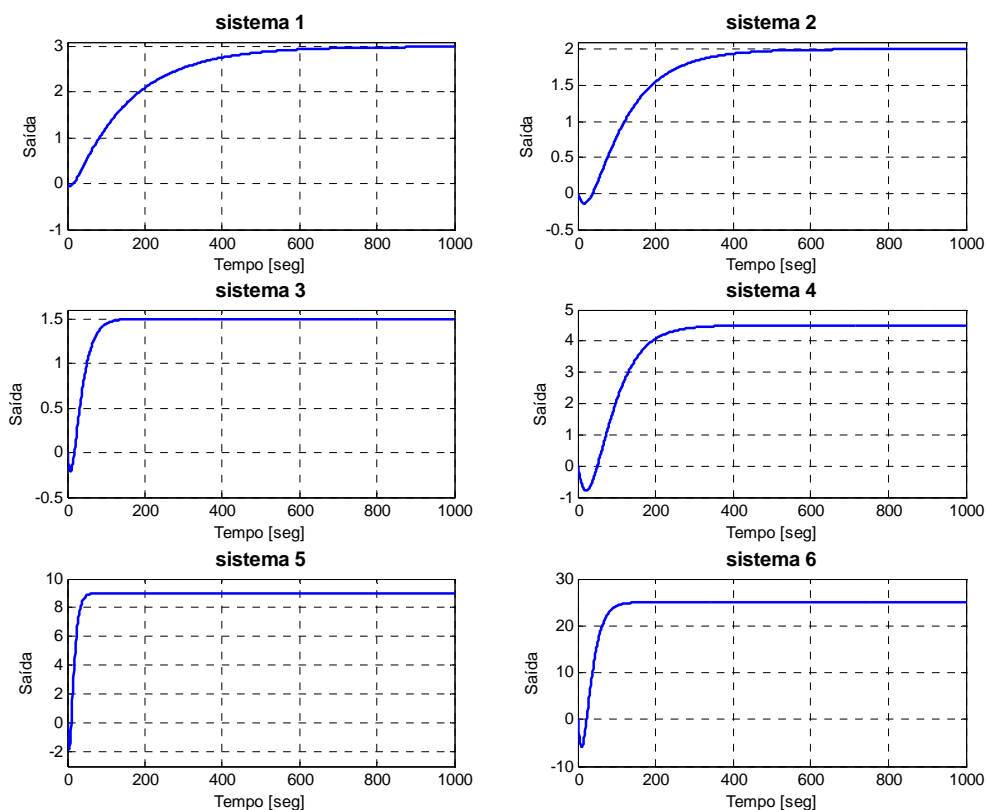


Figura 26 - Resposta dos sistemas malha aberta a uma entrada a degrau unitário

Os resultados mostrados nas próximas seções serão sempre referenciados aos sistemas pelo número correspondente da tabela 8.

5.2. Resultados dos testes

Nesta seção serão mostrados todos os testes realizados para validação deste projeto. Para facilitar o entendimento, os testes foram divididos na seqüência do capítulo 3, que é a seqüência lógica para a identificação e sintonia do sistema, que são:

- Identificação do modelo, onde é inicialmente obtido o modelo ARX representando o sistema real.
- Transformação do modelo, onde é realizada a transformação do modelo ARX da identificação e obtido um modelo FOPDT estimado.
- Sintonia do modelo estimado, onde utilizando o modelo FOPDT são propostas novas sintonias para o sistema. Também é feita uma simulação confrontando a sintonia atual e as propostas.

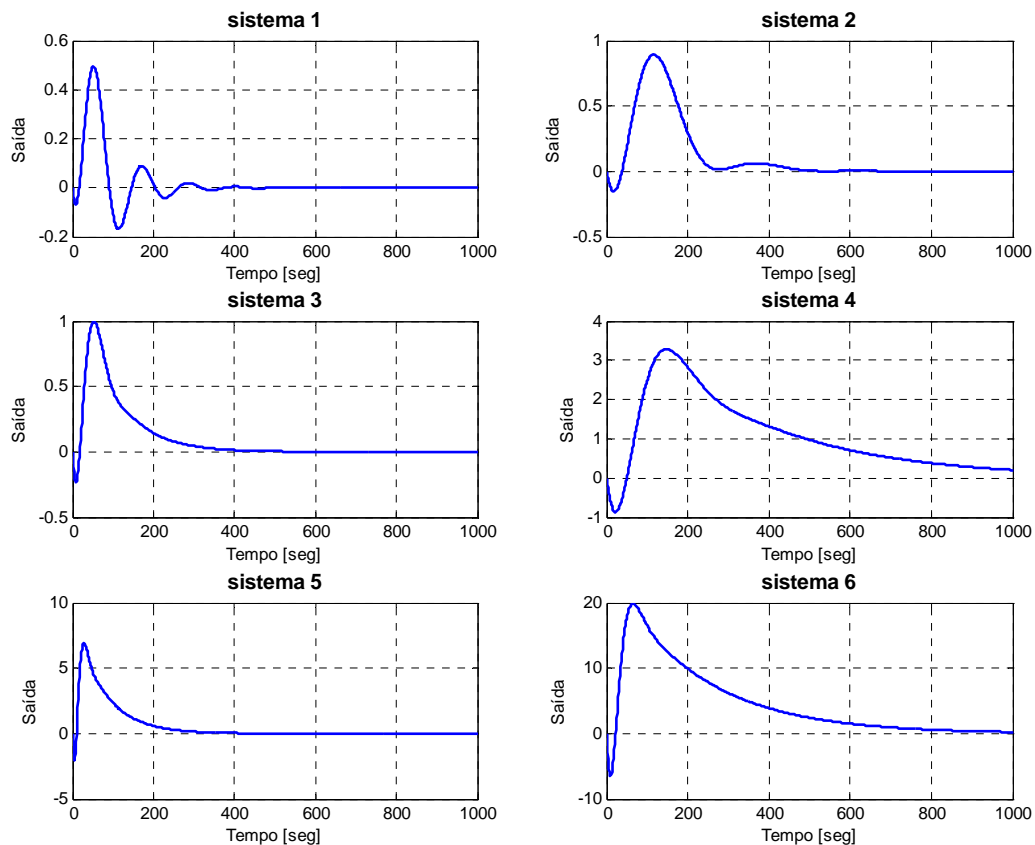


Figura 27 - Resposta dos sistemas malha fechada a variação de carga de degrau unitário

As seções que seguem descrevem cada uma destas etapas e os correspondentes testes de validação. Serão detalhados em cada etapa somente os testes mais relevantes de cada cenário englobando os testes em ambiente local, remoto, e com ruídos de sinal.

Para os testes em ambiente local e remoto foi utilizada a arquitetura mostrada na figura 22. O teste local foi realizado utilizando os dados salvos pelo próprio simulador FBSIMU e a identificação foi feita de forma offline pelo tele-sintonizador. Para o teste remoto utilizou-se o tele-sintonizador comunicando online com o FBSIMU sob tecnologia CyberOPC através de uma comunicação ponto a ponto utilizando a arquitetura mostrada na figura 23. As configurações utilizadas pelo ambiente do FBSIMU e do tele-sintonizador comunicando remotamente são mostradas no Apêndice II.

Os testes com ruído de sinal foram realizados através do MATLAB/SIMULINK utilizando a arquitetura da figura 22. As configurações utilizadas neste ambiente são mostradas no Apêndice I.

Ao final de cada etapa apresenta-se uma tabela completa com todos os resultados e ao final do capítulo será feita uma análise de todos os testes realizados.

5.2.1. Identificação do modelo

A primeira fase do tele-sintonizador é a identificação do modelo ARX através da obtenção dos dados coletados do sistema. A proposta deste trabalho é a obtenção do modelo ARX através do método dos mínimos quadrados recursivo mostrado na seção 3.1.

Nesta seção serão mostrados os testes somente até a fase de obtenção do modelo ARX. Para todos os testes de identificação sempre foi utilizada a mesma configuração do modelo, sendo ordem 4 para identificação em malha aberta e ordem 10 para identificação malha fechada.

Para a identificação, o tele-sintonizador foi restringido a sistemas em malha aberta e malha fechada até 4ª ordem. Esta limitação se deve à estrutura do ARX e ao módulo de cálculo de raízes, também limitado a sistemas de 4ª ordem. Os sistemas simulados em malha fechada com dados de rede necessitaram de modelos de 10ª ordem. Esta ordem não pode ser testada com o tele-sintonizador, porém houve a validação no MATLAB, conforme os resultados abaixo.

5.2.1.1. Teste local em malha aberta

Neste teste foi utilizado o sistema 2 em uma identificação malha aberta com o arquivo gerado pelo FBSIMU. A equação obtida do modelo ARX é mostrada abaixo ($T_0 = 1[\text{seg}]$):

$$\begin{aligned} \text{Zeros} &: [1.0434, (-0.4409 \pm 0.5075i)] \\ \text{Pólos} &: [0.9903, 0.9590, (-0.4675 \pm 0.5063i)] \\ \text{Ganho} &: [-0.0190] \end{aligned} \quad (76)$$

Na figura 28 é apresentada a resposta do modelo estimado e do sistema real. Onde Y_{real} representa o dado obtido do sistema, Y_{arx} é a identificação feita pela função ARX do Matlab e Y_{mqr} a identificação do tele-sintonizador.

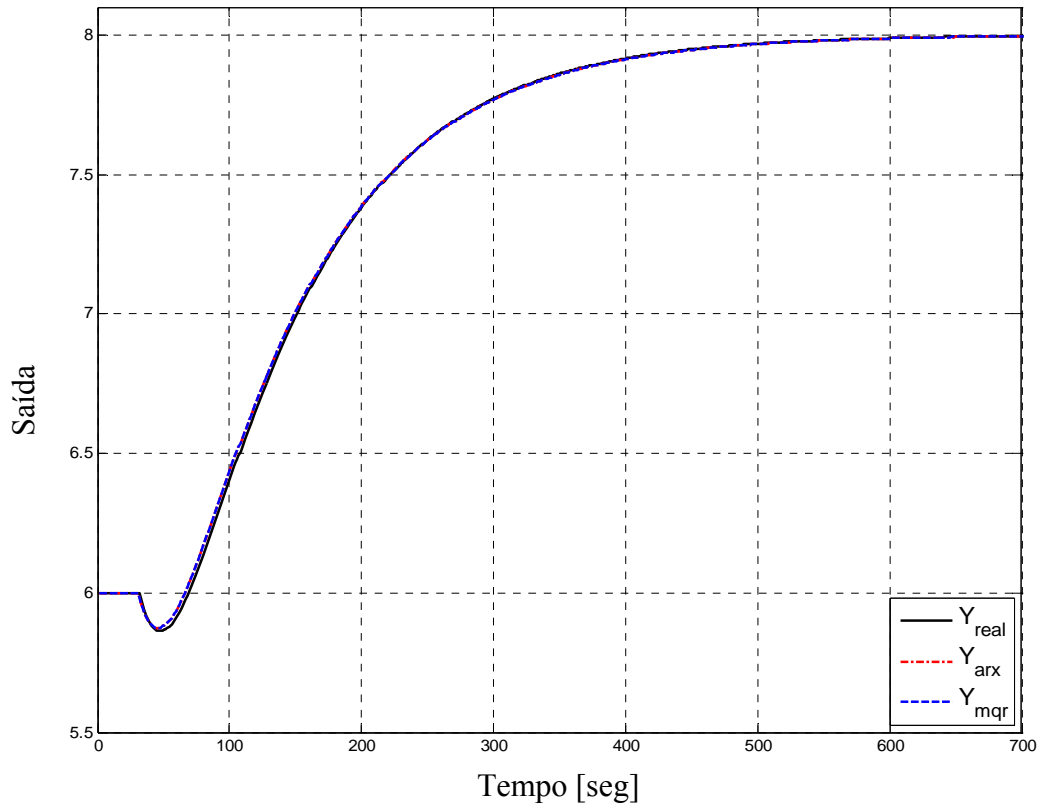


Figura 28 - Gráfico da identificação local do sistema 2 em malha aberta

O resultado da identificação obteve um FIT de 98.50%. Analisando o gráfico acima, pode-se concluir que o FIT mostrou uma identificação muito próxima do modelo.

5.2.1.2. Teste remoto em malha aberta

Neste outro teste foi utilizado o tele-sintonizador identificando o sistema 2 em malha aberta realizado remotamente utilizando o *CyberOPC* de forma *online*.

A equação obtida da estimação do modelo ARX é mostrada abaixo ($T_0 = 5[seg]$):

$$\begin{aligned}
 \text{Zeros} &: [1.1587, (-0.7101 \pm 0.6817i)] \\
 \text{Pólos} &: [0.9644, 0.8777, (-0.4889 \pm 0.4141i)] \\
 \text{Ganho} &: -0.0386
 \end{aligned} \tag{77}$$

A figura 37, apresentada na seção 5.2.2.2., mostra a resposta do modelo estimado e do sistema real. O resultado da identificação foi um FIT de 95.59%. Este resultado mostrou uma identificação também muito próxima dos dados obtidos do modelo.

5.2.1.3. Teste com ruído em malha aberta

Neste teste foi utilizado o MATLAB/SIMULINK com o sistema 2 na configuração mostrada na figura 21 com a aplicação de ruído no sinal de entrada. Foram aplicados três sinais de ruído diferentes correspondentes ao *Power Noise* no MATLAB/SIMULINK (PN): 5E-3, 5E-2, 5E-1. Foram feitos testes com e sem filtro de primeira ordem.

No teste com ruído de PN=0.05 foi obtida a equação do modelo mostrada abaixo ($T_0 = 0.5[\text{seg}]$):

$$\begin{aligned} \text{Zeros} &: [1.1068, (-0.1669 \pm 0.8573i)] \\ \text{Pólos} &: [0.9975, 0.6369, (-0.2656 \pm 0.3794i)] \\ \text{Ganho} &: -0.0150 \end{aligned} \quad (78)$$

A figura 29 apresenta a resposta do modelo estimado e do sistema real. Onde Y_{real} representa o dado obtido do sistema, Y_{arx} é a identificação feita pela função de do MATLAB e Y_{mqr} a identificação do algoritmo dos mínimos quadrados utilizado neste projeto.

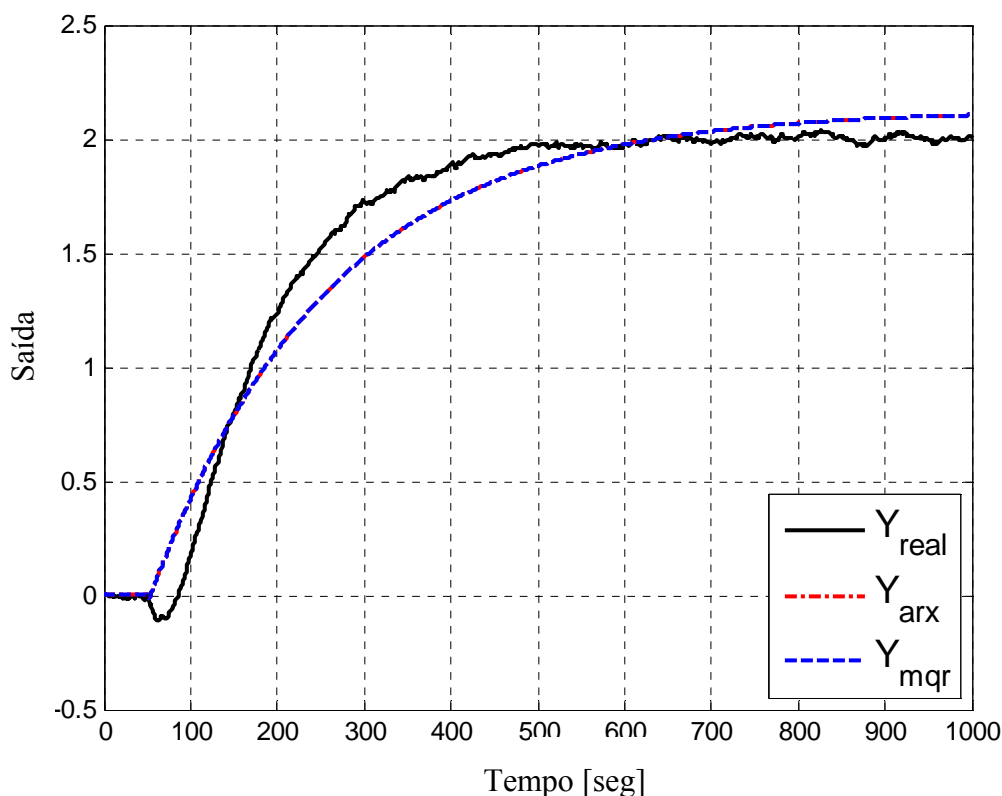


Figura 29 - Gráfico das identificações do sistema 2 com ruído de 0.05 em malha aberta

O resultado da identificação produziu um FIT de 81.52% e mostrou uma degradação da identificação devido à presença do ruído.

Em um segundo teste, foi aplicado um filtro de primeira ordem ($T_{Filter} = 4.5[seg]$). A figura a seguir mostra o gráfico de identificação.

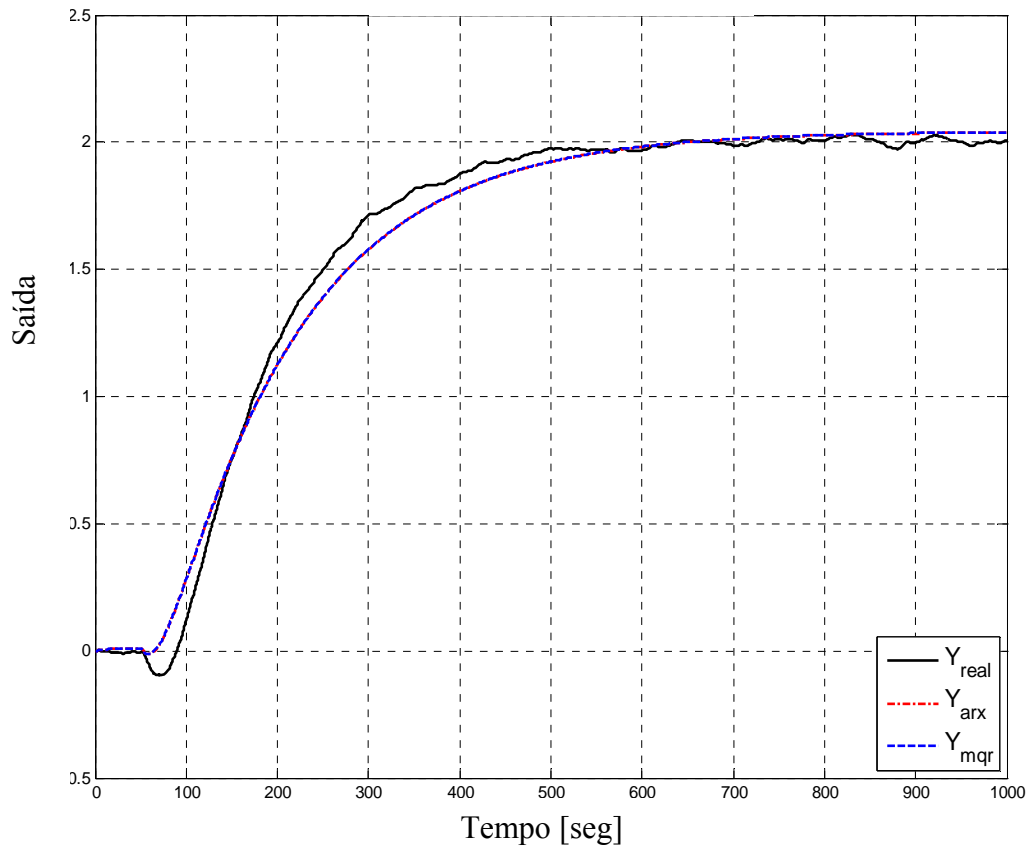


Figura 30 - Gráfico das identificações do sistema 2 com ruído 0.05 em malha aberta e utilizando filtro

A aplicação do filtro mostrou melhoria na identificação com FIT = 90.17%. Portanto, neste caso, obteve-se um bom resultado com a aplicação do filtro de primeira ordem.

Para um valor maior de ruído com $PN=0.5$ há uma degradação ainda maior no sinal com FIT de 85.63%. Neste caso, apesar da aplicação do filtro, não mostrou uma grande melhora no sinal original e o sistema obtido malha aberta apresentou problemas na posterior sintonia do modelo. O valor do filtro que trouxe um melhor resultado foi com um $T_{Filter} = 6.9[seg]$ resultando em um FIT de 86.69%.

5.2.1.4. Teste local em malha fechada

Neste teste foi utilizado o sistema 2 em malha fechada utilizando o arquivo gerado pelo FBSIMU. A equação obtida do modelo ARX de 10ª ordem é mostrada a seguir ($T_0 = 1[seg]$):

$$\begin{aligned}
 \text{Zeros} &: \left[1.0202, 1.0000, (0.2922 \pm 0.8135i), (-0.2575 \pm 0.8004i), -0.8257, \right. \\
 &\quad \left. (-0.6718 \pm 0.4862i) \right] \\
 \text{Pólos} &: \left[0.9950, (0.9944 \pm 0.0122i), (-0.2922 \pm 0.8135i), (-0.2575 \pm 0.8004i), \right. \\
 &\quad \left. (-0.6718 \pm 0.4862i), -0.8257 \right] \quad (79) \\
 \text{Ganho} &: -0.0098
 \end{aligned}$$

Na figura 31 é mostrado o gráfico do sinal medido (Y_{real}) e do ARX estimado pela função do MATLAB (Y_{arx}) e da função dos mínimos quadrados (Y_{mqr}).

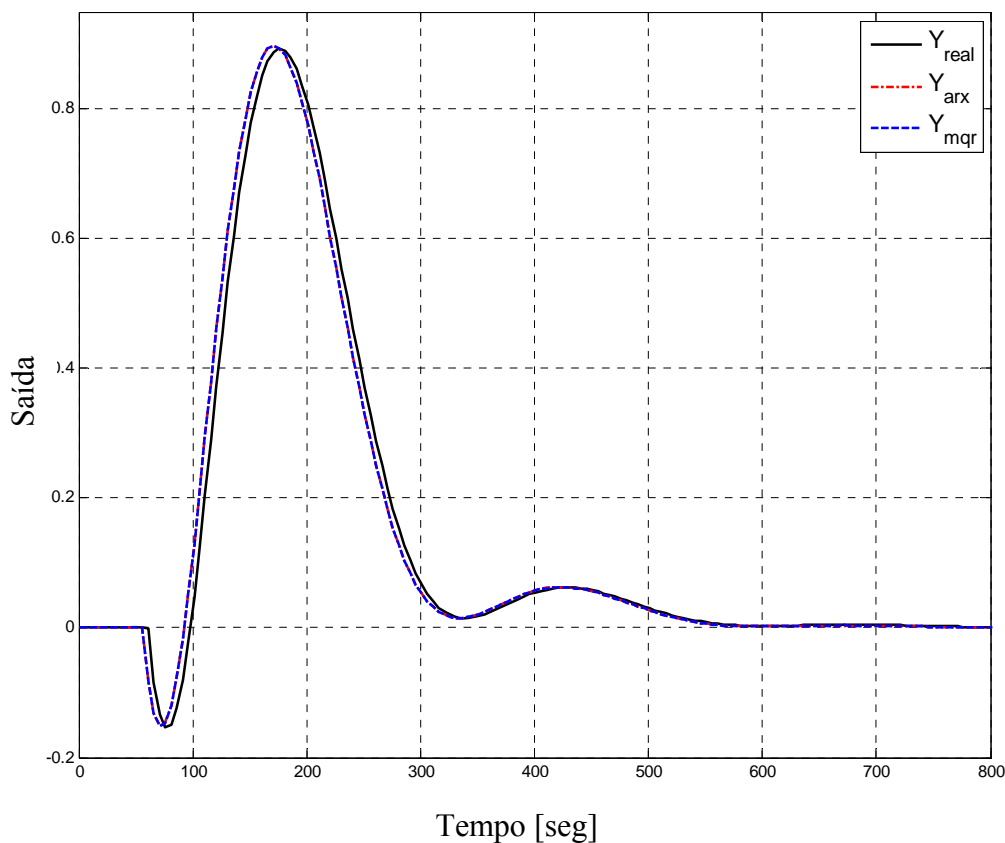


Figura 31 - Gráfico da identificação local do sistema 2 em malha fechada

A identificação em malha fechada obteve FIT de 89.96%. As análises do resultado e do gráfico da figura acima mostraram uma identificação muito próxima do sinal original.

