

Brasília, 6 de setembro de 2007.

RELATÓRIO DE ATIVIDADES DE PROJETO DE PESQUISA
agosto/2006 - setembro/2007

Edgar Jhonny Amaya Simeón

FINATEC
ELETRONORTE
Universidade de Brasília – UnB

1. Objetivo

Apresentar o trabalho feito no primeiro ano do projeto de pesquisa que é parte integrante do projeto de modernização da área de automação de processos da usina hidrelétrica de Balbina, desenvolvido pelo programa de pesquisa e desenvolvimento tecnológico da empresa de geração e transmissão de energia elétrica, ELETRONORTE, em parceria com a FINATEC e UnB.

2. Atividades realizadas

Podem ser listadas as seguintes atividades específicas:

- ✓ Estudo do diagrama organizacional de hierarquia dos equipamentos, subsistemas e componentes, indicando as fronteiras e interfaces entre os subsistemas.
- ✓ Análise do FMEA - Análise de Modos e Efeitos de Falhas - associado ao sistema da turbina das unidades geradoras hidráulicas de Balbina.
- ✓ Análise da Árvore de Falhas relacionando falhas primárias à falha final do sistema (corte de carga).
- ✓ Análise do sistema por diagrama de blocos, com o cálculo de medidas de confiabilidade, disponibilidade, MTBF, taxa de falhas, número esperado de falhas, freqüências e taxas de risco.
- ✓ Estudo das ferramentas de aquisição de dados para implementação do sistema inteligente via OPC usando o software Matlab 7.0
- ✓ Visitas à ELETRONORTE para coleta de dados e aquisição de material referente ao funcionamento da Usina de Balbina.
- ✓ Implementacion de Controle avançado tais como Redes Neurais na planta didática 3 da Smar nos dispositivos Fieldbus usando o Servidor OPC.

- ✓ Utilização dos softwares disponíveis com a planta tais como Logic View, Syscon, Process View, Asset View.
- ✓ Estudo e elaboração dos casos de uso UML usando o software JUDE para o sistema I-Kernel pro SIMPREBAL (Sistema de Manutenção Preditiva de Balbina).
- ✓ Elaboração e submissão do Paper ao CAIP2007 o congresso de computação aplicada 2007
- ✓ Visita Técnica à Usina Hidreletrica de Balbina no periodo 08/01/2007 ao 09/02/2007, Localizado no Município de Presidente Figueredo no Estado de Amazonas a 107 km da cidade de Manaus. O lago de Balbina proporcionado pela construção da usina é localizado a 180 km da cidade de Manaus, no Rio Uatumã, um dos afluentes do Rio Amazonas.
- ✓ Estudo da Instrumentação Fieldbus Foundation Instalada na Usina de Balbina o acesso aos valores disponibilizados pelos transmissores no servidor OPC através de uma conexão cliente servidor usando Matlab 7 e JOPCClient em ambiente Java. Registro de todas as TAGs dos instrumentos instalador e por instalar os valores normais de operação. Elaboração de um sinotico usando o GUI de Matlab para a instrumentação de Temperatura da Usina Hidreletrica de Balbina.
- ✓ Elaboração de Abstracts submetidos ao COBEM 2007 - 19th International Congress of Mechanical Engineering
- ✓ Estudo das ferramentas de aquisição de dados para implementação do sistema inteligente via OPC usando Microsoft Visual Studio Net e aplicativos em Java.
- ✓ Elaboração e submissão dos Papers ao COBEM 2007 - 19th International Congress of Mechanical Engineering, "Different Control Strategies used in Didactic Plant PD-3 Of Smar Through OPC Technology" e "An Intelligent Kernel for the Maintenance System of a Hydroelectric Power Plant".
- ✓ Desenvolvimento da aplicação em java para leitura de tags de um arquivo tags.txt envio dessas tags ao OPC sever e os valores das tags retornadas salva-los em um arquivo dados.txt.
- ✓ Integração dos dados obtido através do OPC server com o Ikernel V0.1 em desenvolvimento, onde os dados de configuração hostname, PrgID e as tags são lidas do arquivo config.ini, os valores das tags obtidas serão processadas por a maquina de inferença de JESS, Fuzzy-JESS, Redes neiras.
- ✓ Papers submetidos e aceitos no COBEM 2007 - 19th International Congress of Mechanical Engineering, "Different Control Strategies used in Didactic Plant PD-3 Of Smar Through OPC Technology" e "An Intelligent Kernel for the Maintenance System of a Hydroelectric Power Plant".
- ✓ Paper Submetido e aceito no CIBIM8 – 8º Congresso Iberoamericano de Engenharia Mecânica – Cusco - Peru, "Sistema Inteligente de Manutenção Baseada em Condição para Usina Hidrelétrica de Balbina".

- ✓ Integração dos dados obtido através do servidor OPC com o Ikernel V3.0 em desenvolvimento, onde os dados de configuração host, progid e as tags são lidas do arquivo config.ini, os valores das tags obtidas são processadas por a maquina de inferência de JESS, e no futuro por o Fuzzy-JESS, Redes Neurais.
- ✓ Integração dos dados obtido através do servidor DB(Banco de Dados) com o Ikernel V3.0 em desenvolvimento, onde os dados de configuração e as tags são lidas do arquivo config.ini, os valores das tags obtidas são processadas por a maquina de inferência de JESS, e no futuro por o Fuzzy-JESS, Redes Neurais.

3. Comunicação Via OPC do Matlab com a Planta Didatica 3 da Smar

A continuação apresento o trabalho feito em entorno Matlab usando comunicação via Servidor OPC quem disponibiliza os dados da planta didatica da smar e fazendo o tratamento das sinais obtidas atravez das TAGs do processo, analize de tendencias é o controle por redes neurais que apois deste trabalho vai dar uma maior visão para elaboraçãõ do sistema SIMPREBAL para Manutenção preditiva da Usina Hidreletrica de Balbina (Fig. 1).

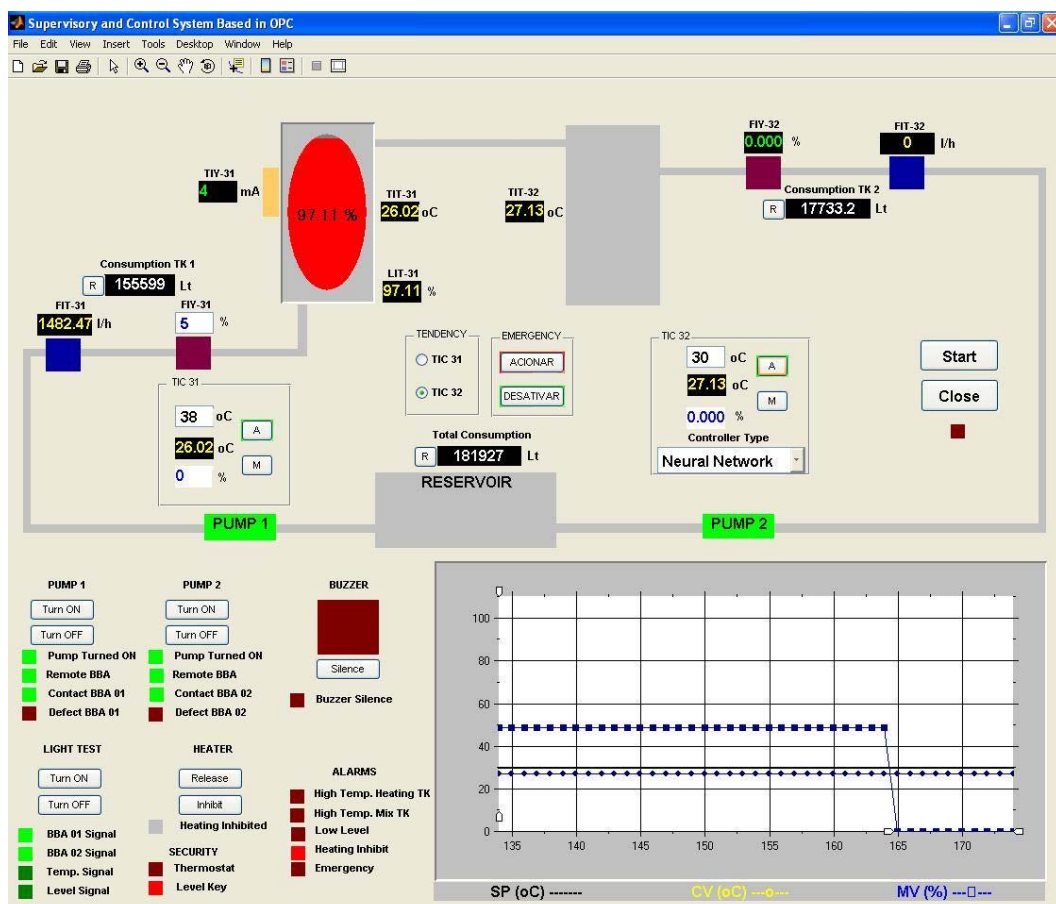


Fig. 1 – supervisorio em Matlab da Planta didatica PD3 da Sma

4. Cadastro da Instrumentação Fieldbus Foundation da Usina Hidreletrica de Balbina

UGH01 Unidade Geradora Hidráulica

GEP01 GERADOR ELÉTRICO PRINCIPAL

Equipamentos

Rotor

Anéis Coletores

Cubículo Terminal de Neutro

TC's Fases A, B e V Saída do Gerador Barr

Caixa Geral de Terminais

Cubículo Terminal de Fases e Surto

Estator

Sistema de Resfriamento

Acopladores/Sensores - Med. PDA Sistem

Medidor de Descargas Parciais 01

Medidor de Descarga Parcial - PDA Sistem

TAG	Descrição	PV	ALM	TRIP	FAIXA
G149G1A	Temperatura do enrolamento do estator fase A	85	130	155	0-200 °C
G149G1B	Temperatura do enrolamento do estator fase B	85	130	155	0-200 °C
G149G1V	Temperatura do enrolamento do estator fase V	85	130	155	0-200 °C
G149G2A	Temperatura do núcleo do estator fase A	80	não	não	0-200 °C
G149G2B	Temperatura do núcleo do estator fase B	80	não	não	0-200 °C
G149G2V	Temperatura do núcleo do estator fase V	80	não	não	0-200 °C