5.2.1.5. Teste remoto em malha fechada

Neste outro teste foi utilizado o MATLAB com identificação do sistema 2 utilizando o arquivo gerado remotamente pelo tele-sintonizador comunicando com o FBSIMU. A identificação foi estimada considerando aproximação ARX de 10^a ordem e taxa de amostragem ($T_0 = 5[seg]$). A equação abaixo mostra o resultado do modelo ARX.

$$\begin{aligned}
 \text{Zeros: } & \left[(0.7585 \pm 1.5687i), -1.5792, (-1.1065 \pm 0.6397i), (-0.0045 \pm 1.1032i), \right. \\
 & \left. 1.0000, 0.2842 \right] \\
 \text{Pólos: } & \left[0.9510, (0.9436 \pm 0.1076i), (0.3007 \pm 0.8393i), (-0.3252 \pm 0.8058i), \right. \\
 & \left. (-0.7973 \pm 0.3343i), -0.4923 \right] \\
 \text{Ganho: } & 0.0086
 \end{aligned} \tag{80}$$

Na figura abaixo é mostrado o gráfico do sinal medido (Y_{real}) e do ARX estimado para o modelo de 10^a ordem (Y_{mqr}).

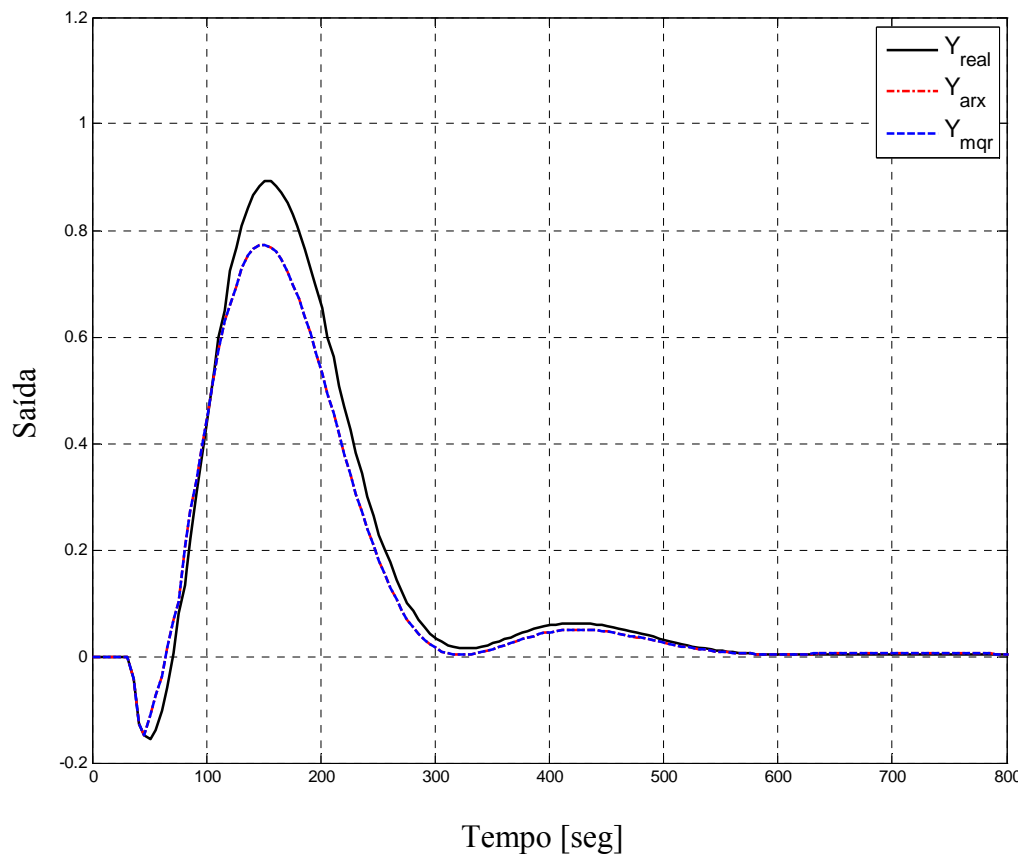


Figura 32 - Gráfico de identificação remota do sistema 2 em malha fechada

A identificação remota em malha fechada apresentou FIT de 83.89%. A identificação obtida está próxima do sinal original, porém inferior aos testes locais.

5.2.1.6. Teste com ruído em malha fechada

Neste teste foi utilizado o MATLAB/SIMULINK utilizando o sistema 2 com a aplicação de ruído no sinal de entrada. Foram utilizados três diferentes sinais de ruído variando o PN de 1E-6, 1E-5, 1E-4. Foram feitos testes com e sem filtro de primeira ordem. Por fim, também foram feitos testes com outros algoritmos paramétricos.

Para um ruído de $1E-5$ foi obtida a equação do modelo mostrada abaixo ($T_0 = 0.5[\text{seg}]$):

$$\begin{aligned} \text{Zeros: } & \left[1.0129, 0.9988, (0.3887 \pm 0.7402i), (-0.1819 \pm 0.7589i), \right. \\ & \left. -0.7866, (-0.5881 \pm 0.5146i) \right] \\ \text{Pólos: } & \left[(0.9946 \pm 0.0100i), 0.9858, (0.3914 \pm 0.7378i), (-0.1744 \pm 0.7666i), \right. \\ & \left. -0.8058, (-0.5949 \pm 0.5183i) \right] \\ \text{Ganho: } & -0.0099 \end{aligned} \quad (81)$$

A figura 33 apresenta a resposta do modelo estimado e do sistema real. Sendo Y_{real} representa o dado obtido do sistema, Y_{arx} é a identificação feita pela função de do MATLAB e Y_{mqr} a identificação do algoritmo dos mínimos quadrados utilizado neste projeto.

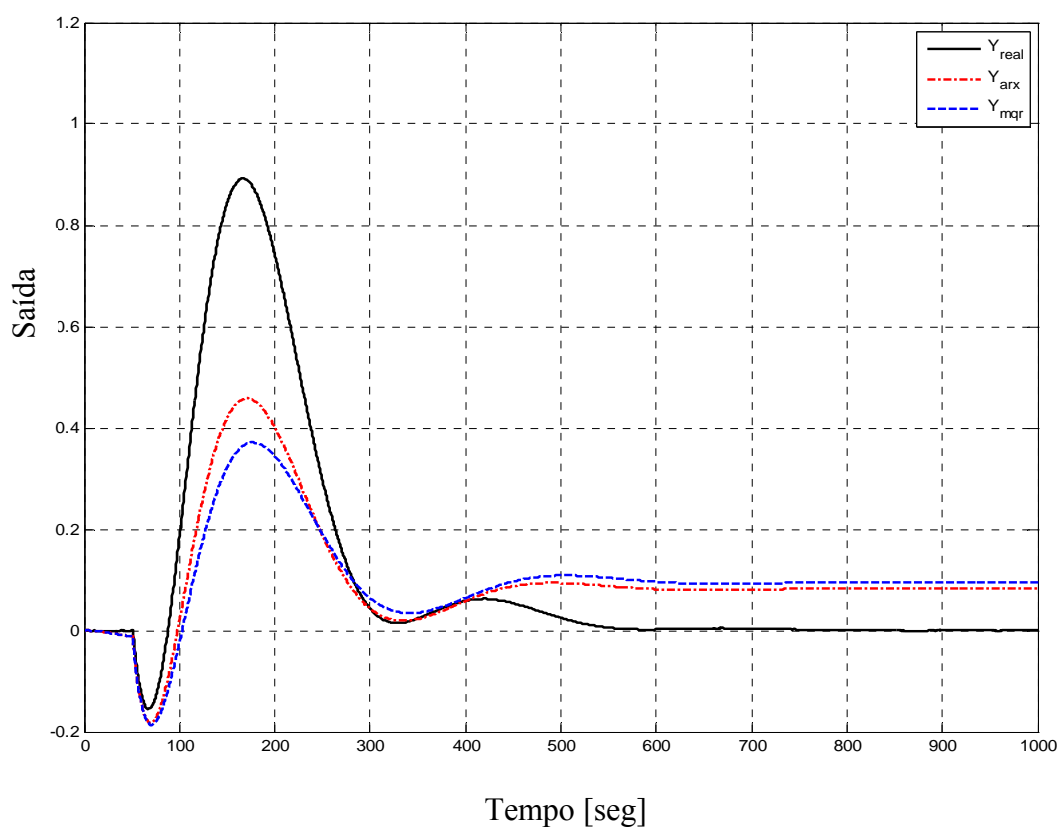


Figura 33 - Gráfico das identificações do sistema 2 com ruído em malha fechada

O resultado da identificação apresentou um FIT de 41.24% e mostrou uma degradação muito grande da identificação devido à presença do ruído. A utilização do filtro não fez com que a identificação melhorasse.

5.2.1.7. Resumo dos testes de identificação

Um resumo de todos os testes realizados de identificação de sistemas em malha aberta e malha fechada será apresentado nas tabelas a seguir. Os dados são compostos pelos testes descritos em detalhes acima e também por outros testes não detalhados aqui.

A tabela é constituída pelos seis sistemas estudados neste trabalho e possui somente os pontos principais da arquitetura. Para cada sistema foram realizados testes nas diferentes categorias criadas: categoria 1 é o sistema ideal sem inclusão de nenhuma incerteza aparente. A categoria 2 representa os testes com o simulador *fieldbus* FBSIMU em ambiente local (mesma estação de Simulação e Identificação). A categoria 3 engloba os testes com comunicação remota utilizando tecnologia *CyberOPC* proposto neste trabalho. As categorias 4, 5 e 6 são correspondentes a testes que possuem incertezas devido a ruídos no processo de medição (simulado como um ruído branco).

Primeiramente é feita a análise dos sistemas em malha aberta. Na tabela abaixo são mostrados todos os testes de identificação ordenados pelo sistema.

Teste	Sistema	Categorias	T _o [seg]	FIT[%] Sem filtro	Filtro [seg]	FIT[%] com filtro
1	1	1	0.5	99.71		
2	1	2	1	95		
3	1	3	5	97.85		
4	1	4	0.5	94.73	0.9	97.58
5	1	5	0.5	93.07	2.9	95.31
6	1	6	0.5	93.20	2.1	93.39
7	2	1	0.5	99.66		
8	2	2	1	98.50		
9	2	3	5	95.59		
10	2	4	0.5	85.30	1.5	95.51
11	2	5	0.5	81.52	4.5	90.19
12	2	6	0.5	85.63	6.9	86.70
13	3	1	0.5	99.14		
14	3	2	1	95.24		
15	3	3	1	95.26		
16	3	4	0.5	84.71	1.5	93.89
17	3	5	0.5	81.70	4.1	87.03
18	3	6	0.5	69.53	7.1	78.07
19	4	1	0.5	99.54		
20	4	2	1	93.77		
21	4	3	2	93.44		
22	4	4	0.5	75.55	1.5	90.49
23	4	5	0.5	71.07	6.5	85.84

Teste	Sistema	Categorias	T _o [seg]	FIT[%] Sem filtro	Filtro [seg]	FIT[%] com filtro
24	4	6	0.5	78.18	9.1	79.90
25	5	1	0.5	98.38		
26	5	2	1	93		
27	5	3	5	96.41		
28	5	4	0.5	87.52	1.5	92.57
29	5	5	0.5	77.86	3.9	83.57
30	5	6	0.5	38.82	8.9	63.51
31	6	1	0.5	99		
32	6	2	1	93.77		
33	6	3	5	91.6		
34	6	4	0.5	80.10	1.9	92.21
35	6	5	0.5	77.96	7.1	85.44
36	6	6	0.5	59.11	10.5	75.76

Tabela 9 - Resultados da identificação em malha aberta

(1) Na tabela 9 a coluna “Categorias” refere-se:

- 1 – Teste de identificação realizado no MATLAB/SIMULINK sem ruído de forma *offline*.
- 2 – Teste de identificação realizado no tele-sintonizador com arquivo FBSIMU sem ruído de forma *offline*.
- 3 – Teste de identificação realizado no tele-sintonizador usando *CyberOPC* com FBSIMU de forma *online*.
- 4 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de 5E-3 de forma *offline*.
- 5 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de 5E-2 de forma *offline*.
- 6 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de 5E-1 de forma *offline*.

Baseado nos resultados da tabela 9 foi elaborado um gráfico comparativo entre os diferentes sistemas nas diferentes categorias com o índice FIT mostrado na figura 34.

A análise da figura 34 mostra que para todas as identificações utilizando o MATLAB/SIMULINK sem ruído (categoria 1) resultam em uma identificação próxima a 100%. Para os sistemas locais com o FBSIMU, obtêm-se índices maiores que 90%. Na categoria 3, onde foram feitos testes com o *CyberOPC* em ambiente de rede ponto a ponto, mostrou valores próximos aos testes locais. A inclusão de ruído no sinal de medição tem uma maior influência na identificação, como era esperado onde dependendo das características do sistema ele sentirá mais ou menos o efeito do ruído. A utilização do filtro sempre mostrou melhoras na identificação.

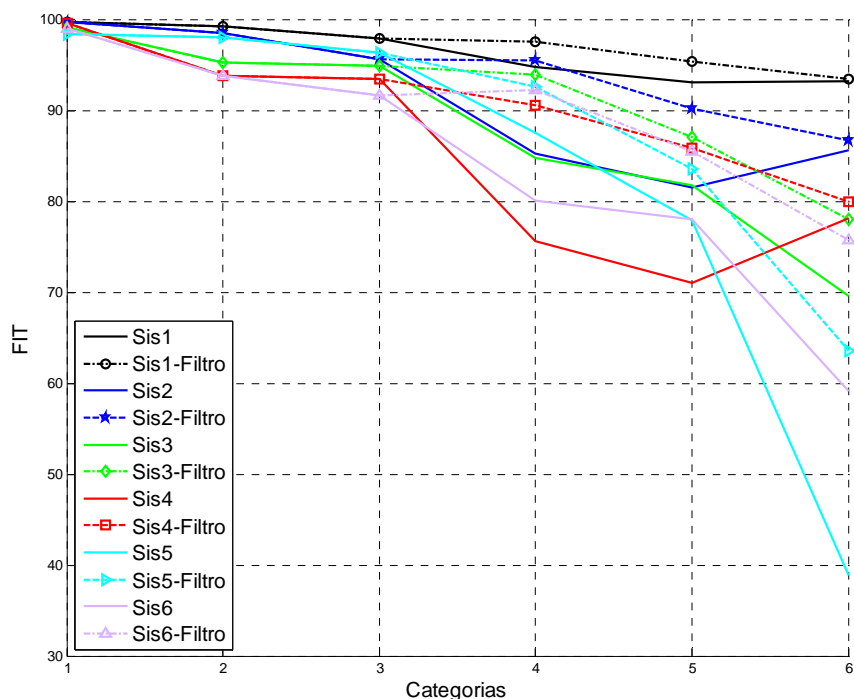


Figura 34 – Gráfico comparativo entre as identificações dos diferentes sistemas em malha aberta

Para os testes dos sistemas em malha fechada, a tabela 10 mostra o resumo de todos os testes realizados na identificação ordenados pelo sistema (coluna “Sis”) e categoria (coluna “Cat”).

Tst	Sis	Cat	FIT [%]	ITAE	FPE	AIC	Filtro [seg]	FIT [%]	ITAE	FPE	AIC
1	1	1	98.58	8.51E2	2.14E-30	-68.32					
2	1	2	90.45	7.90E2	1.49E-7	-15.72					
3	1	3	86.78	2.73E1	5.78E-7	-14.37					
4	1	4	94.15	3.60E3	2.90E-10	-21.96	0.5	91.85	3.84E3	2.11E-10	-22.28
5	1	5	88.74	7.11E3	2.90E-9	-19.66	0.1	83.48	9.28E3	2.05E-09	-20.00
6	1	6	32.66	3.86E4	2.86E-8	-17.37	0.5	33.68	3.76E4	1.98E-08	-17.74
7	2	1	99.03	7.63E2	1.61E-33	-75.51					
8	2	2	89.96	9.44E1	1.24E-7	-15.90					
9	2	3	83.89	2.53E2	4.19E-6	-12.38					
10	2	4	91.10	2.45E4	2.94E-10	-21.95	0.5	86.10	4.23E4	7.71E-11	-23.29
11	2	5	41.24	1.63E5	2.93E-9	-19.65	1	23.68	2.07E5	3.41E-10	-21.80
12	2	6	-31.29	3.24E5	2.82E-8	-17.38	0.5	-36.89	3.39E5	7.13E-09	-18.76
13	3	1	98.18	5.65E2	1.72E-33	-75.44					
14	3	2	91.07	2.20E2	2.69E-7	-15.13					
15	3	3	74.16	7.28E2	4.30E-6	-12.36					
16	3	4	97.07	4.11E3	3.32E-9	-19.52	0.1	95.81	4.73E3	2.42E-09	-19.84
17	3	5	78.98	4.99E4	3.32E-8	-17.22	0.05	78.22	4.46E4	2.76E-08	-17.40
18	3	6	6.89	1.98E5	3.26E-7	-14.94	0.05	7.46	1.96E5	2.69E-07	-15.13
19	4	1	99.35	4.30E3	2.68E-32	-72.70					
20	4	2	90.71	1.50E3	1.77E-7	-15.55					
21	4	3	40.31	9.74E3	2.12E-5	-10.76					
22	4	4	73.09	3.95E5	6.18E-9	-18.90	0.001	73.02	3.93E5	6.15E-09	-18.91
23	4	5	8.58	1.32E6	6.13E-8	-16.61	0.001	8.53	1.32E6	6.11E-08	-16.61

Tst	Sis	Cat	FIT [%]	ITAE	FPE	AIC	Filtro [seg]	FIT [%]	ITAE	FPE	AIC
24	4	6	-17.06	1.54E6	5.79E-7	-14.36	0.05	-17.06	1.56E6	5.76E-07	-14.37
25	5	1	96.97	3.35E3	6.64E-32	-71.79					
26	5	2	90.70	1.37E4	1.08E-5	-11.43					
27	5	3	83.50	5.10E2	2.29E-5	-10.68					
28	5	4	96.85	7.41E3	9.74E-7	-13.84	0.001	96.84	7.20E3	9.71E-07	-13.85
29	5	5	91.98	9.32E4	9.74E-6	-11.54	0.005	92.01	8.95E4	9.55E-06	-11.56
30	5	6	48.92	6.72E5	9.62E-5	-9.25	0.1	50.03	6.29E5	6.66E-05	-9.62
31	6	1	96.97	3.35E3	6.64E-32	-71.79					
32	6	2	83.75	4.16E5	1.49E-5	-11.11					
33	6	3	64.85	4.19E4	5.42E-4	-7.52					
34	6	4	93.87	3.82E5	1.98E-6	-13.13	1.00E-05	93.87	3.82E5	1.98E-06	-13.13
35	6	5	55.09	3.36E6	1.97E-5	-10.83	0.01	55.01	3.32E6	1.90E-05	-10.87
36	6	6	-21.91	8.20E6	1.92E-4	-8.56	0.001	-21.90	8.20E6	1.91E-04	-8.56

Tabela 10 - Resultados da identificação em malha fechada

(1) Na tabela 10 a coluna “Cat” refere-se às categorias:

- 1 – Teste de identificação realizado no MATLAB/SIMULINK sem ruído de forma *offline*.
- 2 – Teste de identificação realizado no MATLAB com arquivo FBSIMU de forma *offline*.
- 3 – Teste de identificação realizado no MATLAB usando *CyberOPC* com FBSIMU de forma *offline*.
- 4 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de 1E-6 de forma *offline*.
- 5 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de 1E-5 de forma *offline*.
- 6 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de 1E-4 de forma *offline*.

A figura 35 representa um gráfico comparativo do FIT entre e os diferentes sistemas nas diferentes categorias baseado na tabela 10.

A análise da figura 35 mostra que todas as identificações utilizando o MATLAB/SIMULINK (categoria 1) resultam em uma identificação com média acima de 98%. Para os sistemas com identificação local (categoria 2), há uma pequena degradação em comparação com a categoria 1 obtendo valores em média da ordem de 89%. A identificação remota utilizando tecnologia *CyberOPC* (categoria 3) mostrou resultados de identificação bons com pequena degradação do índice de desempenho comparado com os testes das categorias 1 e 2. Houve uma degradação maior para os sistemas com θ/τ na ordem de 2. Os sistemas também se comportam de maneira diferente com a inclusão de ruído no sinal, sendo que os sistemas 1 e 5 apresentaram maior influência do ruído quando comparados aos sistemas 2, 4 e 6. O filtro de primeira ordem não proporcionou uma melhora na identificação em todos os casos.

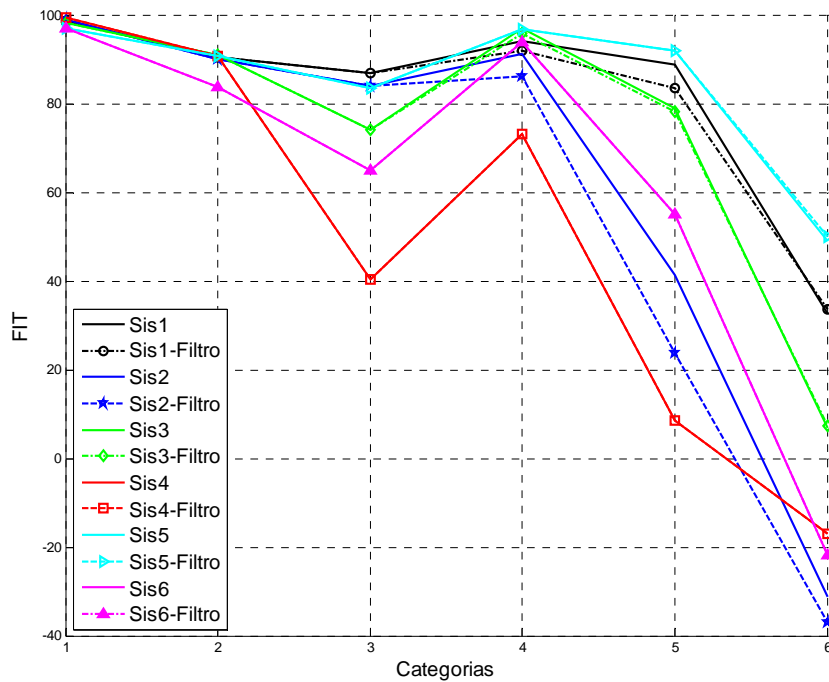


Figura 35 – Gráfico comparativo das identificações dos diferentes sistemas em malha fechada

Os problemas da categoria 3 considerados como incerteza de rede remota e das categorias com ruído de sinal pode ser explicado pelo fato do modelo ARX não trabalhar muito bem com incertezas. Para a obtenção de melhores resultados neste caso seria necessário considerar a incerteza no modelo o que leva a necessidade de outro modelo de identificação.

5.2.2. Transformação do modelo

Após a identificação do modelo ARX, é feita a redução do modelo utilizando o método mostrado na seção 3.2. A redução do modelo se aplica somente nos casos onde foi obtido um modelo maior do que esperado para o algoritmo de transformação do modelo. Para a identificação de sistemas em malha aberta a transformação do modelo precisa de um modelo ARX de 2ª ordem. Para a identificação em malha fechada é necessário um modelo de 3ª ordem.

Para todos os testes realizados, sempre se trabalhou com ordem superior ao esperado pelo algoritmo. Então, para todos os testes, sempre houve a necessidade de redução do modelo para ordens menores. Porém, os resultados de redução apresentaram-se, na maioria das vezes, satisfatórios, muito próximos ao modelo original. Alguns resultados ruins foram

obtidos para sistemas com ruído principalmente em malha fechada. Sendo assim, os resultados de redução não serão mostrados em detalhes neste trabalho.

Após a redução, é feita a transformação do modelo. Esta transformação consiste na obtenção dos valores de θ , τ , K_p do FOPDT, baseado no modelo obtido na identificação do ARX.