SRG01 Sistema de Resfriamento do Gerador

Equipamentos

Radiador 01, Radiador 02, Radiador 03, Radiador 04, Radiador 05, Radiador 06,

Radiador 07 e Radiador 08

TAG	Descrição	PV	ALM	TRIP	FAIXA.
G126GAF1	Temperatura de ar frio do radiador n° 1	44	45	não	20-60 °C
G126GAF2	Temperatura de ar frio do radiador n° 2	44	45	não	20-60 °C
G126GAF3	Temperatura de ar frio do radiador n° 3	44	45	não	20-60 °C
G126GAF4	Temperatura de ar frio do radiador n° 4	44	45	não	20-60 °C
G126GAF5	Temperatura de ar frio do radiador n° 5	44	45	não	20-60 °C
G126GAF6	Temperatura de ar frio do radiador n° 6	44	45	não	20-60 °C
G126GAF7	Temperatura de ar frio do radiador n° 7	44	45	não	20-60 °C
G126GAF8	Temperatura de ar frio do radiador n° 8	44	45	não	20-60 °C
G126GAQ1	Temperatura de ar quente	65	75	85	20-100°C

BRB01 Barramento Blindado

ALM / HI	21	ALM.
CONDIÇÃO 2	20	desliga o compressor do sistema de pressurização, através da SD G163PBBA.
CONDIÇÃO 1	5	liga o compressor do sistema de pressurização, através da SD G163PBBA.
ALM / LO	4	ALM.

TAG	Descrição	PV	ALM	TRIP	FAIXA
G163PBB	Pressão do sistema de pressurização do barramento blindado	N	20	não	0-50 mbar

CTN01 Cubículo Terminal de Neutro

CTF01 Cubículo Terminal de Fase

SRT01 - Sistema de Regulação de Tensão

Equipamentos

Pontes Retificadoras (Tiristores)
Regulador Manual
Regulador Automático
Motoventiladores de Refrigeração
Cartelas Eletrônicas
Trafo de Excitação
Contator de Campo
Transdutores Para medição

TAG	Descrição	PV	ALM	TRIP	FAIXA
G149TEA1	Temperatura do trafo de excitação fase A enrolamento nº 1	96	110	130	0-200 °C
G149TEA2	Temperatura do trafo de excitação fase A enrolamento nº 2	96	110	130	0-200 °C
G149TEB1	Temperatura do trafo de excitação fase B enrolamento nº 1	93	110	130	0-200 °C
G149TEB2	Temperatura do trafo de excitação fase B enrolamento nº 2	93	110	130	0-200 °C
G149TEV1	Temperatura do trafo de excitação fase V enrolamento nº 1	99	110	130	0-200 °C
G149TEV2	Temperatura do trafo de excitação fase V enrolamento nº 2	99	110	130	0-200 °C

SMN01 SISTEMA DE MANCAL

MGT01 Mancal Guia da Turbina (Mancal Guia Inferior)