Para a transformação é necessário saber se o sistema está em malha aberta ou fechada. Para um modelo malha aberta somente é necessário o modelo ARX de ordem compatível com o algoritmo e também a taxa de amostragem T_o . Se o sistema estiver em malha fechada é necessário saber também os parâmetros do controlador PID.

O algoritmo de transformação do modelo sempre obtém mais de uma solução para o sistema. A escolha da melhor solução é feita após a aplicação de um degrau em cada sistema candidato, e então escolhida a solução que tiver o menor erro ITAE comparado com o modelo ARX obtido na identificação. Também é calculado o FIT da solução estimada com relação aos sistemas reais.

Abaixo serão detalhados os mesmos testes que foram descritos na etapa de identificação.

5.2.2.1. Teste local em malha aberta

Este teste utilizou o modelo obtido na estimação descrita na seção 5.2.1.1.

Após a redução do modelo da equação (76) foram obtidas as soluções candidatas do sistema em malha aberta mostradas na tabela 11.

Sol	K_p	τ	θ	ITAE	FIT
yE_1	2.00	102.05	46.62	9.38E+2	98.09
yE_2	2.00	23.24	204.09	2.45E+4	51.39

Tabela 11 - Resultados de transformação do modelo para teste local em malha aberta

Neste teste, a solução yE_1 mostrada abaixo foi escolhida por apresentar o menor índice de erro.

$$G_p = \frac{2.00}{102.05s + 1} e^{(-46.62s)} \quad (82)$$

Na figura 36 é apresentada a resposta do modelo estimado e do sistema real. Onde Y_{real} representa o dado obtido do sistema, Y_{arx} é o modelo ARX obtido na identificação, Y_{red}

é o modelo reduzido e Y_{est} a identificação final malha aberta após a transformação do modelo.

A solução final tem FIT de 98.09%. A análise do resultado mostrou que a identificação ficou muito próxima do modelo.

5.2.2.2. Teste remoto em malha aberta

Este teste é uma continuação do teste de identificação remota iniciado na seção 5.2.1.2.

Para a transformação do modelo da equação (77) foram obtidas as soluções candidatas do sistema em malha aberta. Os resultados são mostrados na tabela 12.

Sol	K_p	τ	θ	ITAE	FIT%
yE_1	2.00	22.68	201.12	7.57E2	54.94
yE_2	2.00	100.56	45.36	1.37E2	93.63

Tabela 12 - Resultados de transformação do modelo para o teste remoto em malha aberta

Neste exemplo, a solução da equação (83) foi escolhida por apresentar o menor índice de erro.

$$G_p = \frac{2.00}{100.56s + 1} e^{(-45.36s)} \quad (83)$$

Na figura 37 é mostrada a resposta do modelo estimado e do sistema real.

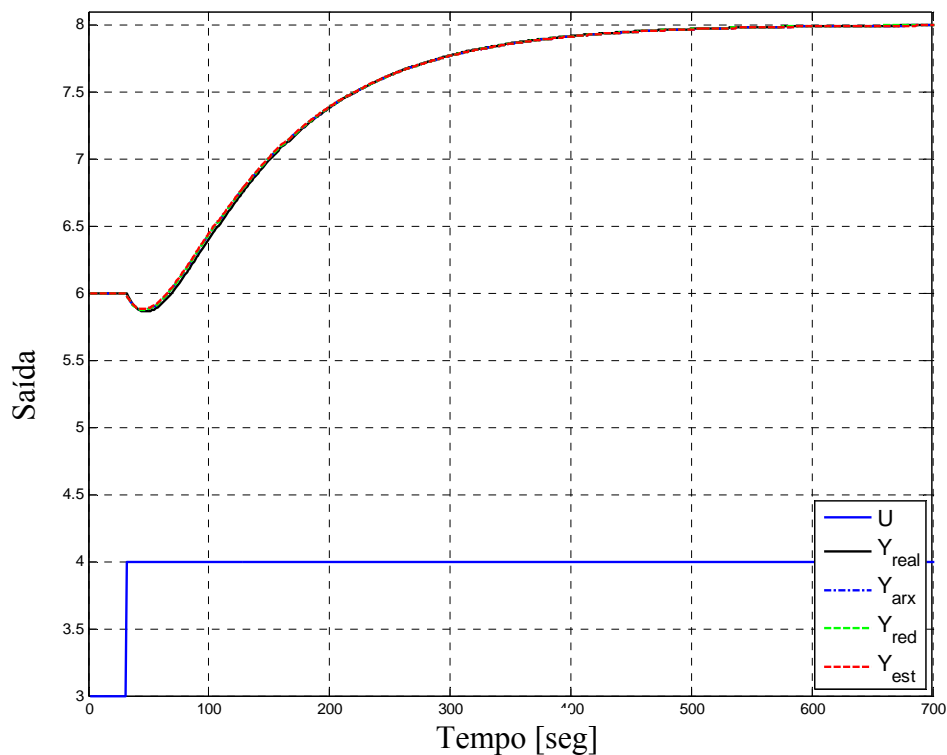


Figura 36 - Gráfico da identificação final do sistema em malha aberta em teste local

Para a figura 37, Y_{real} representa o dado obtido do sistema, Y_{arx} é o modelo ARX obtido na identificação, Y_{red} é o modelo reduzido e Y_{est} a identificação final malha aberta após a transformação do modelo. A solução final tem FIT de 93.63% que mostrou resultado muito próximo do sistema real.

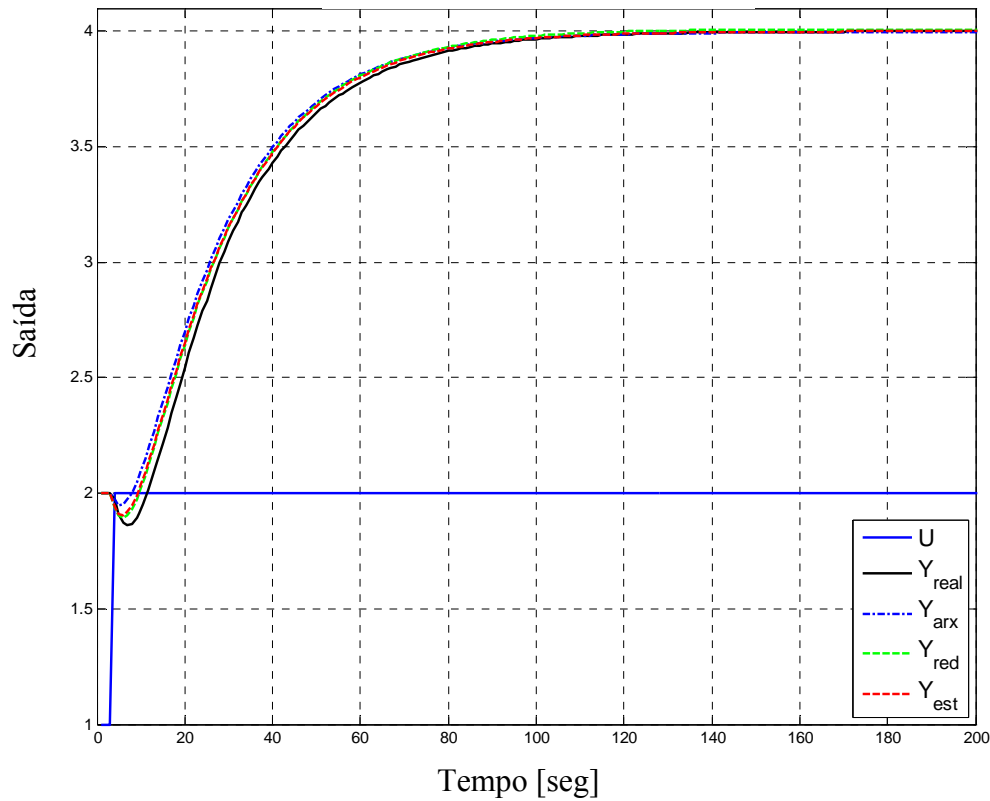


Figura 37 - Gráfico da identificação final do sistema em malha aberta em teste remoto

5.2.2.3. Teste com ruído em malha aberta

Os testes apresentados nesta seção são a continuação dos testes de identificação descritos na seção 5.2.1.3.

Para o ruído de 0.05 utilizando a equação (78) como modelo estimado e posterior a redução do modelo, foram obtidas as equações candidatas da transformação do modelo apresentadas na tabela abaixo.

Sol	K_p	τ	θ	ITAE	FIT%
yE_1	2.01	1.11	399.59	5.77E5	0.00
yE_2	2.01	199.79	2.22	2.06E5	75.32

Tabela 13 - Resultados de transformação do modelo para o teste em malha aberta com ruído de sinal

A solução que obteve o menor erro é mostrada abaixo:

$$G_p = \frac{2.01}{199.79s + 1} e^{(-2.22s)} \quad (84)$$

A figura 38 mostra a resposta do modelo estimado comparado com o sistema real.

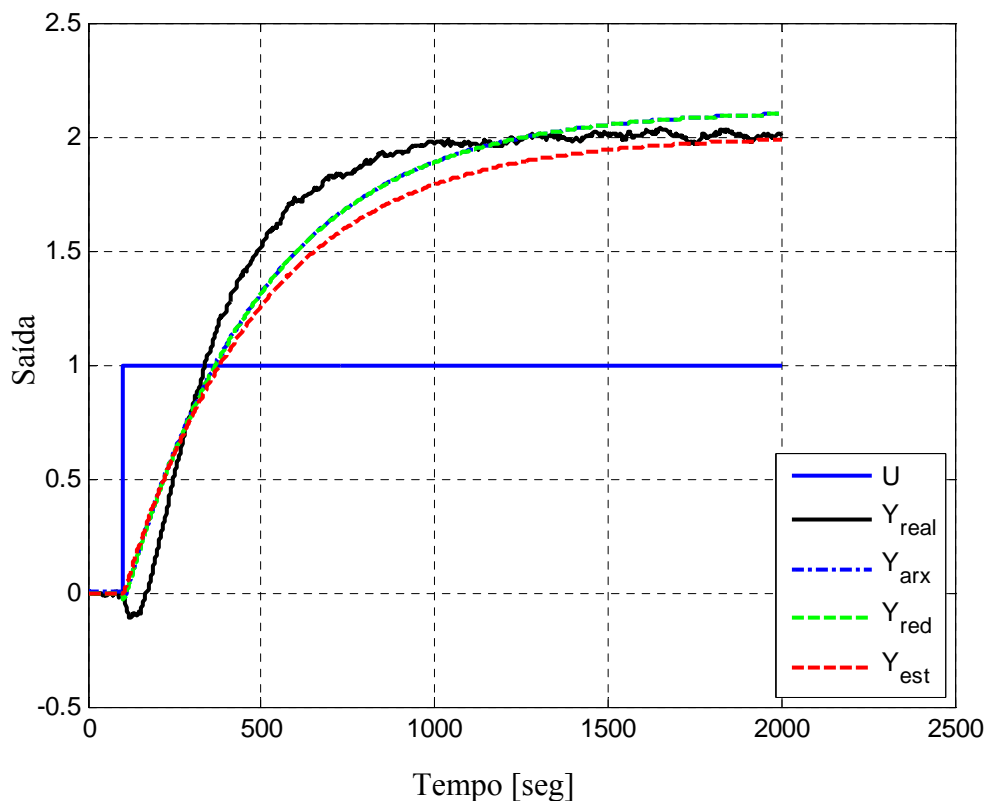


Figura 38 - Gráfico da identificação final do sistema com ruído de 0.05 em malha aberta

No gráfico da figura acima o sinal Y_{real} é o dado obtido do sistema, Y_{arx} representa o modelo ARX obtido na identificação, Y_{red} representa o modelo reduzido e Y_{est} a identificação final malha aberta após a transformação do modelo. A solução final apresentou FIT de 75.32%.

A análise dos resultados mostra que a identificação apresentou uma constante de tempo grande e um tempo morto pequeno comparado com o real.

A utilização de um filtro primeira ordem ($T_{Filter} = 4.5 [seg]$) mostrou melhoria na identificação com FIT=86.09%. A equação final obtida foi:

$$G_p = \frac{2.01}{146.1s + 1} e^{(-42.8s)} \quad (85)$$

O gráfico da figura 39 mostra os resultados finais da identificação onde os sinais mostrados são os mesmos da figura anterior.

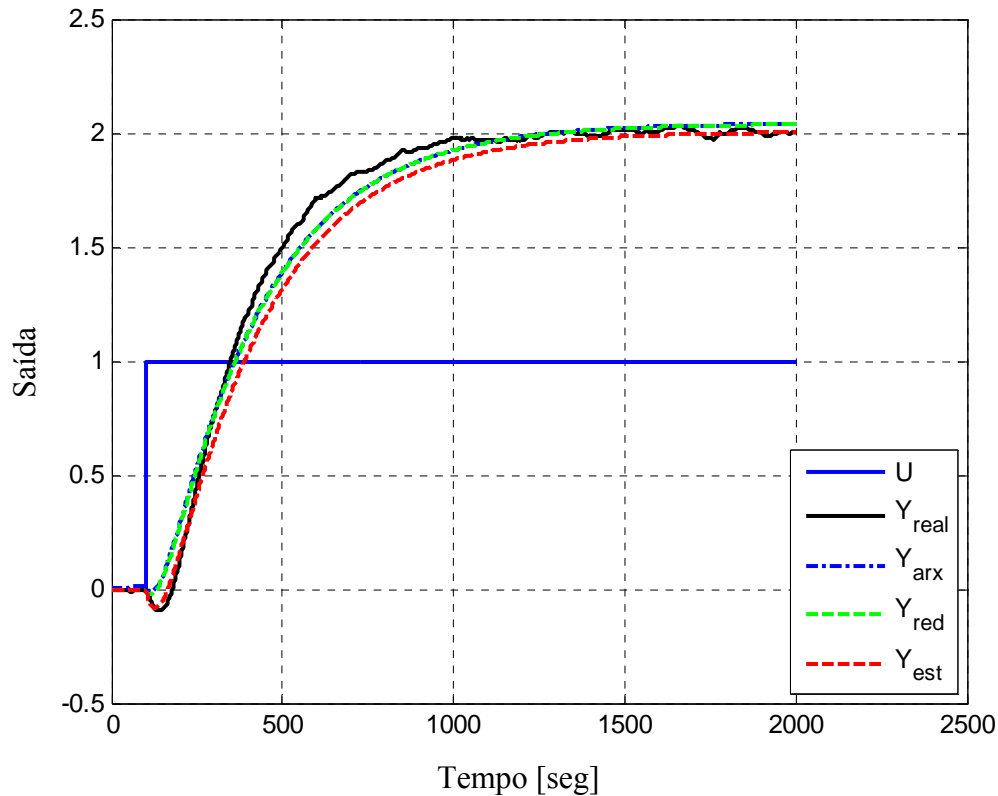


Figura 39 - Gráfico da identificação final do sistema com ruído de 0.05 e filtro para sistema em malha aberta

A utilização do filtro apresentou uma melhor identificação e conseqüentemente uma melhora também na transformação do modelo.

5.2.2.4. Teste local em malha fechada

Este teste utilizou o modelo obtido na estimação descrita na seção 5.2.1.4.

Baseado na equação do modelo ARX mostrada em (79) são obtidas as quatro soluções candidatas mostradas na tabela 14.

Sol	K_p	τ	θ	FIT%	ITAE
yE_1	1.05	46.91	70.11	73.39	5.00E+2
yE_2	1.50	64.81	57.54	83.43	2.78E+2
yE_3	-4.43	-160.41	32.78	0.00	2.02E+3
yE_4	-1.24	-19.66	14.37	0.00	5.26E+7

Tabela 14 - Resultados de transformação do modelo para o teste local em malha fechada

A solução yE_2 mostrada em (86) é a solução final malha aberta.

$$G_p = \frac{1.50}{64.81s + 1} e^{(-57.54s)} \quad (86)$$

Na figura 40 é apresentada a resposta do modelo estimado e do sistema real. Onde Y_{real} representa o dado obtido do sistema, Y_{arx} é o modelo ARX obtido na identificação, Y_{red} é o modelo reduzido e Y_{est} a identificação final malha aberta após a transformação do modelo.

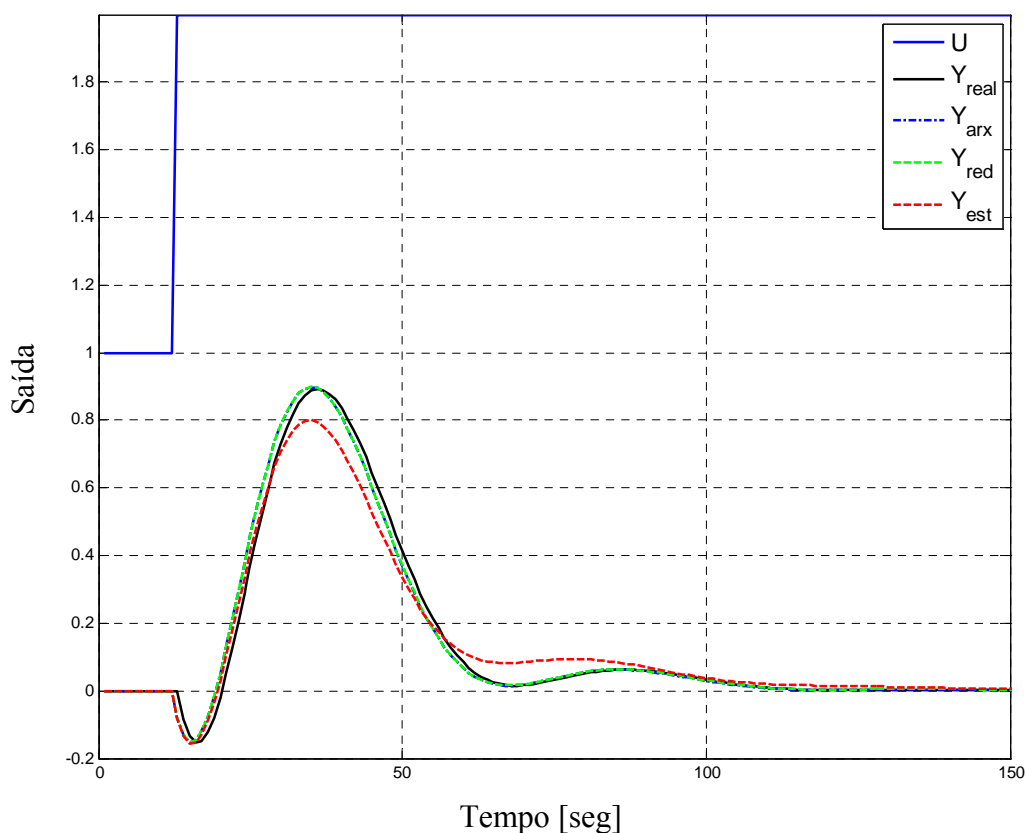


Figura 40 - Gráfico da identificação final do sistema em malha fechada em ambiente local

A solução final resultou em um FIT de 83.43%. A análise do gráfico mostrou que a identificação ficou próxima do modelo.

5.2.2.5. Teste remoto em malha fechada

Este teste é uma continuação do teste de identificação remota iniciado na seção 5.2.1.5.

Na transformação do modelo, obtiveram-se quatro soluções candidatas, que são apresentadas na tabela 15.

Sol	K_p	τ	θ	FIT	ITAE
yE_1	1.25	63.36	64.46	78.69	2.76E2
yE_2	1.25	63.36	64.46	78.69	2.76E2
yE_3	1.48	110.22	59.69	60.35	5.55E2
yE_4	-1.12	-12.39	10.26	0.00	2.97E12

Tabela 15 - Resultados de transformação do modelo para o teste remoto em malha fechada

Duas soluções candidatas apresentaram os melhores resultados, yE_1 e yE_2 . Isto ocorre quando o polinômio de transformação do modelo obtém raízes complexas. Portanto, a equação final obtida foi:

$$G_p = \frac{1.25}{63.36s + 1} e^{(-64.78s)} \quad (87)$$

O gráfico da figura 41 mostra os resultados finais da identificação. No gráfico, o sinal Y_{real} é o dado obtido do sistema, Y_{arx} representa o modelo ARX obtido na identificação, Y_{red} representa o modelo reduzido e Y_{est} a identificação final malha aberta após a transformação do modelo. A solução final apresentou FIT de 78.69%.

A análise do gráfico mostrou que a identificação ficou próxima do modelo apesar do modelo reduzido ter apresentado um erro grande em relação ao modelo ARX estimado.

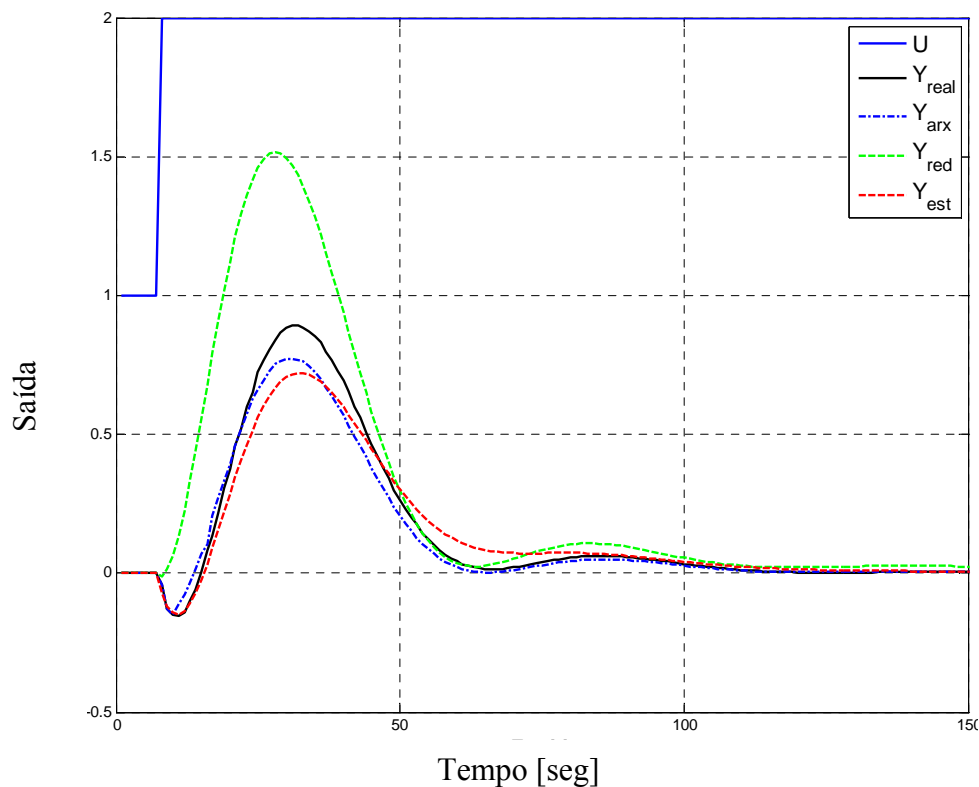


Figura 41 - Gráfico da identificação final do sistema em malha fechada em ambiente remoto

5.2.2.6. Teste com ruído em malha fechada

O teste desta seção é baseado no modelo obtido na seção 5.2.1.6 com ruído de $1E-5$.

A partir da equação obtida do modelo ARX mostrado em (81) foram obtidas as equações candidatas da tabela 16.

Sol	K_p	τ	θ	ITAE	FIT%
yE1	0.42	20.62	101.25	1.85E5	35.88
yE2	1.16	23.17	39.60	8.06E4	29.32
yE3	39.62	1900.78	29.33	5.57E4	45.54
yE4	-1.10	-1.04	1.01	3.51E45	0.00

Tabela 16 - Resultados da de transformação do modelo para o teste em malha fechada com ruído no sinal

A solução que mostrou o menor erro é mostrada abaixo.

$$G_p = \frac{39.62}{1901s + 1} e^{(-29.30s)} \quad (88)$$

Na figura 42 são mostrados os gráficos de todas as soluções estimadas. Onde o sinal Y_{real} é o dado obtido do sistema, Y_{arx} representa o modelo ARX obtido na identificação, Y_{red} representa o modelo reduzido e Y_{est} a identificação final malha aberta após a transformação do modelo.

A solução final apresentou FIT de 45.54%. A análise dos resultados mostrou resultados ruins na presença do ruído.

Em um outro teste, com a aplicação do filtro com $T_{Filter} = 1[seg]$ apresentou uma melhora significativa na identificação. A solução final é mostrada abaixo.

$$G_p = \frac{2.39}{133s + 1} e^{(-36.6s)} \quad (89)$$

O gráfico da figura 43 mostra os resultados finais da identificação onde os sinais mostrados são os mesmos da figura anterior.

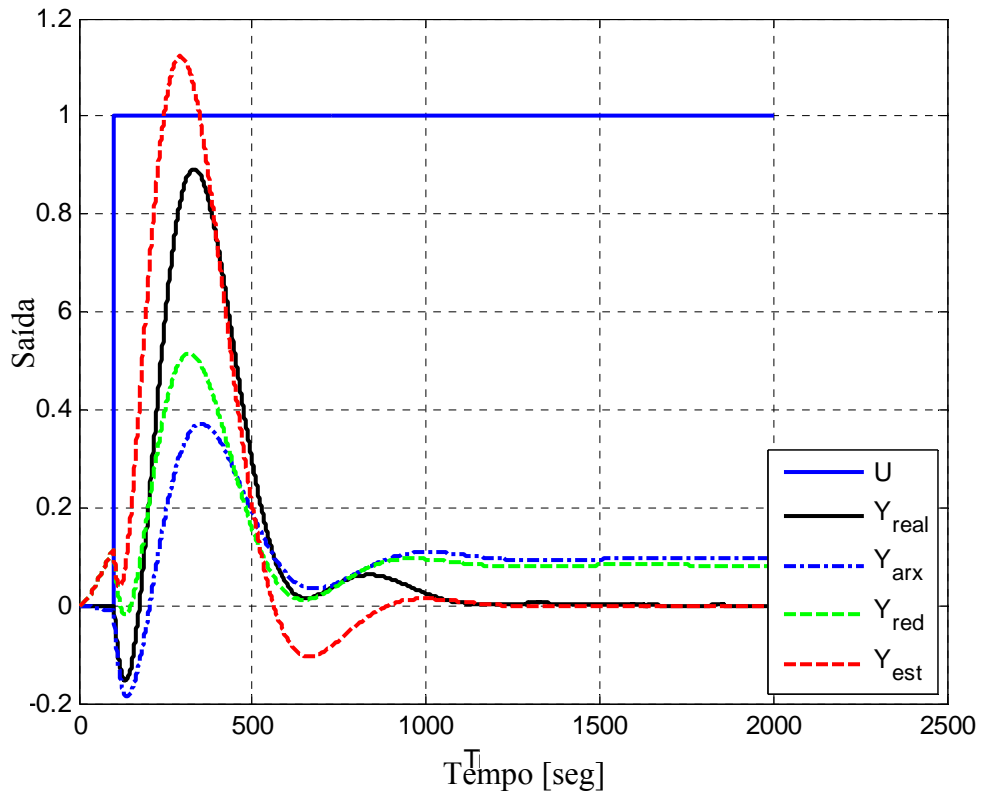


Figura 42 - Gráfico do sistema real e de todos os modelos obtidos na identificação

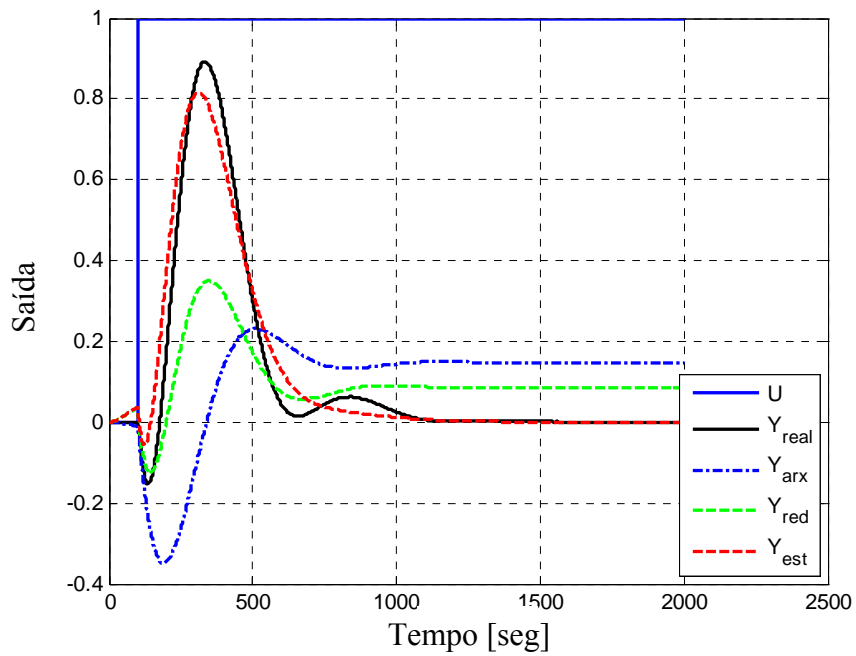


Figura 43 - Gráfico do sistema real e de todos os modelos obtidos na identificação com aplicação de filtro

A solução final apresentou FIT de 77.52%. Neste teste, apesar do sistema ter conseguido um valor alto de FIT a análise do gráfico mostra que tanto a identificação quanto a redução do modelo tiveram comportamento fora do modelo real.

5.2.2.7. Resumo dos testes de transformação do modelo

Abaixo estão sumarizados todos os testes realizados de transformação do modelo em malha aberta e fechada. Os dados são uma continuação dos testes de identificação mostrados anteriormente.

As tabelas estão divididas pelos diferentes sistemas e possuem somente os pontos principais da arquitetura. Para cada sistema foram realizados testes nas diferentes categorias. As categorias são as mesmas consideradas nas tabelas 9 e 10 da seção 5.2.1.

Primeiramente é feita a análise dos sistemas em malha aberta. Na tabela a seguir são mostrados todos os testes de transformação do modelo ordenados por sistema e categoria.

Teste	Sistema	Categoria	To [s]	Ruído	Filtro [s]	Kp	τ	θ	FIT	ITAE
1	1	1	0.5			2.99	149.75	19.50	99.35	1.05E04
2	1	2	1			3.00	150.66	18.21	98.74	5.69E03
3	1	3	2			3.00	159.05	17.98	97.71	1.16E03
4	1	4	0.5	5E-03		2.99	178.01	6.33	91.93	1.12E05
5	1	4	0.5	5E-03	0.9	2.99	161.82	17.71	95.85	6.17E04
6	1	5	0.5	5E-02		2.98	190.87	1.82	88.28	1.78E05
7	1	5	0.5	5E-02	2.9	2.99	176.74	16.10	91.42	1.28E05
8	1	6	0.5	5E-01		2.94	196.50	0.14	84.95	2.61E05
9	1	6	0.5	5E-01	2.1	2.96	191.00	5.82	87.83	2.07E05
10	2	1	0.50			2.00	100.52	48.87	99.26	1.89E03
11	2	2	1			2.00	102.15	46.48	98.09	9.77E02
12	2	3	5			2.00	100.56	45.36	93.63	1.37E02
13	2	4	0.50	5E-03		2.00	171.77	11.03	81.27	1.40E05
14	2	4	0.50	5E-03	1.5	2.00	114.23	53.45	92.96	3.82E04
15	2	5	0.50	5E-02		2.00	199.80	2.22	75.32	2.06E05
16	2	5	0.50	5E-02	4.5	2.01	146.13	42.81	86.09	1.01E05
17	2	6	0.50	5E-01		2.01	194.34	0.05	79.46	1.87E05

Teste	Sistema	Categoria	To [s]	Ruído	Filtro [s]	Kp	τ	θ	FIT	ITAE
18	2	6	0.50	5E-01	6.9	2.02	184.79	16.25	82.38	1.63E05
19	3	1	0.50			1.50	21.80	24.43	98.50	5.75E02
20	3	2	1			1.50	24.84	18.64	96.49	1.40E03
21	3	3	1			1.50	39.12	212.02	92.39	1.47E03
22	3	4	0.50	5E-03		1.49	49.61	4.15	83.17	3.73E04
23	3	4	0.50	5E-03	1.5	1.49	28.52	24.45	93.91	2.22E04
24	3	5	0.50	5E-02		1.47	54.86	1.35	78.59	9.05E04
25	3	5	0.50	5E-02	4.1	1.49	42.74	16.77	86.86	6.11E04
26	3	6	0.50	5E-01		1.41	38.11	0.01	64.11	2.44E05
27	3	6	0.50	5E-01	6.9	1.48	42.78	12.00	78.06	1.59E05
28	4	1	0.50			4.50	48.91	74.23	99.16	3.08E03
29	4	2	1			4.50	44.45	86.36	70.24	1.40E04
30	4	3	2			4.50	32.52	124.04	97.32	1.04E03
31	4	4	0.50	5E-03		4.51	152.44	9.19	71.88	3.48E05
32	4	4	0.50	5E-03	1.5	4.52	80.35	55.29	90.89	9.26E04
33	4	5	0.50	5E-02		4.54	183.57	1.86	66.22	4.95E05
34	4	5	0.50	5E-02	6.5	4.54	94.89	65.44	84.57	1.91E05
35	4	6	0.50	5E-01		4.63	-0.03	311.91	16.56	8.71E05
36	4	6	0.50	5E-01		4.61	147.78	19.18	79.80	3.52E05
37	5	1	0.50			9.00	7.25	13.75	97.45	2.68E03
38	5	2	1			9.01	11.25	18.92	81.25	1.01E04
39	5	3	3			9.00	9.70	12.12	90.35	1.10E03
40	5	4	0.50	5E-03		8.76	18.90	2.21	81.52	4.98E05
41	5	4	0.50	5E-03	1.5	8.87	8.76	16.77	90.87	3.00E05
42	5	5	0.50	5E-02		8.24	16.61	1.06	60.05	1.53E+06
43	5	5	0.50	5E-02	3.9	8.74	15.67	10.53	81.17	6.68E+05
44	5	6	0.50	5E-01		6.60	0.03	19.97	0.00	4.79E+06
45	5	6	0.50	5E-01	8.9	8.34	13.73	12.09	56.59	1.70E+06
46	6	1	0.5			25.00	14.75	34.50	98.46	9.95E3
47	6	2	1			25.01	131.64	38.19	79.96	2.13E4
48	6	3	5			25.00	11.98	61.56	93.12	5.93E2
49	6	4	0.5	5E-03		24.91	53.38	3.67	78.18	7.21E+05
50	6	4	0.5	5E-03	1.9	25.05	22.21	34.07	93.22	4.42E+05
51	6	5	0.5	5E-02		24.70	56.40	1.22	75.42	1.65E+06
52	6	5	0.5	5E-02	7.1	25.11	39.39	26.77	86.20	1.10E+06
53	6	6	0.5	5E-01		24.05	-0.01	61.35	54.04	4.58E+06
54	6	7	0.5	5E-01	10.5	25.25	44.80	13.59	76.61	2.81E+06

Tabela 17 - Resultados da transformação de modelo para sistemas em malha aberta

(1) Na tabela 17 a coluna “Categorias” refere-se aos seguintes testes :

- 1 – Teste de identificação realizado no MATLAB/SIMULINK sem ruído de forma *offline*.
- 2 – Teste de identificação realizado no tele-sintonizador com arquivo FBSIMU de forma *offline*.
- 3 – Teste de identificação realizado no tele-sintonizador usando *CyberOPC* com FBSIMU de forma *online*.
- 4 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de $5E-3$ de forma *offline*.
- 5 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de $5E-2$ de forma *offline*.
- 6 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de $5E-1$ de forma *offline*.

O gráfico comparativo entre os diferentes cenários apresentado a seguir baseia-se nos resultados da tabela 17.

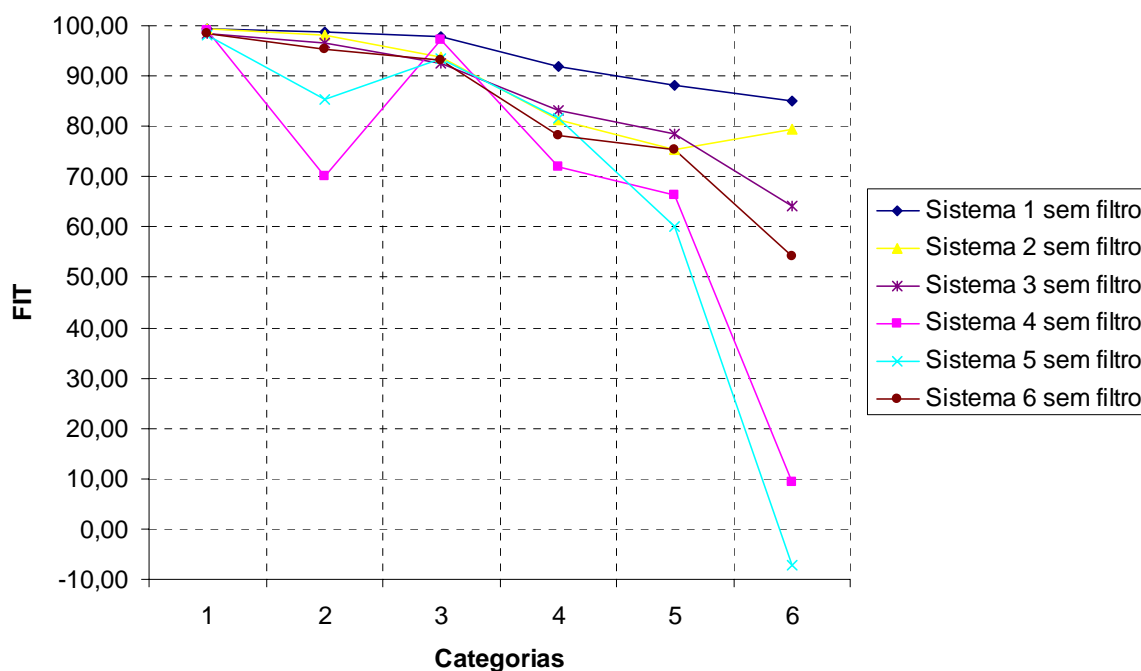


Figura 44 - Comparação da transformação do modelo dos diferentes sistemas em malha aberta nas diferentes categorias

A análise da figura 44 mostra que a transformação do modelo com os dados do MATLAB/SIMULINK (categoria 1) obteve valores próximos de 100%, validando, desta forma, o algoritmo. Para os testes locais (categoria 2), os sistemas 4 e 5 obtiveram boa identificação, porém não houve uma boa transformação do modelo. Isto devido ao fato que o algoritmo depende de ajustes matemáticos, que podem resultar equações com valores que não sejam adequados aos valores matemáticos. Para os testes de rede, as variáveis foram amostradas com taxas superiores a 1 segundo. Isto faz com que o algoritmo se adéque melhor a certos valores. Sendo assim, por exemplo, para o sistema 4, a uma taxa de

amostragem de identificação de 1 segundo, o resultado obtido mostrou-se pior quando comparado a uma identificação remota com taxa de amostragem de 2 segundos. Quando o ruído é incluído ao sinal (categorias 4, 5, 6), quanto melhor a identificação melhor será a transformação do modelo. Desta forma, o uso do filtro pode melhorar tanto a identificação quanto a transformação do modelo.

Na tabela a seguir são mostrados os resultados de transformação de modelo em malha fechada.

Teste	Sistema	Categorias	To [s]	Ruído	Filtro [s]	K_p	τ	θ	FIT [%]	ITAE
1	1	1	0.5			5.06	246.39	19.39	91.60	8.20E+03
2	1	2	1			-4.75	-220.26	16.89	70.56	1.89E+03
3	1	3	5			1.53	67.66	31.24	75.46	1.67E+02
4	1	4	0.5	1E-06		9.56	457.74	18.93	85.47	5.26E+03
5	1	4	0.5	1E-06	0.5	3.61	175.53	20.06	92.80	2.74E+03
6	1	5	0.5	1E-05		-9.64	-457.23	17.14	77.59	6.13E+03
7	1	5	0.5	1E-05	0.1	2.49	122.73	20.29	87.32	4.23E+03
8	1	6	0.5	1E-04		-0.96	-41.13	9.32	36.49	1.90E+04
9	1	6	0.5	1E-04	0.5	-1.33	-61.07	10.88	54.34	1.28E+04
10	2	1	0.5			2.24	110.24	48.72	95.15	6.19E+03
11	2	2	1			1.05	64.81	57.24	83.43	2.78E+02
12	2	3	5			1.25	63.36	64.47	78.69	2.77E+02
13	2	4	0.5	1E-06		2.54	124.45	45.97	84.01	1.35E+04
14	2	4	0.5	1E-06	0.5	2.01	101.54	48.87	93.69	5.46E+03
15	2	5	0.5	1E-05		39.62	1900.79	29.34	45.54	5.58E+04
16	2	5	0.5	1E-05	1	2.39	133.01	36.59	77.52	2.60E+04
17	2	6	0.5	1E-04		1.39	127.64	8.71	44.21	6.99E+04
18	2	6	0.5	1E-04	0.5	-176.03	-9948.65	20.58	45.22	6.39E+04
19	3	1	0.5			1.39	19.61	26.67	96.10	3.56E+03
20	3	2	2			1.48	18.85	26.91	92.88	1.37E+02
21	3	3	3			2.16	34.06	25.83	71.97	8.00E+02
22	3	4	0.5	1E-06		1.44	20.32	25.91	95.30	3.17E+03
23	3	4	0.5	1E-06	0.1	1.79	24.22	23.97	86.92	8.01E+03
24	3	5	0.5	1E-05		1.05	14.80	29.70	81.02	2.26E+04
25	3	5	0.5	1E-05	0.05	1.68	22.67	23.12	82.61	5.71E+03
26	3	6	0.5	1E-04		2.29	46.07	10.09	54.57	2.18E+04
27	3	6	0.5	1E-04	0.5	2.25	43.43	11.02	56.84	2.05E+04
28	4	1	0.5			4.51	48.35	75.50	99.35	4.26E+03
29	4	2	4			5.79	57.54	67.73	82.60	4.84E+03
30	4	3	5			5.85	61.28	73.76	78.62	6.65E+03
31	4	4	0.5	1E-06		3.84	41.17	77.58	84.75	2.36E+05

Teste	Sistema	Categorias	To [s]	Ruído	Filtro [s]	K _p	τ	θ	FIT [%]	ITAE
32	4	4	0.5	1E-06	0.001	3.93	42.08	76.34	84.58	2.22E+05
33	4	5	0.5	1E-05		1.54	16.30	97.78	43.03	8.98E+05
34	4	5	0.5	1E-05	0.001	1.55	16.37	97.45	43.15	8.96E+05
35	4	6	0.5	1E-04		0.28	2.47	114.22	0.00	9.31E+05
36	4	6	0.5	1E-04	0.001	11.55	138.28	4.89	16.98	6.24E+05
37	5	1	0.5			9.49	7.24	14.33	95.35	1.22E+04
38	5	2	1			9.63	6.91	14.22	92.59	3.46E+03
39	5	3	3			17.71	9.85	13.86	52.73	3.58E+03
40	5	4	0.5	1E-06		9.65	7.36	14.12	93.20	1.84E+04
41	5	4	0.5	1E-06	0.001	9.72	7.41	14.05	92.85	1.99E+04
42	5	5	0.5	1E-05		9.24	7.02	14.20	90.57	2.09E+04
43	5	5	0.5	1E-05	0.005	9.54	7.26	13.89	89.48	2.38E+04
44	5	6	0.5	1E-04		5.88	4.28	16.53	72.82	1.72E+05
45	5	6	0.5	1E-04	0.1	9.24	7.14	11.70	78.38	5.58E+04
46	6	1	0.5			25.70	14.83	-34.90	97.42	1.01E+05
47	6	2	1			26.09	14.97	33.83	95.77	4.24E+04
48	6	3	3			48.02	28.98	29.91	52.33	9.09E+04
49	6	4	0.5	1E-06		26.15	15.08	33.79	89.46	1.95E+05
50	6	4	0.5	1E-06	1E-05	26.15	15.08	33.78	89.46	1.95E+05
51	6	5	0.5	1E-05		22.04	12.61	33.84	79.63	1.06E+06
52	6	5	0.5	1E-05	0.01	23.48	13.49	32.01	78.74	8.36E+05
53	6	6	0.5	1E-04		8.38	4.41	36.47	39.59	5.28E+06
54	6	6	0.5	1E-04	0.1	8.40	4.43	36.56	39.66	5.28E+06

Tabela 18 - Resultados da transformação do modelo para sistemas em malha fechada

(1) Na tabela 18 a coluna “Categorias” refere-se:

- 1 – Teste de identificação realizado no MATLAB/SIMULINK sem ruído de forma *offline*.
- 2 – Teste de identificação realizado no MATLAB com arquivo FBSIMU de forma *offline*.
- 3 – Teste de identificação realizado no MATLAB usando *CyberOPC* com FBSIMU de forma *offline*.
- 4 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de 1E-6 de forma *offline*.
- 5 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de 1E-5 de forma *offline*.
- 6 – Teste de identificação realizado no MATLAB/SIMULINK com ruído de 1E-4 de forma *offline*.

Baseado nos resultados da tabela 18, o gráfico da figura 45 foi elaborado considerando os diferentes sistemas nas diferentes categorias.

A análise da figura 45 mostra que a transformação do modelo com os dados do MATLAB/SIMULINK (categoria 1) obteve valores, em média, maiores que 95%, o que comprova a validade do algoritmo também para identificação em malha fechada.

Para os testes locais (categoria 2), os sistemas 4 e 5 obtiveram boa identificação, porém a transformação do modelo não se mostrou satisfatória. Isto pode ser explicado pela adequação matemática comentada anteriormente para o sistema malha aberta. Para os testes

de rede, as variáveis foram amostradas com taxas superiores a 1 segundo. Isto faz com que o algoritmo se ajuste melhor matematicamente a certos valores.

Quando o ruído é incluído ao sinal (categorias 4, 5, 6), quanto melhor a identificação melhor será a transformação do modelo.

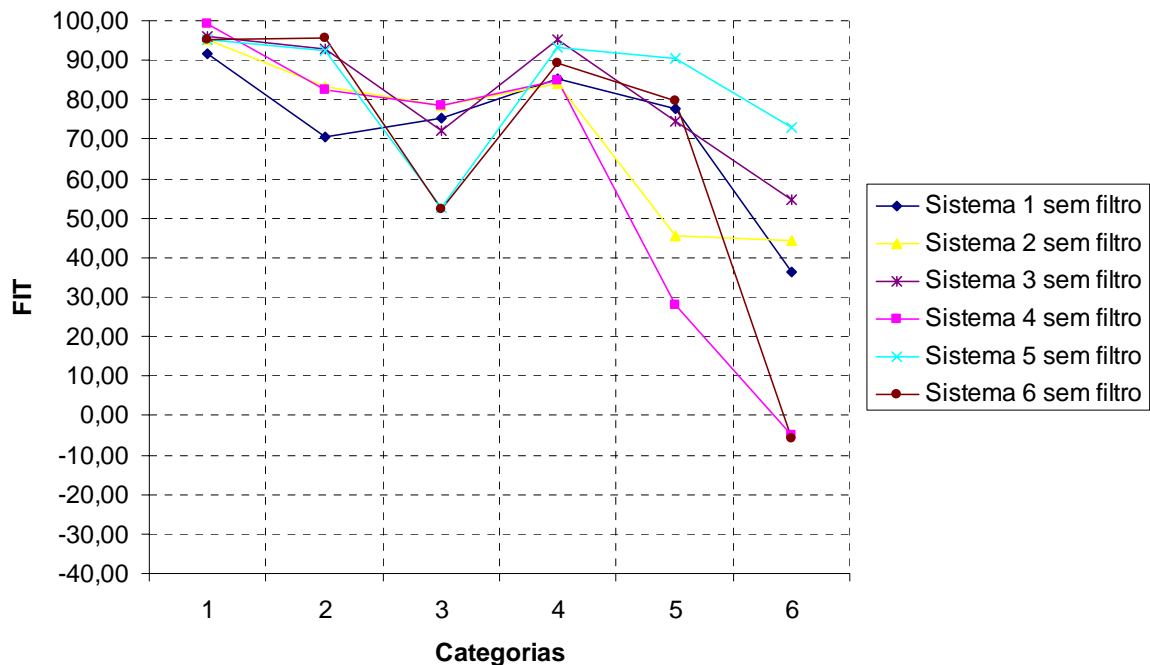


Figura 45 - Comparação da transformação do modelo dos diferentes sistemas em malha fechada para as diferentes categorias da tabela 20

5.2.3. Sintonia

Após a obtenção do modelo malha aberta, é realizada a última etapa do tele-sintonizador: propor uma nova sintonia, conforme ilustra a figura 18. Os métodos utilizados nos testes são os métodos propostos na seção 3.4. Como a sintonia é dependente da obtenção de um modelo, para alguns resultados não foi obtida a sintonia devido ao modelo não conseguir apresentar resultados satisfatórios.

Abaixo são detalhados os principais testes realizados.

5.2.3.1. Teste local em malha aberta

Este teste utilizou o modelo obtido na estimação descrita na seção 5.2.2.1.

Foram realizadas as sintonias com os métodos propostos baseado no modelo obtido em (82) e os resultados são mostrados na figura a seguir.

Método	K_c	T_i	ITAE
Atual	0.9850	93.2369	8.94E+9
ITSE	1.3408	301.8756	2,40E+10
ITAE	0.9234	88.8742	8.88E+9
IMC	0.5760	102.0469	1.71E+10

Tabela 19 - Resultados de sintonia para teste local em malha aberta

O gráfico abaixo mostra a simulação de variação de carga para cada sintonia. O método que apresentou a melhor performance foi o ITAE.

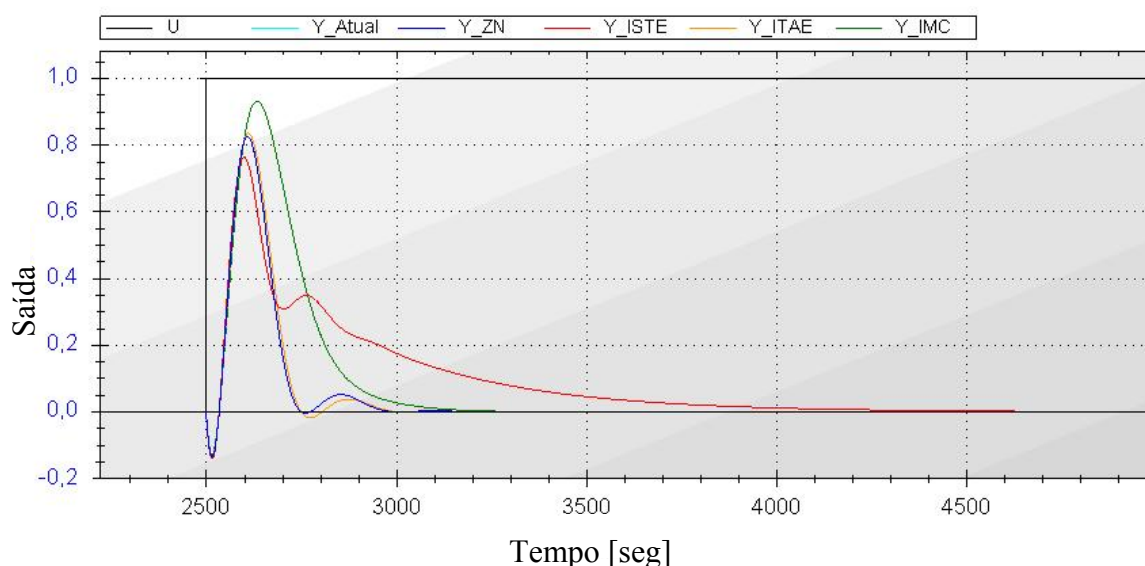


Figura 46 - Gráfico de sintonia dos diferentes métodos para teste local em malha aberta

5.2.3.2. Teste remoto em malha aberta

Para este teste remoto em malha aberta foi utilizado o modelo (83) obtido na estimação descrita na seção 5.2.2.2. Então, foi realizada a sintonia com os métodos propostos e os resultados são mostrados na tabela abaixo.

Método	K_c	T_i	ITAE
Atual	1.055	84.08	2.26E+8
ITSE	1.43	303.28	5.64E+8
ITAE	0.99	81.91	2.15E+8
IMC	0.62	98.51	4.15E+8

Tabela 20 - Resultados de sintonia para o teste remoto em malha aberta

O método que teve o menor erro foi o método ITAE. Os gráficos da simulação de variação de carga para cada sintonia são mostrados na figura a seguir.

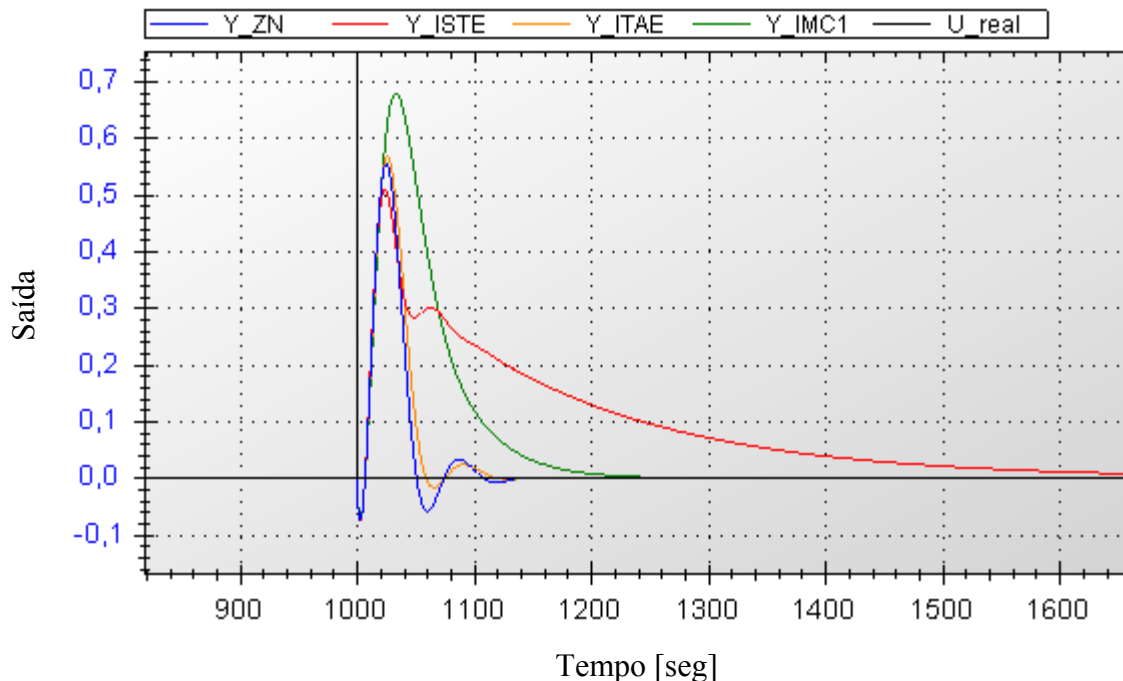


Figura 47 - Gráfico de sintonia dos diferentes métodos para teste remoto em malha aberta

5.2.3.3. Teste com ruído em malha aberta

Este teste utilizou o modelo obtido na estimação descrita na seção 5.2.2.3 para o ruído de 0.05. A tabela abaixo mostra os resultados das sintonias propostas baseados no modelo (84).

Método	K_c	T_i	ITAE
Atual	0.90	100	1.53E5
ITSE	40.6139	4904.5628	7.32E64
ITAE	31.3956	14.9336	2.1024
IMC	4.4436	199.7963	0.0222

Tabela 21 - Resultados de sintonia para o teste em malha aberta com ruído de sinal

O teste com o sistema obtido em malha fechada com as sintonias propostas mostrou características oscilatórias.

Para o modelo estimado com o filtro de primeira ordem mostrado em (85) foi possível obter parâmetros estáveis de sintonia. O método ITAE foi o que apresentou o melhor desempenho. A tabela abaixo mostra as novas sintonias propostas.

Método	K_c	T_i	ITAE
Atual	0.90	100	1.06E5
ITSE	2.0229	558,8853	7.32E5
ITAE	1.4128	94.4615	8.03E4
IMC	0.8903	146.1284	1.84E5

Tabela 22 - Resultados de sintonia para o teste em malha aberta com ruído de sinal e filtro

Na figura a seguir é mostrada a resposta do sistema a uma variação de carga.

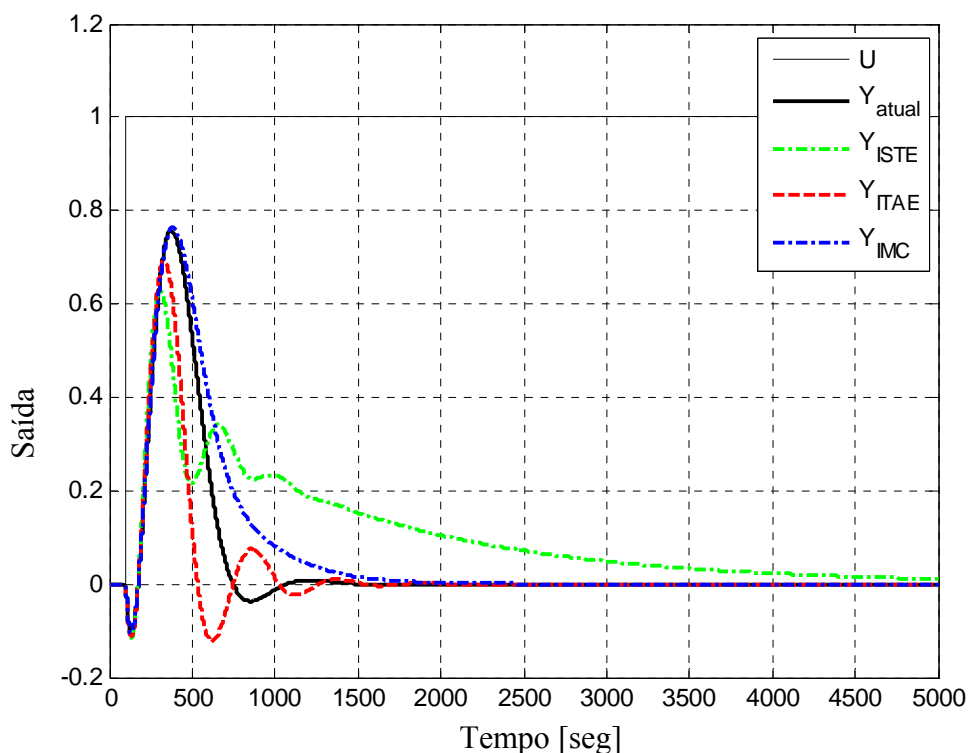


Figura 48 - Gráfico de sintonia dos diferentes métodos para teste com ruído em malha aberta

5.2.3.4. Teste local em malha fechada

Este teste utilizou o modelo obtido na estimação descrita na seção 5.2.2.4. A sintonia com os métodos propostos e os resultados são mostrados na tabela 23.

Método	K_c	T_i	ITAE
Atual	0.90	100	3.09E3
ITSE	0.9210	175.20	6.07E3
ITAE	0.6239	99.2483	4.51E3
IMC	0.3846	80.1820	5.92E3

Tabela 23 - Resultados de sintonia para o teste local em malha fechada

O gráfico a seguir apresenta a resposta à variação de carga dos vários métodos de sintonia. A sintonia atual obteve o menor erro.

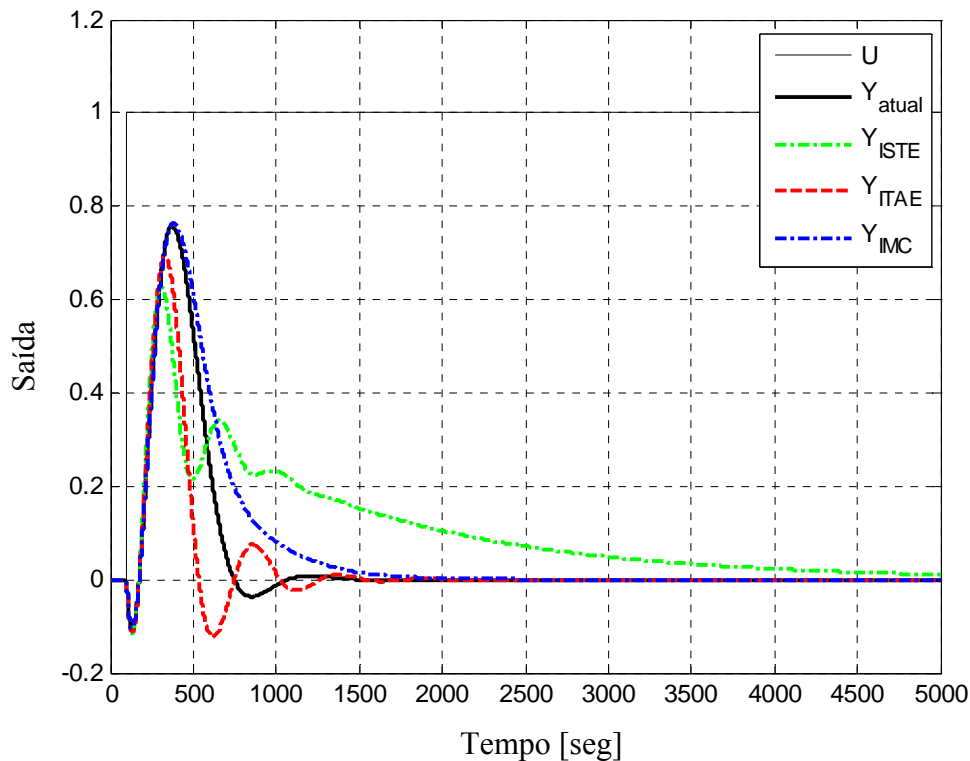


Figura 49 - Gráfico de sintonia dos diferentes métodos para teste local em malha fechada

5.2.3.5. Teste remoto em malha fechada

Este teste utilizou o modelo obtido na estimação descrita na seção 5.2.2.5.

Baseado no modelo malha aberta obtido na equação (87) foi realizado a sintonia com os métodos propostos e os resultados são mostrados na tabela abaixo.

Método	K_c	T_i	ITAE
Atual	0.90	100	2.75E4
ITSE	0.9712	114.6494	2.94E4
ITAE	0.6512	97.6158	3.69E4
IMC	0.3984	63.3611	3.87E4

Tabela 24 - Resultados de sintonia para o teste remoto em malha fechada

O gráfico a seguir apresenta a resposta à variação de carga dos vários métodos de sintonia. O método de sintonia atual, novamente, obteve o menor erro.

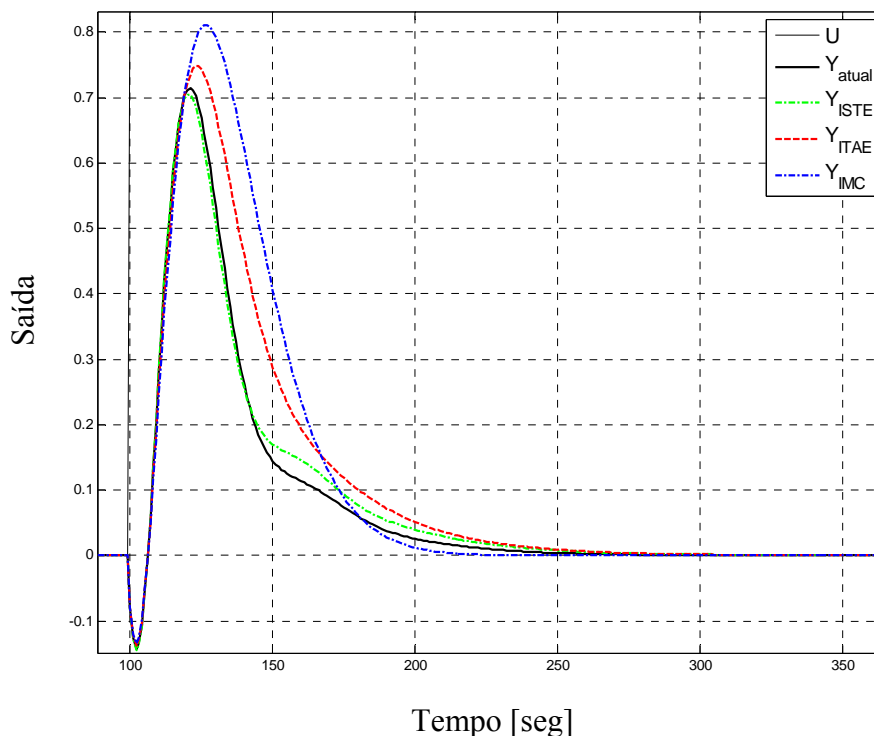


Figura 50 - Gráfico de simulação da resposta do sistema identificado à variação de carga

5.2.3.6. Teste com ruído em malha fechada

Este teste utilizou o modelo obtido na estimação descrita na seção 5.2.2.6 com ruído de $1E-5$. A tabela a seguir apresenta os resultados de sintonia baseado no modelo obtido em (88).

Método	K_c	T_i	ITAE
Atual	0.90	100	5.42E5
ITSE	1.6497	40734.95	7.19E6
ITAE	1.2658	166.3226	3.37E5
IMC	0.2184	1900.7895	2.80E7

Tabela 25 - Resultados de sintonia para o teste em malha fechada com ruído de sinal

Após a aplicação do filtro de primeira ordem ($T_{Filter} = 1[seg]$) foi possível obter parâmetros estáveis de sintonia, porém houve oscilação para alguns métodos. O método atual foi o que obteve o menor erro. A nova sintonia proposta é mostrada na seguinte tabela.

Método	K_c	T_i	ITAE
Atual	0.90	100	9.85E4
ITSE	1.7994	527.538	7.50E5
ITAE	1.2592	82.4295	1.37E5
IMC	0.7946	133.0086	1.71E5

Tabela 26 - Resultados de sintonia para o teste em malha fechada com ruído de sinal e filtro

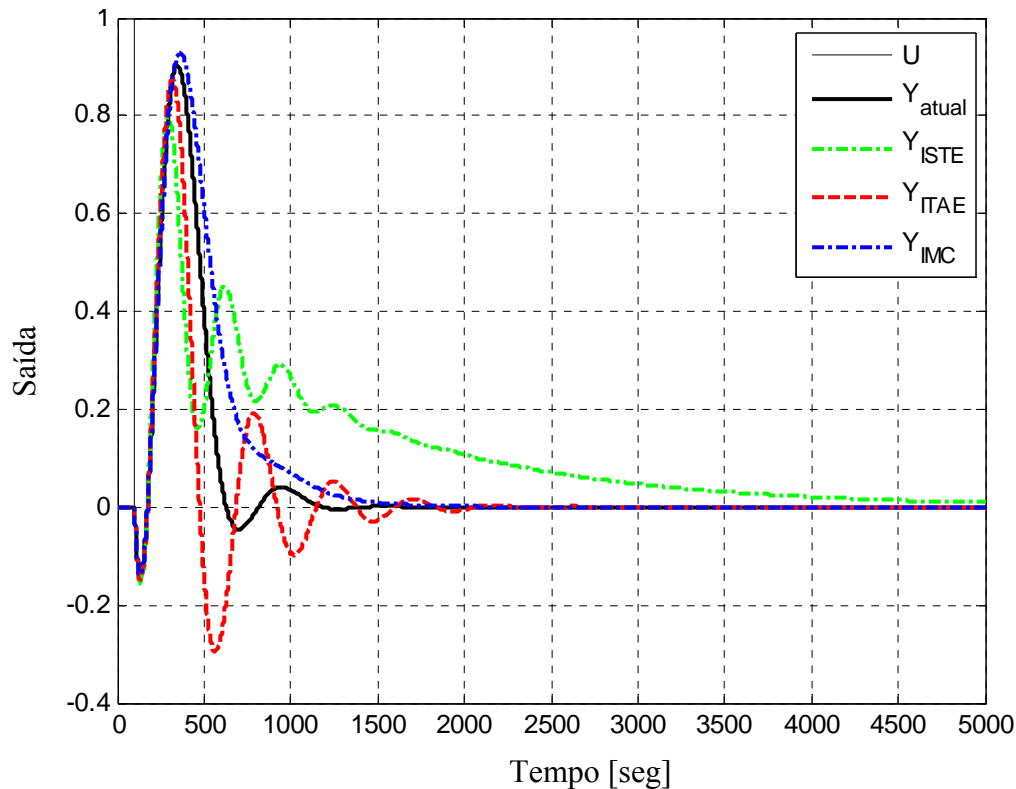


Figura 51 - Gráfico de sintonia dos diferentes métodos para teste com ruído em malha fechada

5.2.4. Análise final dos resultados

Abaixo é realizada uma análise de todos os testes realizados com o objetivo de ser um guia na determinação de métodos de identificação e sintonia de sistemas FOPDT. Para isso foram mapeados os diferentes sistemas com relação à razão θ/τ nas diferentes categorias. A análise divide-se nas duas fases de identificação criadas neste projeto: a identificação ARX e a transformação de modelo tanto em malha aberta quanto em malha fechada.

As figuras 52 e 53 mostram os gráficos de identificação do modelo ARX e transformação do modelo para os sistemas em malha aberta.

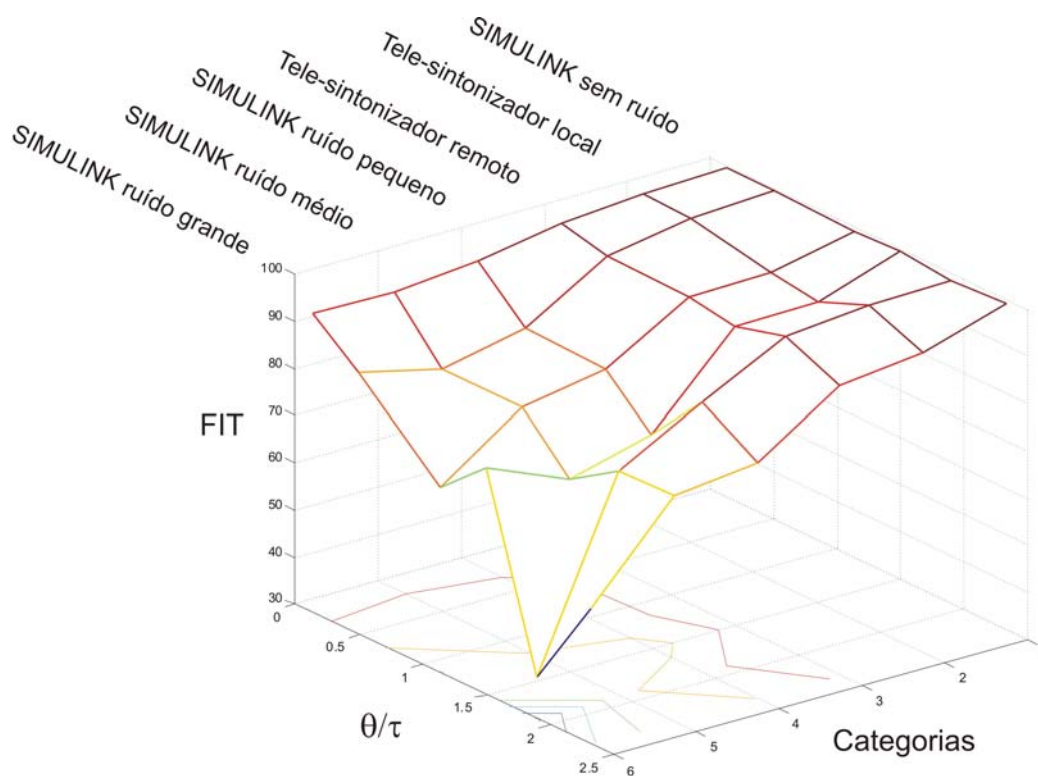


Figura 52 - Gráfico de Identificação do modelo ARX para sistemas em malha aberta

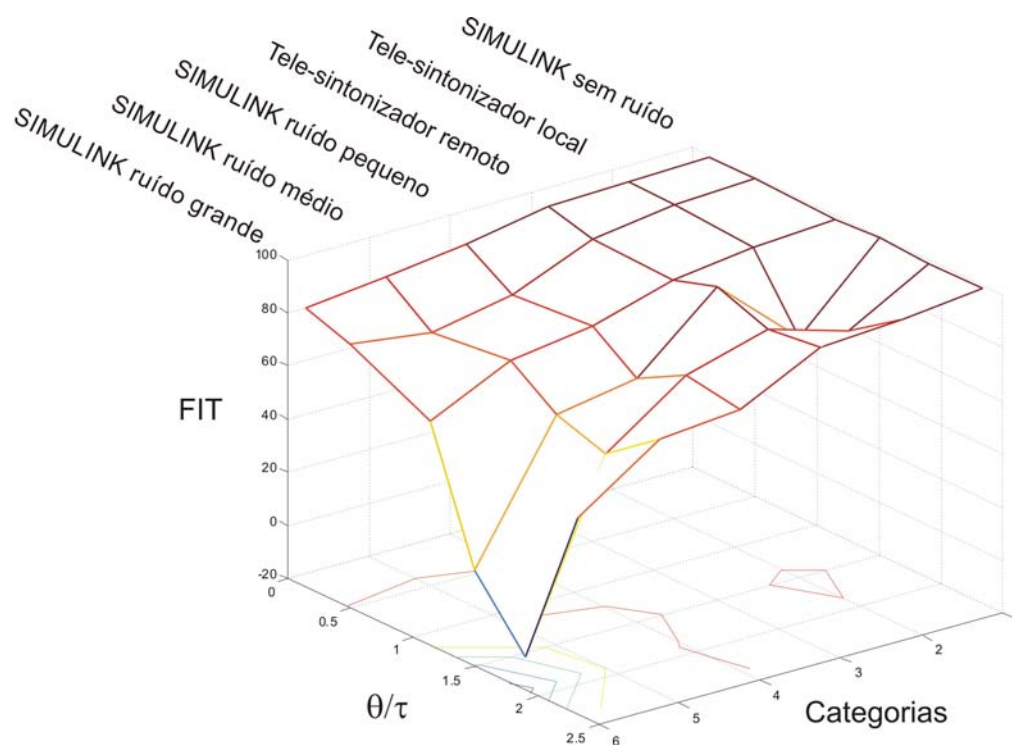


Figura 53 - Gráfico de transformação de modelo para sistemas em malha aberta

A análise da identificação do modelo ARX (figura 52) mostra que para qualquer θ/τ estudado não existe diferença significativa se a identificação é local ou remota (categorias 1,

2 e 3). Porém, quando incluído ruído no sinal (categorias 4, 5 e 6), um aumento em θ/τ mostrou uma degradação na identificação.

Na análise da transformação do modelo (figura 53) pode ser aplicada a mesma conclusão da identificação. Entretanto, como o algoritmo de transformação é baseado em cálculos polinomiais, é necessário verificar se um melhor modelo de identificação não se encontra na vizinhança do sinal amostrado. Para este projeto foram utilizadas diferentes amostragens do sinal para conseguir um modelo que trouxesse uma melhor transformação do modelo.

Nas figuras 54 e 55 são mostrados os gráficos de identificação do modelo ARX e transformação do modelo para os sistemas em malha fechada.

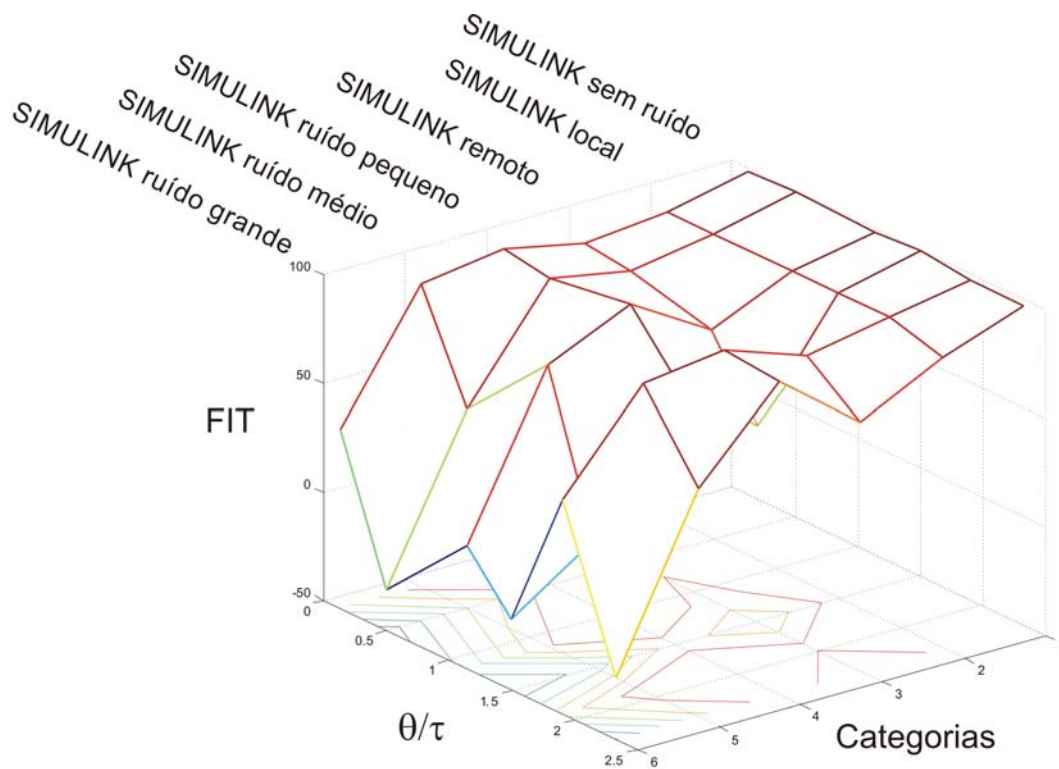


Figura 54 - Gráfico da identificação do modelo ARX para sistemas em malha fechada

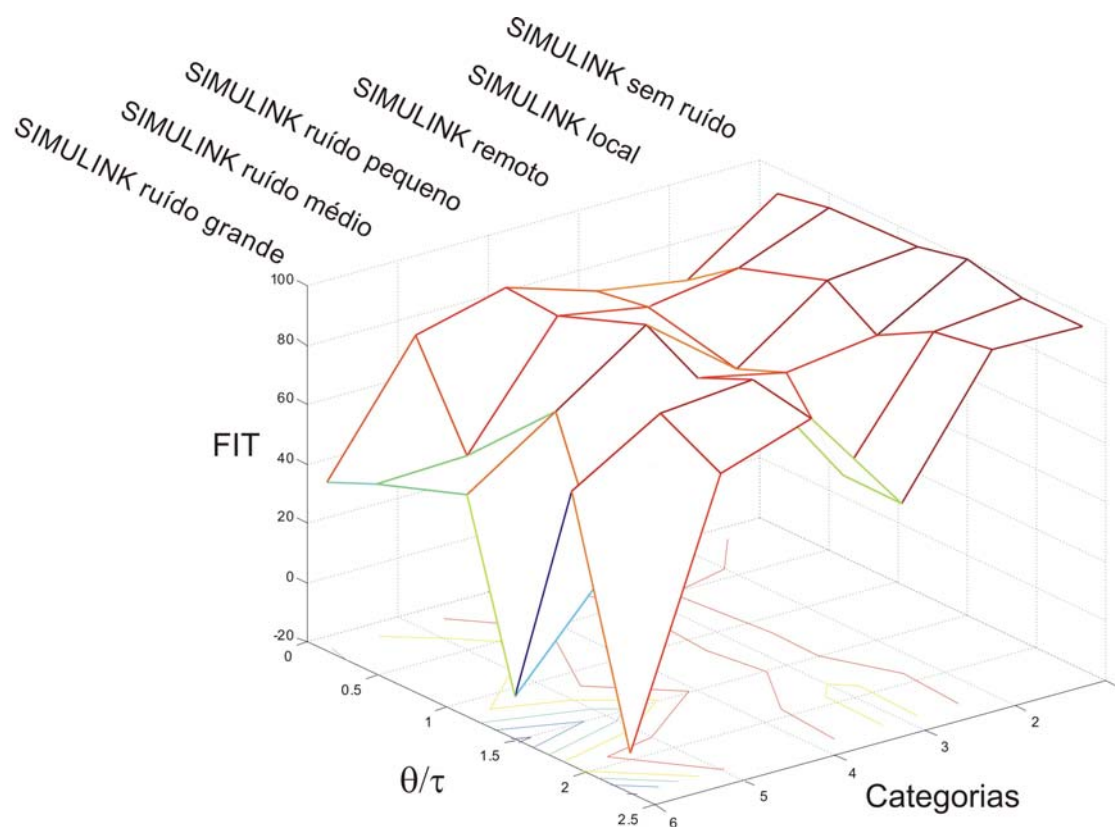


Figura 55 - Gráfico de transformação do modelo para sistemas em malha fechada

A análise do gráfico da identificação (figura 54) mostra que somente a identificação local (categorias 1 e 2) apresentou uma boa identificação para qualquer θ/τ . Para o ambiente de rede com *CyberOPC* (categoria 3), a identificação apresentou uma degradação devido a incertezas de rede. E quando incluído ruído no sinal (categorias 4, 5 e 6), os sistemas reagem de diferentes formas onde a utilização de filtro de primeira ordem pouco contribuiu para a melhora da identificação.

Na análise da transformação do modelo (figura 55) somente os testes com MATLAB/SIMULINK (categoria 1) conseguiram trazer resultados uniformes para qualquer θ/τ . Nos testes locais e remotos sem ruído (categorias 2 e 3), a transformação do modelo já mostra dificuldade em apresentar bons resultados devido ao modelo identificado com menor qualidade e as aproximações nas fórmulas. Neste caso, os testes remotos trouxeram piores resultados do que os testes locais. E para os testes com ruído no sinal, a transformação do modelo também reage de diferentes formas para a relação θ/τ .

Por fim, a sintonia dependerá muito de uma boa identificação. Os diferentes métodos utilizados para sintonia mostraram resultados muito próximos. E nenhum se destacou dentre todos os métodos estudados.

6. Conclusão

1) O projeto e o desenvolvimento da arquitetura do tele-sintonizador descrita no capítulo 4 mostraram-se possíveis, destinada à aplicação de sintonia remota de malhas de controle de sistemas FOPDT com controlador PID, conforme o objetivo número um desta pesquisa. Com a aquisição de dados foi possível desenvolver e validar a arquitetura tanto localmente quanto remotamente como proposto no item 1. O tele-sintonizador também é capaz de identificar o modelo, validar a identificação, propor uma sintonia e simular o antigo e o novo sistema conforme os objetivos 2 e 3 propostos.

2) Os resultados experimentais de identificação mostraram que para modelo onde o sinal se apresenta sem incertezas de rede ou ruído (simulação no MATLAB/SIMULINK), é possível utilizar modelos ARX de ordem baixa como proposto no modelo matemático (ordem 2 para sistemas malha aberta e ordem 3 para sistemas malha fechada), porém quando existir qualquer incerteza no sinal, são necessários sistemas de ordem mais alta.

3) A seção 5.2.1 apresentou resultados de identificação satisfatórios utilizando o modelo ARX para os sistemas com diferentes características (θ/τ) tanto para malha aberta quanto para malha fechada, em ambiente local para sinal sem ruído. O sistema malha fechada apresentou maior dificuldade na identificação quando comparado ao sistema malha aberta, por se tratar de um sistema com maior complexidade de identificação.

4) Para a identificação com ruído no sinal os resultados mostraram que a identificação de sistemas em malha aberta é menos susceptível a ruído de que a identificação de sistemas em malha fechada. Para sistemas malha fechada, valores de ruído com valores muito pequenos levam a uma identificação e sintonia com resultados oscilatórios. Para casos com ruídos deveriam ser utilizados outros métodos de identificação como filtro de Kalman ou outro método paramétrico menos susceptível a ruído.

5) No contexto da transformação do modelo, o algoritmo proposto neste trabalho foi validado tanto para malha aberta quanto para malha fechada, como mostraram os resultados da seção 5.2.2. Porém, como o método é baseado em cálculos de equações polinomiais, portanto, quanto melhor a identificação, melhor será a transformação do modelo. Também foi verificado que uma transformação do modelo ruim pode ser melhorada através da utilização de métodos que aproximam melhor a equação do ARX. Neste caso foi utilizado

um filtro para suavizar o sinal amostrado. Este filtro foi realizado espaçando as amostras em até 5 vezes. Isto fez com que a transformação do modelo mostrasse resultados melhores em alguns casos.

6) Para a sintonia remota, a arquitetura proposta utilizando *CyberOPC* e a reconstrução dos dados mostrou resultados próximos aos resultados *offline* e aos de identificação local como mostraram os resultados no capítulo 5, fato que viabiliza a arquitetura.

Como proposta de trabalhos futuros, sugere-se a melhoria da identificação através de métodos mais robustos considerando o ruído dos sistemas como filtro de Kalman. Na transformação de modelo, sugere-se tratar raízes complexas. Quanto à filtragem do sinal, sugere-se uma verificação automática de sinais ruins e a habilitação automática de diferentes filtros para a respectiva identificação. Também, sugere-se uma forma de configurar a ordem do modelo ARX que mais se ajuste ao modelo identificado. Para a comunicação em rede, as sugestões são diminuir a ocupação de banda de rede utilizando OPC e *CyberOPC*, através da diferenciação de dados estáticos e dinâmicos e também envio do sinal orientado a evento. Sugere-se complementarmente, validar o tele-sintonizador em uma planta real em ambiente da *internet* e testar o procedimento completo em condições variadas de carregamento de rede. Por fim, a utilização do tele-sintonizador remoto deve ser avaliada em quesitos de segurança e de garantia da qualidade de serviço quando vários tele-sintonizadores forem utilizados paralelamente em diferentes malhas de uma mesma planta industrial.

Referências

- AGUIRRE, L. A. (2004). Introdução a Identificação de Sistemas, Técnicas Lineares e Não Lineares aplicadas a sistemas reais. 2ª edição, 2004, Editora UFMG.
- AKTAN, B.; BOHUS, C.A.; CROWL, L.A.; SHOR, M.H (1996). Distance learning applied to control engineering laboratories. In: IEEE Transactions on Education, Vol. 39, pp 320 – 326, Aug. 1996.
- ALTUS (2007). Product data sheet: PID-Tool. Disponível em: < <http://www.altus.com.br> >. Acesso em: 27 dez. 2007.
- ANDRADE, S.S; MACEDO, R. J. A. (2005) – A component-Based Real-Time Architecture for Distributed Supervision and Control Applications – 10th IEEE Conference on Emerging Technologies and Factory Automation – ETFA 2005.
- ANG, K. H.; CHONG, G.; LI, Y. (2005). PID Control System Analysis, Design, and Technology. In: IEEE Transaction on Control Systems Technology, Vol 13, No 4, July 2005.
- ASTRÖM, K.J.; HÄGGLUND, T. (1984). Automatic tuning of simple regulators with specifications on phase and amplitude margins. In: Automatica, Vol. 20, No 5, pp 645-651, 1984.
- ASTRÖM, K.J.; HÄGGLUND, T., (1995), PID Controllers: theory, design and tuning. 2ª Edição, 1995, Editora Instrument Society of America (ISA)
- ASTRÖM, K.J.; HÄGGLUND, T. (2001). The future of PID control. In: Control Engineering Practice 9 (2001) 1163-1175.

- ATOS (2007). Product data sheet: PID-Analyser. Disponível em: < <http://www.atos.com.br> >. Acesso em: 27 dez. 2007.
- AVOY, T. M.; JOUNELA, S.L.J.; PATTON, R.; PERRIER, M.; WEBER, H.; GEORGAKIS, C. (2004). Milestone report for area 7 industrial applications. In: Control Engineering Practice 12 (2004) 113-119.
- BALDA, P.; SCHLEGEL, M.; STĚTINA, M. (2004); An open autotuning PID control remote laboratory based on industrial standards. In Preprints of the IFAC Workshop IBCE'04 Internet Based Control Education, 1–5. Grenoble, France.
- BATUR, C.; MA, Q.; LARSON, K., KETTENBAUER, N. (2000). Remote tuning of a PID position controller via internet. American Control Conference , Chicago, 2000.
- BLACKWELL, R. (1996). Microsoft Solutions for Manufacturing Industry. In IEE colloquium on next generation manufacturing: future trends in manufacturing and supply chain management, 1996, London, Proceedings, London, p.647-727.
- BOGGS, D.R., MOGUL, J.C., KENT, C.A. (1988). Measured capacity of an Ethernet: Myths and reality. In Symposium proceedings on Communications architectures and protocols (ACM SIGCOMM' 88), pages 222-234, 1988.
- BRANDÃO, D. (2005). Ferramenta de simulação para projeto, avaliação e ensino de redes *fieldbus*. 151f. Tese (Doutorado em Engenharia Mecânica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2005.
- CALVO, I.; MARCOS, M.; ORIVE, D.; SARACHAGA, I. (2006). A methodology based on distributed object-oriented technologies for providing remote access to industrial plants. Control Engineering Practice 14 (2006), pp. 975-990

- CAON, J.R. (1999) – Controladores PID industriais com sintonia automática por realimentação a Relê. (Dissertação de Mestrado) - Escola Federal de Engenharia de São Carlos, USP.
- CHIEN, K. L.; HRONES, J. A.; RESWICH, J.B. (1952). On the automatic control of generalized passive systems. Trans. ASME, 74, pp 175-185
- CONTROLSOFT (2007). Product data sheet: INTUNE. Disponível em: < <http://www.controlsoftinc.com> > . Acesso em: 20 dez. 2007.
- CUNHA, M. J. (2004). Desenvolvimento e implementação de uma ferramenta computacional aplicada no processo de identificação de sistemas em ambientes Fieldbus Foundation. (Dissertação de Mestrado)- Escola de Engenharia de São Carlos, USP.
- DESBOROUGH, L.; MILLER, R. (2001). Increasing Customer Value of Industrial Control Performance Monitoring—Honeywell’s Experience. In Chemical Process Control, vol. VI, Tucson, AR (pp. 153-185).
- EMERSON PROCESS MANAGEMENT (2003). Product data sheet: Foundation Fieldbus in the Real World. Disponível em: <<http://www.easydeltav.com/keytechnologies/fieldbus/seminar.asp> >. Acesso em 15 jul. 2007.
- EXPERTTUNE (2007). Product data sheet: Expertune. Disponível em: < <http://www.experttune.com/> >. Acesso em: 27 dez. 2007.
- FERNANDES, R. F. & BRANDÃO, D. (2008). Método de Identificação online de sistemas industriais com controladores PI em malha fechada, INDUSCON, VIII Conferência Internacional de Aplicações Industriais, Poços de Caldas, Agosto, 2008.
- FOUNDATION fieldbus (2003a). FOUNDATION Specification System Architecture FF-581, Revision FS1.3, October, 2003.

- FOUNDATION fieldbus (2003b). FOUNDATION Specification Function Block Application Process – Part 1- FF-890, Revision FS1.6, October, 2003.
- FOUNDATION fieldbus (2003c). FOUNDATION Specification Function Block Application Process – Part 2 - FF-891, Revision FS1.6, October, 2003.
- FRUEHAUF, P.S.; CHIEN, I.L.; LAURITSEN, M.D. (1994) – Simplified IMC-PID tuning rules. ISA Transactions 33 (1994) 43-59.
- GERRY, J., BUCKBEE G. (2005). The Link between Automation KPIs and Enterprise KPIs. Disponível em: < <http://www.expertune.com/articles> > Acesso em: 25 jul 2007.
- GROUP, O.M. (2007). Corba Specification. Disponível em: < <http://www.omg.org/gettingstarted/corbafaq.htm>.> Acesso em: 4/nov/2007.
- HAN, K. H.; KIM, S.; KIM, Y. J.; & KIM, J. H. (2001). Internet control architecture for Internet-based personal robot. Autonomous Robots, 10, 135–147.
- HANG, C.C.; ASTROM, K.J.; WANG, Q.G. (2002). Relay feedback auto-tuning of process controllers – a tutorial review. Journal of Process Control 12 (2002) 143-162.
- HARRIS, T.J.; SEPPALA,C.T.; DESBOROUGH, L.D. (1999). A review of performance monitoring and assessment techniques for univariate and multivariate control systems. Journal of Process Control 9 (1999) 1-17.
- HO, W.K., HANG, C.C., and CAO, L.S. (1995). Tuning of PID Controllers based on gain and phase margins specifications. In: Automatica, Vol 31, No 3, pp 497-502
- IEEE (2001). Information Technology - Telecommunications and Information exchange between systems -Local and Metropolitan Area Networks specific requirements -

- Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access and method Physical Layer Specifications. 2001. ISO/IEC 8802-3.
- ISERMANN, R. (1989). Digital Control Systems, 2ª edição, vol 1, Springer-Verlag Berlin Heidelberg.
- JANKE, M. (2000). OPC – Plug and Play Integration to Legacy Systems. In: IEEE Conference Record Of 2000 Annual Pulp And Paper Industry Technical Conference, 2000, Atlanta. Conference. Atlanta, p.68-72.
- Jenkings, M.A. e Traub, J.F. (1970). A Three-stage variables-sift iteration for polynomial zeros and its relation to generalized rayleigh iteration, Numer. Math. 14, 252-263.
- JSON (2007). Especificação JSON, Disponível em: <<http://json.org/>> Acesso : 4/nov/2007
- KAPSALIS, V.; KOUBIAS, S.; PAPADOPOULOS, G. (2002). OPC-SMS: a wireless gateway to OPC-based data sources. In: Computer Standards & Interfaces 24 (2002) 437-451.
- HANSEN, F. T. Y.; JANSEN, P. G. (2003). Real-time communication protocols: an overview. Technical Report TR-CTIT-03-49, Centre for Telematics and Information Technology, Univ. of Twente, The Netherlands, Oct. 2003.
- KO, C.C.; CHEN B.M.; CHEN, J.; ZHANG, J.; TAN, K.C. (2005). A web-based laboratory on Control of two degrees of freedom helicopter. In: International Journal of Engineering, Vol 21, n 23.
- KUNES, M.; e SAUTER, T. (2001). Fieldbus-Internet connectivity: The SNMP approach. IEEE Transactions on Industrial Electronics, 48(6), 1248–1256.
- KUROSE, J. F.; ROSS, K. W. (2006). Redes de computadores e a Internet. 3ª Edição, 2006, Editora Addison Wesley.

- LIPTAK, B. G. (1995). Instrument Engineers' Handbook – Process Control. 3rd Edition, 1995, Chilton Book Company.
- LJUNG, L. (1999). System Identification – Theory for the User. Segunda Edição, 1999, Englewood, Prentice Hall.
- LJUNG, L. (2008). System Identification Toolbox – User Guide, The MathWorks, Inc. Version 7.2.1, October 2008.
- LOESER , J. e HAERTIG, H. (2004). Low-latency hard real-time communication over switched Ethernet. In 16th Euromicro Conference on Real-Time Systems, Catania, Italy, 2004.
- LUO, R.C.; CHEN, T.M. (2000). Development of a multibehavior-based mobile robot for remote supervisory control through the Internet. IEEE/ASME Transactions on mechatronics, vol. 5, no. 4, December 2000.
- MICROSOFT (2007). DotNET Framework, Disponível em: <<http://www.microsoft.com/net>> Acesso em: 4/nov/2007
- MIRANDA, F.C.P.R (2005). Identificação de sistemas em malha fechada usando controlador preditivo multivariável: um caso industrial.(Dissertação de Mestrado) – Escola Politécnica, USP, São Paulo, 2005.
- MOSSIN, E.A. (2007) – Laboratório remoto para ensino a distância de sistemas de controle distribuído. (Dissertação de Mestrado) – Escola de Engenharia de São Carlos, USP.
- NEWCOMER, E.; LOMOW, G. (2004). Understanding SOA with Web Services, Editora Addison Wesley Professional, 2004.
- OGATA, K. (2003). Engenharia de Controle Moderno. Quarta Edição. Rio de Janeiro, 2003, Editora Prentice Hall do Brasil.

- OPC FOUNDATION (2006). Especificação OPC. Disponível em: <www.opcfoundation.org>. Acesso em: 26 jul. 2007
- OVERSTREET, J. W.; & TZES, A. (1999). Internet-based client/server virtual instrument designs for real-time remote-access control engineering laboratory. In Proceedings of the American control conference, Vol. 2, pp. 1472–1476.
- PANTONI, R.P., (2006). Desenvolvimento e implementação de uma descrição de dispositivos aberta e não-proprietária para equipamentos Foundation Fieldbus Baseada em XML.(Dissertação de Mestrado) - Escola de Engenharia de São Carlos, USP.
- PARKIN, R.M.; CZARNECKI, C.A.; SAFARIC, R.; CALKIN, D.W. (2002). A PID servo control system experiment conducted remotely via Internet. In: Pergamon Mechatronics 12 – pages 833-843.
- PATTLE, R.; RAMISCH, J. (1997). OPC the de facto standard for real time communication. Proceedings of the Joint Workshop on Parallel and Distributed Real-Time Systems, pp. 289-294
- PHAM, L., GEHLEN, G. (2005), Realization and Performance Analysis of a SOAP Server for MÓbile Devices, Proceedings of the 11th European Wireless Conference 2005, Vol. 2, Nicosia, Cyprus, April 2005, pp. 791-797
- QIN,W.; WANG, Q. (2007). An LPV approximation for admission control of an internet web server: Identification and control. Control Engineering Practice, Feb 2007.
- YU, Q.; CHEN, B.; CHENG, H. H. (2004). Web-Based Control System Design and Analysis. IEEE Control Systems Magazine, june 2004.

- RIVERA, D. E.; MORARI, M.; e SKOGESTAD, S. (1986). Internal Model Control, 4. PID Controller Design. *Industrial and Engineering Chemistry Process Design and Development*, vol 25 (1986), p. 252.
- RUEL, M. (2003). The conductor directs this orchestra –Instrument Society of America. *Intech*, November 2003, vol 50, no 11, pp. 20-22.
- SAYYAR-RODSARI, B. (2001). Nonlinear and hybrid modeling, simulation, and control: current technology and future views at pavilion technologies. *American Control Conference*, Arlington, VA, USA, 2001, Vol. 2, pp 1706-1707.
- SMAR EQUIPAMENTOS INDUSTRIAIS (2007). Function blocks instruction manual. Jul/07, Sertãozinho.
- TORRES, B. S. (2005). Sintonia de Malhas de Controle PID Através de *Software* Dedicado. 5º Congresso Internacional de Automação, Sistemas e Instrumentação – ISA Show 2005, São Paulo.
- TORRES, B. S.; HORI, E. S. (2005). Análise de desempenho de malhas de controle em indústria petroquímica. 4º Seminário nacional de controle e automação, Salvador/BA, Julho 2005.
- TORRISI, N. M.; FERRAZ, F. J.; VALENTE, C.M.; BRAGA, A.P.S.; OLIVEIRA, J.F.G. (2007a). An open communication system for remote control and monitoring dedicated to industrial applications over TCP-IP applied to public Networks. XVI Congresso Brasileiro de Automática, 2006, Salvador – BAHIA, Brasil.
- TORRISI, N. M.; BRANDÃO, D.; PANTONI, R.P.; OLIVEIRA, J.F.G. (2007b). Design of a communication system for integration of industrial networks over public IP networks. In: *Industrial Informatics, 2007 5th IEEE International Conference on* , vol.1, no., pp.201-206, 23-27 June 2007

- TORRISI, N. M. , (2007c). Sistemas de comunicação para redes IP dedicado a dados industriais codificados em JSON. Patente submetida ao Instituto Nacional da Propriedade Industrial (INPI), sn 018070047508, 2007.
- Van den Hof, P. M. J. (2004). System identification. Lecture notes. Delft University of Technology.
- VARADARAJAN, S., CHIUEH, T. (1999). Automatic fault detection and recovery in real time switched Ethernet networks. In Proceedings 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99), volume 1, pages 161-169, New York, USA, Mar. 1999.
- VENKATRAMANI, C., & CHIUEH, T. (1994). Supporting real-time traffic on Ethernet. In Proceedings 15th IEEE Real-Time Systems Symposium, pages 282-286, San Juan, Puerto Rico, Dec. 1994. Computer Society Press ISBN 0-8186-6600-5.
- VISIOLI, A.(2001). Optimal tuning of PID controllers for integral and unstable processes. Control Theory and Applications, IEE Proceedings - , vol.148, no.2, pp.180-184, Mar 2001
- WANG, Z.P., XIONG, G.Z., LUO, J., LAI, M.Z., ZHOU, W. (2000). A hard real-time communication control protocol based on the Ethernet. In Proceedings 7th Australian Conference on Parallel and Real-Time Systems, pages 161-170, Sydney, Australia, Nov. 2000. Springer-Verlag. ISBN 962-430-134-4.
- WANG, S.; XIE, J. (2002). Integrating Building management Systems and Facilities management on the Internet. Automation in Construction, February 2002.
- YANG, S.H.; CHEN, X.; ALTY, J.L. (2003). Design issues and implementation of internet-based process control systems. Control Engineering Practice (11) 2003, 709-720.

- YANG, S. H., DAI, C., KNOTT, R.P. (2007). Remote Maintenance of Control System performance over the Internet, *Control Engineering Practice*, Volume 15, Issue 5, May 2007, Pages 533-544
- YU, C. C. (2006). *Autotuning of PID Controllers : A Relay Feedback Approach*. 2nd Edition, Springer, 2006.
- ZEILMANN, R.; GOMES JR.; J.M., BAZANELLA; A., AND PEREIRA, C. E. (2003). Web-based Control Experiments on a Foundation Fieldbus Plant, 5th IFAC International Conference on Fieldbus Systems and their Applications, Aveiro, Portugal, July 2003, pp. 325-330.
- ZEILMANN, R. P. (2002). *Uma Estratégia para Controle e Supervisão de Processos Industriais via Internet*. 140p. (Dissertação Mestrado) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.
- ZIEGLER, J. G. AND N. B. NICHOLS (1942). Optimum settings for automatic controllers. *Trans. ASME*, 65, 433-444.
- ZHENG, L. E.; NAKAGAWA, H. (2002). OPC (OLE for process control) specification and its developments. *SICE 2002, Proceeding of the 41st SICE Annual Conference*, pp. 917-920, vol.2.
- ZHUANG, M.; ATHERTON, D.P. (1993). Automatic Tuning of Optimum PID controllers. In: *IEEE Proceeding-D*, 140, no 3, pp 216-224.

Apêndice

I. Exemplo de configuração do tele-sintonizador comunicando com FBSIMU via *CyberOPC*

Aqui será mostrado, em detalhes, os passos para configurar cada elemento da arquitetura proposta com o tele-sintonizador comunicando com *CyberOPC* sobre o FBSIMU. A figura 22 mostra a arquitetura utilizada nos testes.

Na configuração FBSIMU, para validação dos testes, foram utilizados os blocos: entrada analógica (AI), Controlador PID (PID), e saída analógica (AO). Também é necessário configurar o sistema simulado (Gs_Planta). Na figura 55 é mostrado o esquemático de interligação dos blocos funcionais.

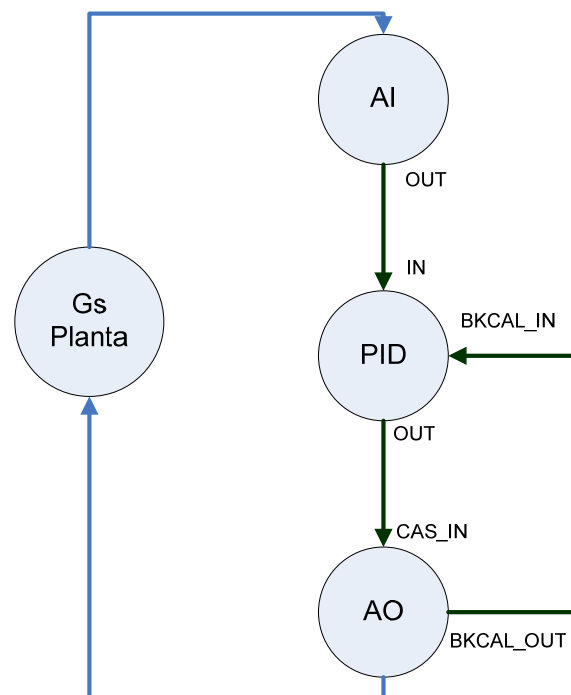


Figura 56 - Arquitetura do sistema de teste usando FBSIMU e tele-sintonizador remoto

A configuração dos parâmetros dos blocos é mostrada abaixo (somente são mostrados os parâmetros que possuem valor diferente de *default*).

AI.Mode_Blk = Auto *PID.Mode_Blk = Man* *AO.Mode_Blk = Auto*
AI.XD_Scale = 0;100 *PID.PV_Scale = 0;100* *AO.XD_Scale = 0;100*
AI.Out_Scale = 0;100 *PID.Out_Scale = 0;100* *AO.Out_Scale = 0;100*

O período de atualização dos blocos utilizado em todos os testes foi de 1000ms. Abaixo segue a tabela de escalonamento dos blocos utilizados:

Bloco	Offset
AI	0
AI.OUT → PID.IN	100
PID	200
PID.OUT → AO.CAS_IN	300
AO	400
AO.BKCALOUT → PID.BKCALIN	500
<i>Macrocycle_Total</i>	1000

Tabela 27 – Tabela de escalonamento da configuração fieldbus utilizada no FBSIMU

O simulador FBSIMU dispõe de um simulador de plantas com função de transferência de até quinta ordem onde foi possível fazer todos os testes. O salvamento do arquivo dos dados é feito automaticamente quando se inicia a simulação e os dados são salvos a uma taxa especificada, para os testes realizados foi de 1seg.

O próximo módulo a ser configurado é o servidor OPC responsável por fornecer todos os dados simulados para um cliente OPC. O módulo responsável pela aquisição de dados do FBSIMU e disponibilização para um cliente OPC é chamado DanSrv. Neste *software* são configurados somente os pontos necessários para a supervisão. A taxa de atualização é determinada pela taxa de publicação de dados do “*shared variable*” que é determinada no FBSIMU. A atualização é feita por evento a cada macro ciclo *fieldbus*. Na figura a seguir é mostrada a tela do configurador do DanSrv.

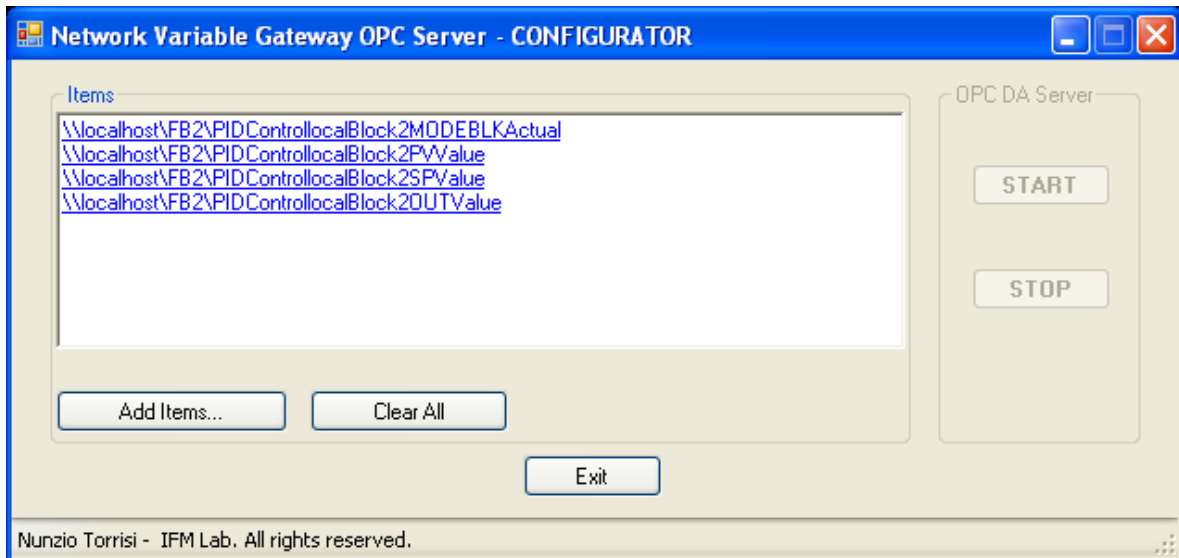


Figura 57 - Tela de configuração do servidor OPC do FBSIMU (DanSrv)

Após a configuração do módulo servidor OPC é possível ligar o *CyberOPC gateway*, que é responsável pela troca de dados através da rede. Porém, antes de ligar o *CyberOPC* é necessário ter um certificado de autenticação digital. Um exemplo deste certificado é mostrado na figura seguinte.

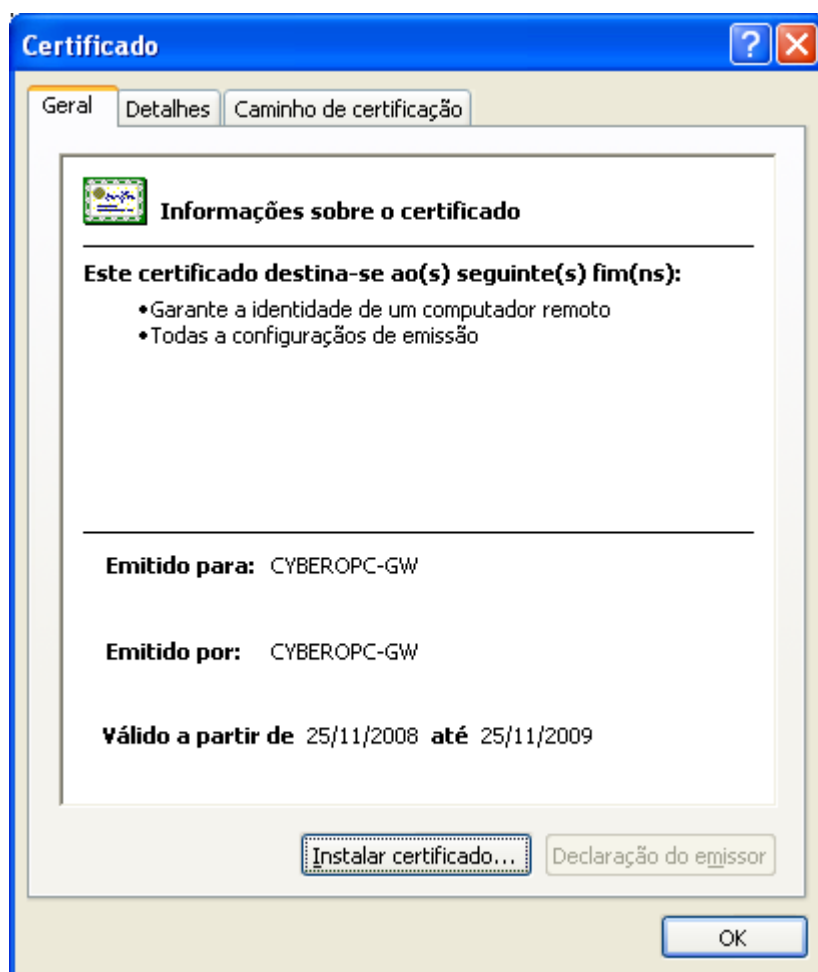


Figura 58 - Certificado de autenticação digital necessário para comunicação CyberOPC

Após instalar o *CyberOPC* é necessário configurar o *CyberOPC gateway* através de um arquivo de inicialização, onde será informado o arquivo de autenticação digital e o arquivo de configuração dos parâmetros OPC a serem monitorados. No arquivo de configuração também é informada a taxa de comunicação entre o *gateway* e o cliente *CyberOPC*. Para os nossos testes foram utilizados valores de 250 ms e 1000ms. A figura a seguir mostra a tela do *CyberOPC gateway*.

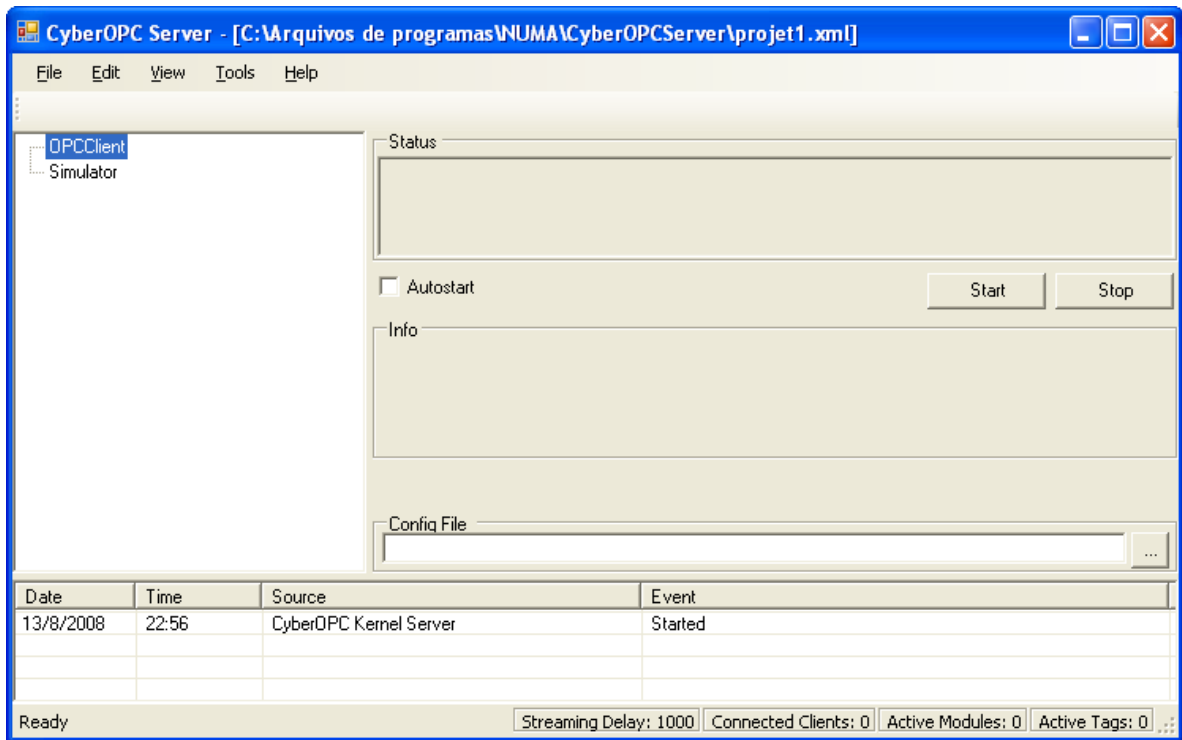


Figura 59 - Tela principal de configuração do CyberOPC gateway

Para o próximo passo é necessária a configuração do tele-sintonizador comunicando sobre *CyberOPC*. Neste caso, é feito de forma remota em outra máquina. Primeiramente, é necessário também instalar o certificado, conforme figura 57 na máquina cliente. Então, é necessário configurar o IP e nome do certificado no próprio tele-sintonizador. Na figura a seguir é mostrado um exemplo de configuração.

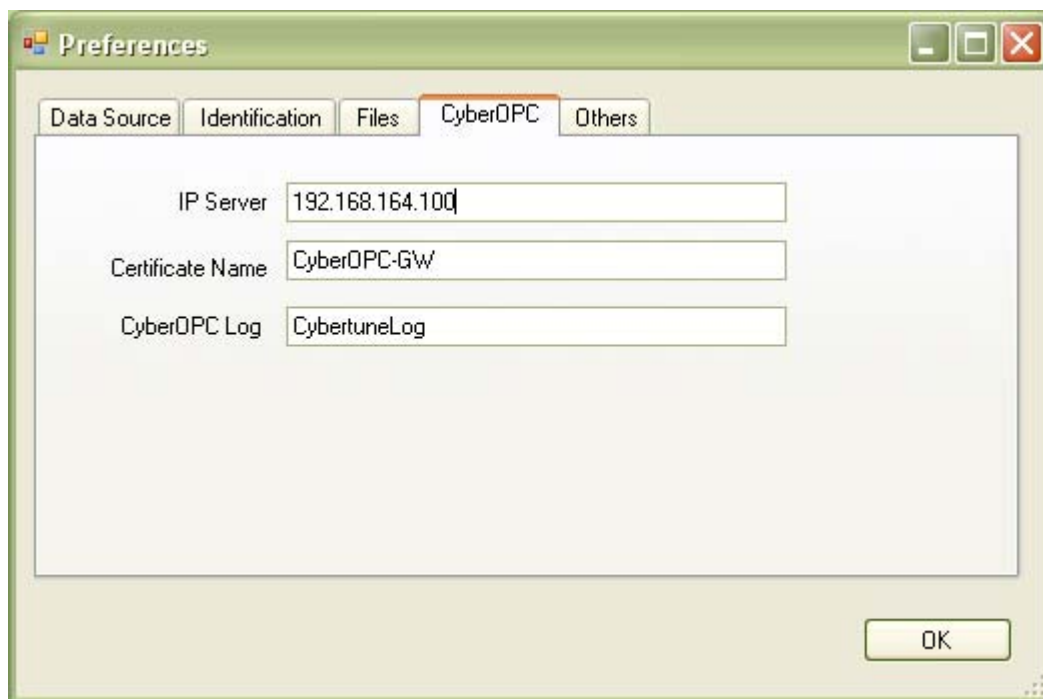


Figura 60 - Tela de configuração do CyberOPC no tele-sintonizador

Neste exemplo foi configurado o tele-sintonizador a acessar o *CyberOPC gateway* no IP 192.168.164.100 e requer um certificado com o nome de CyberOPC-GW. Este certificado é mostrado na figura 57.

Por fim, é feita a conexão no tele-sintonizador e a comunicação se inicializa. A seguir é mostrado um exemplo de identificação online feita pelo tele-sintonizador.

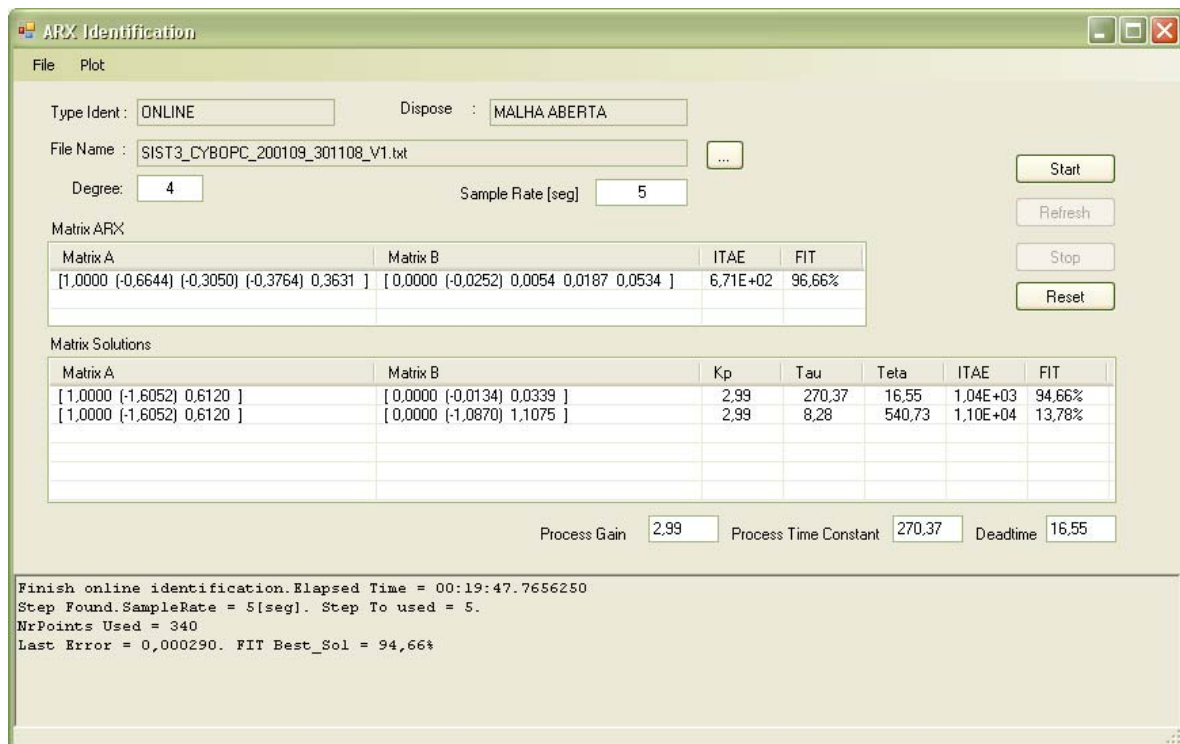


Figura 61 - Tela do tele-sintonizador comunicando com o CyberOPC

II. Exemplo de comunicação do tele-sintonizador comunicando com o DF51

Nesta seção serão apresentados os detalhes da configuração do ambiente de teste com o controlador DF51 da empresa SMAR Equipamentos Industriais. As simulações com o controlador DF51 possibilitaram a validação da comunicação local e remota com um equipamento comercial.

Neste cenário, a simulação da planta e do ambiente *fieldbus* é feita no controlador DF51 e o sintonizador pode comunicar tanto na estação local quanto na estação remota, conforme mostrado na figura 23. No cenário local o tele-sintonizador comunica OPC com o servidor OPC do DF51 (DF51OPC Server). No cenário remoto, o *CyberOPC gateway* da estação local comunica OPC com o servidor OPC do DF51 e o tele-sintonizador da estação remota comunica *CyberOPC* com o *CyberOPC gateway*.

Para a elaboração da estratégia de simulação e controle foram necessários os blocos *fieldbus*: controlador PID (PID), saída analógica (AO), aritmético (ARTH) e função de

transferência (TF). O *macrocycle fieldbus* foi de 1000ms. Na figura abaixo é mostrada a estratégia de controle utilizada.

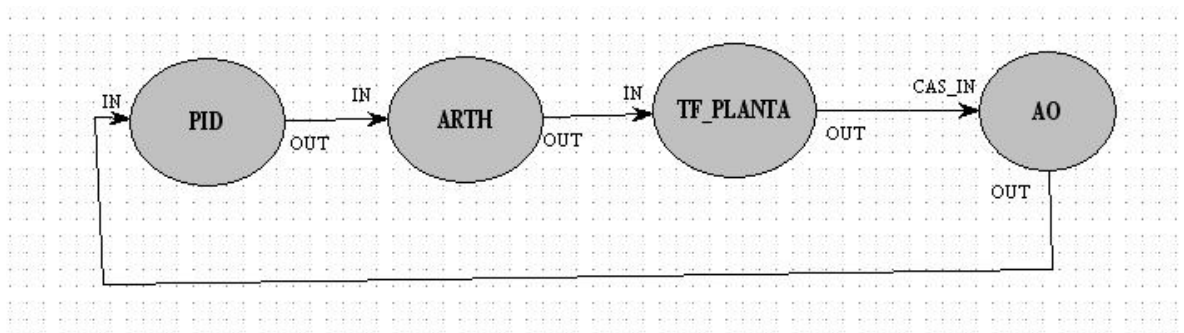


Figura 62 - Estratégia de simulação de planta industrial fieldbus utilizado DF51

A aquisição do OPC também foi configurada com taxa de aquisição de 1000ms. O bloco função de transferência (TF) consegue gerar uma função de transferência no domínio s até 3ª ordem e pode ser cascadeado para formar uma função de transferência de ordem maior. Na figura a seguir é mostrado um exemplo de configuração de uma função de transferência no bloco TF. O bloco ARTH é responsável por gerar a variação de carga no sistema em malha fechada. A configuração dos parâmetros dos blocos é mostrada na tabela abaixo.

Bloco	Parâmetro	Valor
PID	MODE_BLK	Man - para malha aberta Auto – para malha fechada
	PV_SCALE	0 ; 100
	OUT_SCALE	0 ; 100
ARTH	MODE_BLK	Auto
	ARTH_TYPE	Soma Tradicional
	IN_1	1 - quando variação de carga 0 - para regime normal
TF_PLANTA	MODE_BLK	Auto
	A,B,C,D,E,F	A função de transferência do sistema simulado (considerando o tempo morto)
AO	MODE_BLK	Càs
	PV_SCALE	0;100%
	XD_SCALE	0;100%

Tabela 28 – Configuração dos parâmetros dos blocos FF para os testes de simulação da planta fieldbus

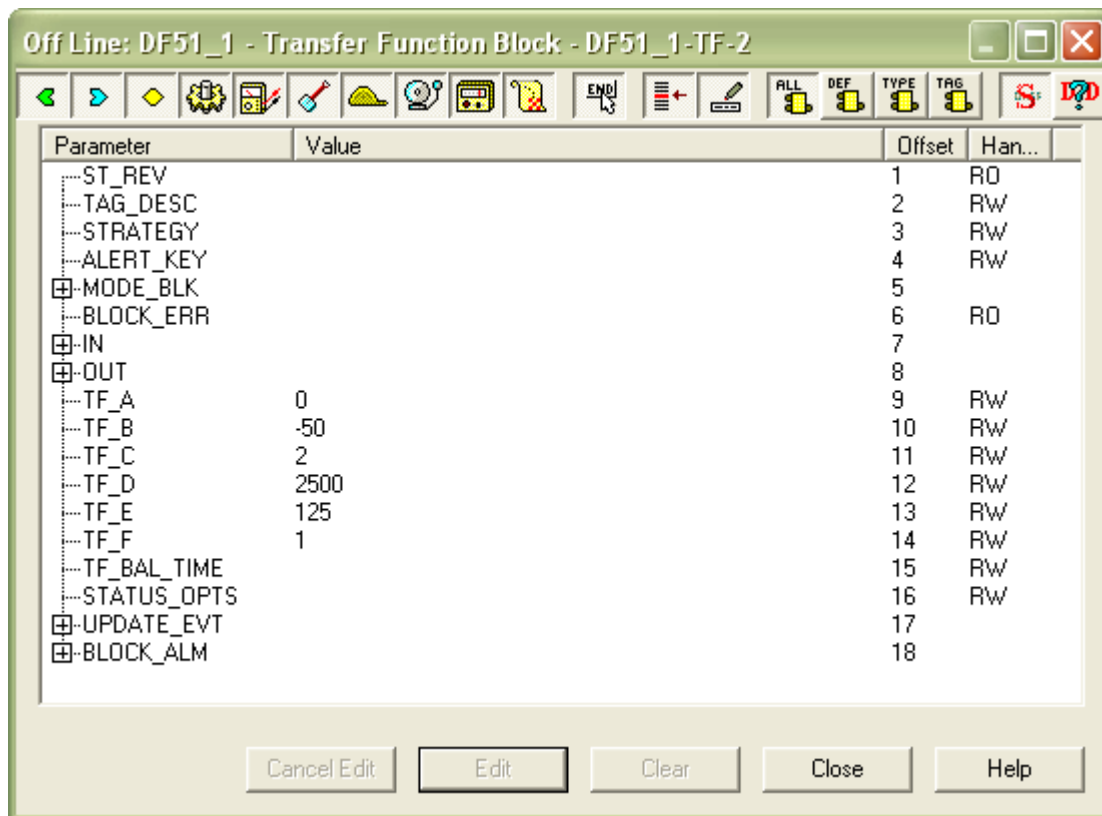


Figura 63 - Arquitetura do sistema de teste usando DF51 e tele-sintonizador local

Para configuração do *CyberOPC gateway* é feito da mesma forma mostrada no Apêndice I, porém é necessário trocar o servidor OPC do FBSIMU pelo do DF51OPC, nos respectivos nomes dos parâmetros. Para a configuração do tele-sintonizador também somente é necessário configurar o correto OPC e os nomes dos parâmetros.

III. Exemplo da identificação e sintonia utilizando o tele-sintonizador

Nesta seção será apresentado um exemplo de identificação e sintonia utilizando o tele-sintonizador. Aqui já é pressuposto que o usuário já configurou a comunicação com os equipamentos como mostrado nos itens I e II do apêndice.

Inicialmente o usuário deve preencher a configuração do equipamento a ser monitorado e do bloco PID Foundation Fieldbus que será sintonizado, configurando o endereço do servidor, tipo de OPC Server, tags OPC do bloco PID. Esta configuração é executada na opção “preference” do tele-sintonizador mostrado na figura 60.

Então o usuário inicia a comunicação como o equipamento através do botão “connect”. Após conectado ao equipamento a tela principal do tele-sintonizador mostra os dados online sendo capturados como mostrado na figura 63. No rodapé da tela é mostrado um log da conexão.

A tela principal possui dois grupos de informações do controlador e do processo. O grupo “*Controller*” mostra os dados dinâmicos do PID: Setpoint (SP), variável de Processo (PV) e saída (OUT). Estas informações dinâmicas também são mostradas no gráfico da tela principal “*Online Graph*”. Também são mostradas as informações estáticas do bloco PID: Modo de Operação (*Mode*), K_c (*Gain*), T_i (*Reset*), T_d (*Rate*). O modo de operação indica se a malha esta em automático ou manual. O grupo “*Test Process*” mostra as informações do sistema de primeira ordem identificado.

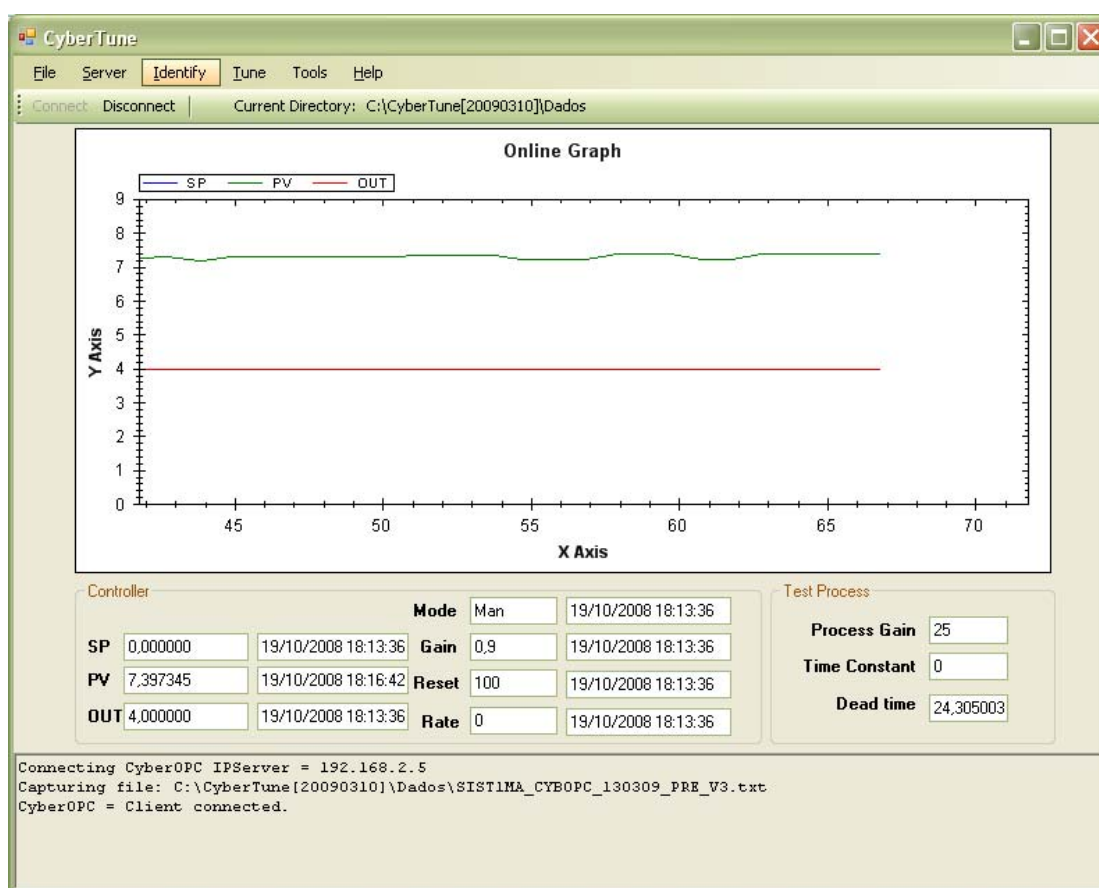


Figura 64 – Tela principal do tele-sintonizador online capturando dados do PID

Na janela principal, clicando no botão “*Identify*”, o programa entra no módulo de identificação mostrado na figura 65. Nesta fase o usuário pode iniciar uma identificação online ou offline, escolher a ordem do modelo ARX e a taxa de amostragem. Pressionando o botão “*Start*” inicia-se a identificação e somente terminará quando o sistema convergir ou

atingir um número máximo de iterações. Ao final da identificação é mostrado na seção “*Matrix ARX*” o modelo ARX obtido, e na seção “*Matrix Solutions*” as soluções candidatas. Para cada uma das seções são mostradas as matrizes A e B da forma de (8) e também o correspondente ITAE e FIT calculado. Por fim é mostrado o sistema mais adequado ao modelo. Para realizar a validação do modelo final obtido é necessário clicar no botão “*Verify*” e iniciar uma nova coleta de dados.

No rodapé da janela é mostrado um *log* com informações sobre a identificação como o tempo de identificação, número de pontos utilizados e o último erro calculado.

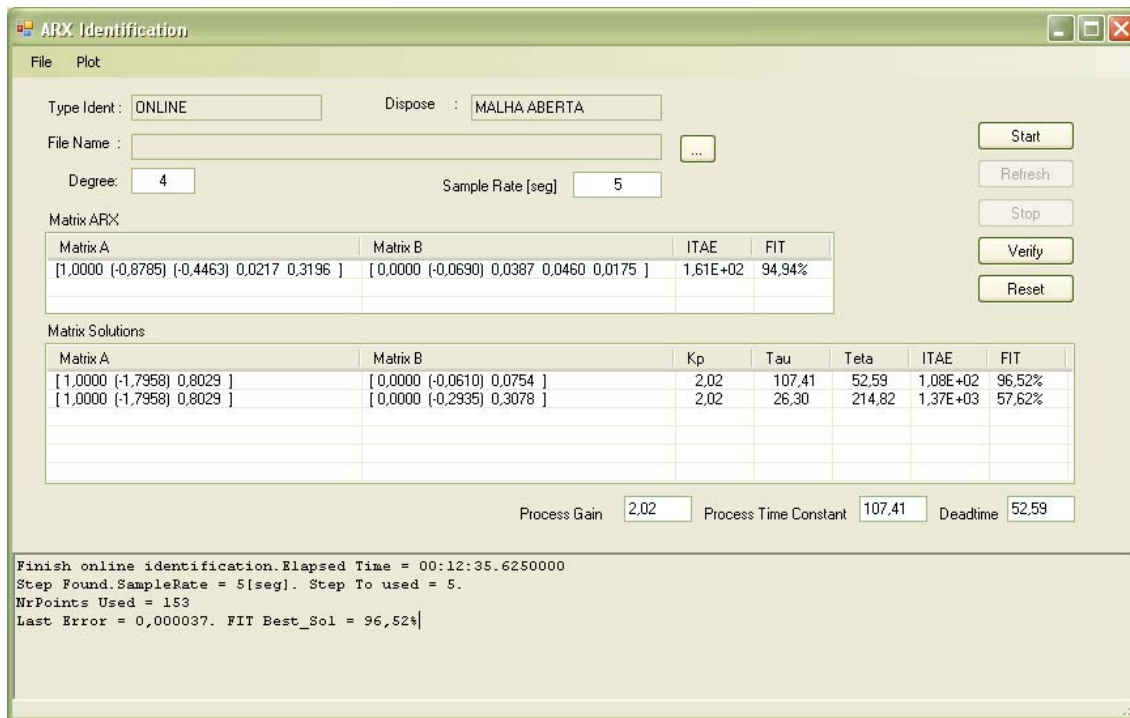


Figura 65 – Tela de identificação do tele-sintonizador

Após o final da identificação é possível mostrar os gráficos de resposta a degrau do sistema identificado ARX e o modelo malha aberta final identificada. Também é possível mostrar o gráfico do erro entre os modelos e a entrada real. Exemplos destes dois gráficos são mostrados nas figuras 66 e 67.

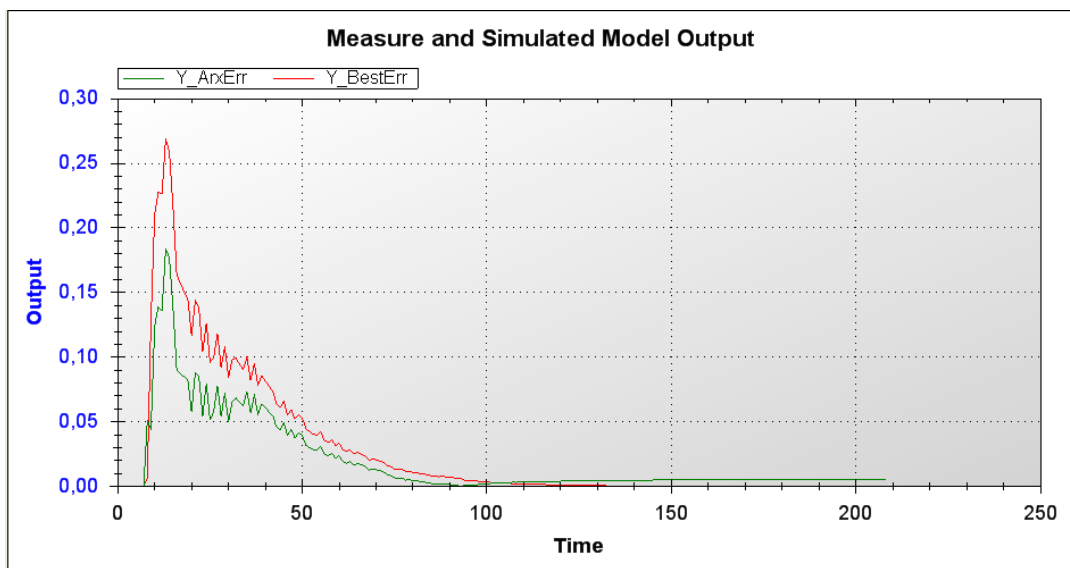


Figura 66 – Gráfico do erro de identificação comparado com a aquisição

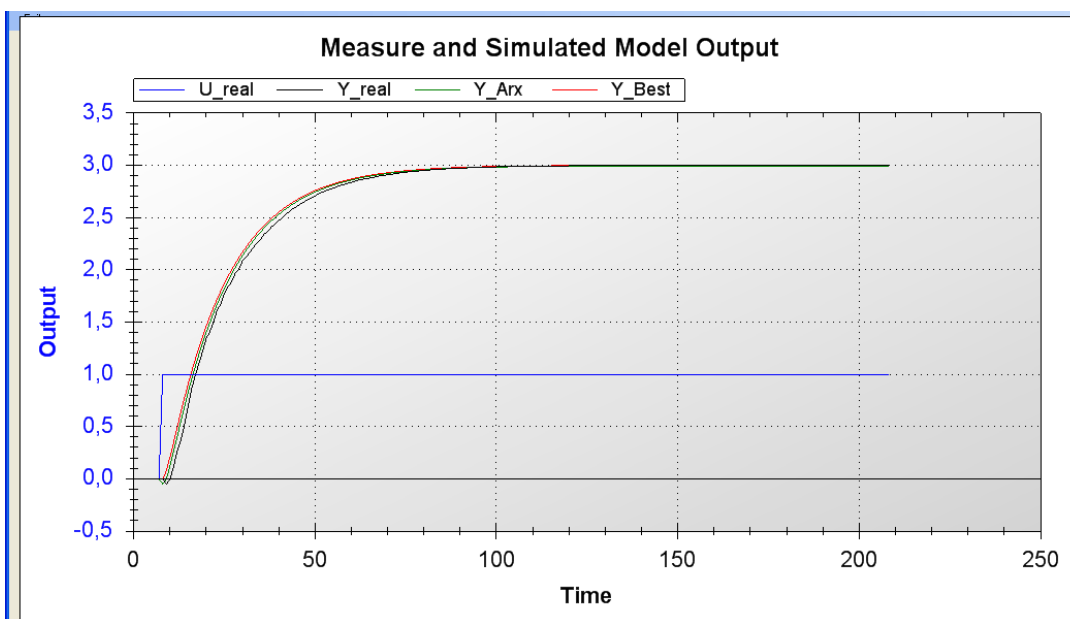


Figura 67 – Gráfico de resposta a degrau dos sistemas Arx e malha aberta

Após a obtenção do modelo FOPDT é possível realizar a sintonia baseada em um dos métodos baseados em modelo. Para entrar no módulo de sintonia é necessário clicar no botão “Tune” na tela principal. No módulo de sintonia é possível escolher entre os vários métodos disponíveis e então é calculado e mostrado o gráfico de todos os métodos escolhidos. As figuras 68 e 69 mostram um exemplo de sintonia.

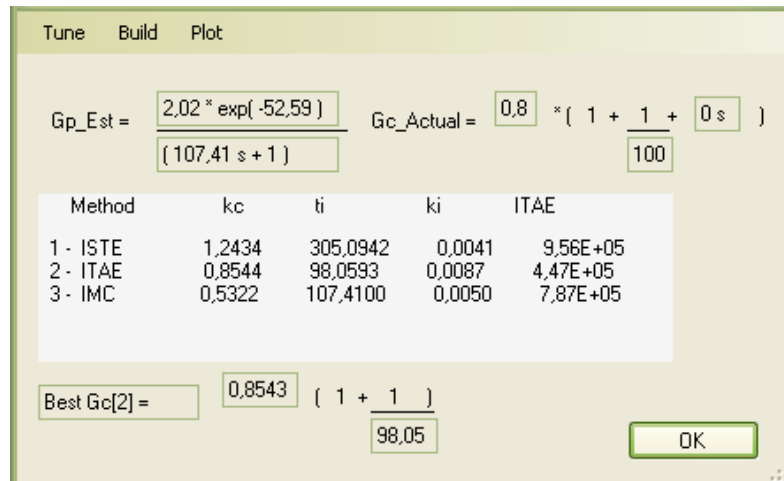


Figura 68 – Parâmetros do PID para os métodos de sintonia escolhido

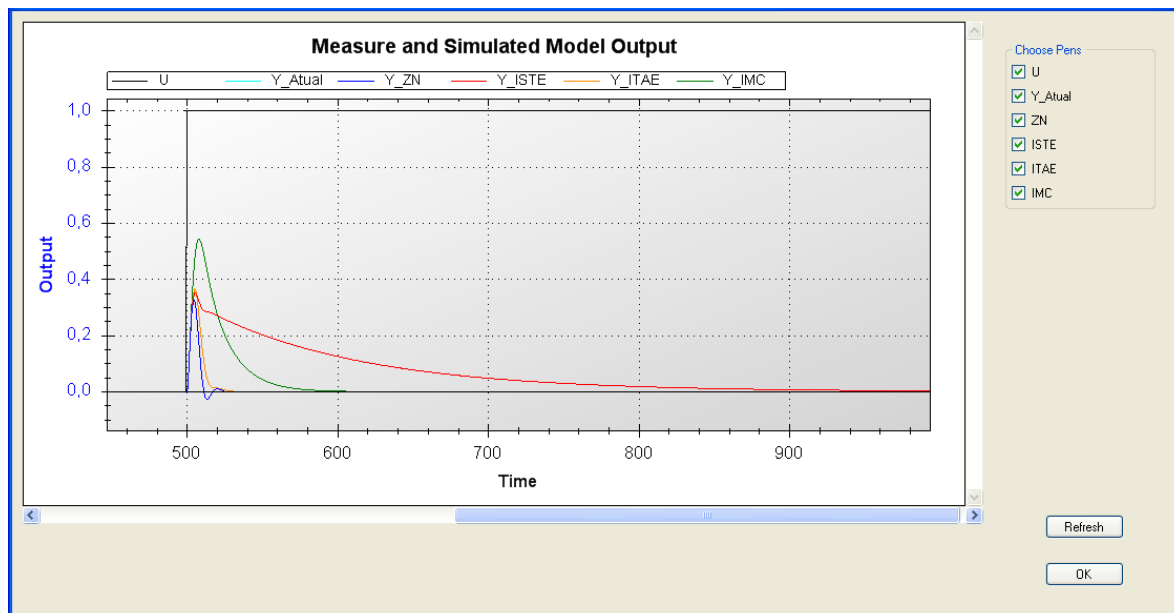


Figura 69 – Gráfico de simulação de variação de carga do sistema atual e os métodos escolhidos

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)