Equipamentos

Patins (Sapata)
Cuba de Óleo
Sensores de Temperatura (Sondas)

TAG	Descrição	PV	ALM	TRIP	FAIXA
G138MK1	Temperatura do metal do mancal guia inferior nº 1	54	75	85	0-200 °C
G138MK2	Temperatura do metal do mancal guia inferior nº 2	54	75	85	0-200 °C
G138MK3	Temperatura do metal do mancal guia inferior nº 3	54	75	85	0-200 °C

G138MJ1	Temperatura do óleo do mancal guia inferior nº 1	53	70	não	0-200 °C
G138MJ2	Temperatura do óleo do mancal guia inferior nº 2	53	70	não	0-200 °C
G138MJ3	Temperatura do óleo do mancal guia inferior nº 3	53	70	não	0-200 °C
G163MS	Pressão de óleo na cuba	N	0.25	0.06	0-10 bar

Itens de PV

ItmPv138MK1 = additem(grp_PvGdr1,'138MK1_AI1.OUT.VALUE','single');
 ItmPv138MK2 = additem(grp_PvGdr1,'138MK2_AI1.OUT.VALUE','single');
 ItmPv138MK3 = additem(grp_PvGdr1,'138MK3_AI2.OUT.VALUE','single');
 ItmPv138MJ1 = additem(grp_PvGdr1,'138MJ1_AI2.OUT.VALUE','single');
 ItmPv138MJ2 = additem(grp_PvGdr1,'138MJ2_AI1.OUT.VALUE','single');
 ItmPv138MJ3 = additem(grp_PvGdr1,'138MJ3_ALM.OUT.VALUE','single');

Itens de Alarme

ItmAlm138MK1 = additem(grp_AlmGdr1,'138MK1_ALM.HI_LIM.VALUE','single');
 ItmAlm138MK2 = additem(grp_AlmGdr1,'138MK2_ALM.HI_LIM.VALUE','single');
 ItmAlm138MK3 = additem(grp_AlmGdr1,'138MK3_ALM.HI_LIM.VALUE','single');
 ItmAlm138MJ1 = additem(grp_AlmGdr1,'138MJ1_ALM.HI_LIM.VALUE','single');
 ItmAlm138MJ2 = additem(grp_AlmGdr1,'138MJ2_ALM.HI_LIM.VALUE','single');
 ItmAlm138MJ3 = additem(grp_AlmGdr1,'138MJ3_ALM.HI_LIM.VALUE','single');

Itens de Trip

ItmTrip138MK1 = additem(grp_TripGdr1,'138MK1_TRIP.HI_HI_LIM.VALUE','single');
 ItmTrip138MK2 = additem(grp_TripGdr1,'138MK2_TRIP.HI_HI_LIM.VALUE','single');
 ItmTrip138MK3 = additem(grp_TripGdr1,'138MK3_TRIP.HI_HI_LIM.VALUE','single');

MGG01Mancal Guia do Gerador (Mancal Guia Superior)

Equipamentos

Patins (Sapata)

Cuba de Óleo

Sensores de Temperatura (Sondas)

condição para G171GMO

Condição	>= 20	intertravamento na pré-condição para partida da máquina.
ALM / HI	260	gera alarme no campo, através da SD G171GMOAA.

TAG	Descrição	PV	ALM	TRIP	FAIXA
G171GMO	Nível de óleo da cuba	N	20	10	0-300 mmH2O
G138GMM1	Temperatura do metal do mancal guia superior nº 1	59	80	85	0-200 °C
G138GMM2	Temperatura do metal do mancal guia superior nº 2	59	80	85	0-200 °C
G138GMM3	Temperatura do metal do mancal guia superior nº 3	59	80	85	0-200 °C
G138GMO1	Temperatura do óleo do mancal guia superior nº 1	55	70	75	0-200 °C
G138GMO2	Temperatura do óleo do mancal guia superior nº 2	55	70	75	0-200 °C
G138GMO3	Temperatura do óleo do mancal guia superior nº 3	55	70	75	0-200 °C

Itens de PV

ItmPv138GMM1 = additem(grp_PvGdr1,'138GMM1_AI1.OUT.VALUE','single');
 ItmPv138GMM2 = additem(grp_PvGdr1,'138GMM2_AI2.OUT.VALUE','single');
 ItmPv138GMM3 = additem(grp_PvGdr1,'138GMM3_AI1.OUT.VALUE','single');
 ItmPv138GMO1 = additem(grp_PvGdr1,'138GMO1_AI1.OUT.VALUE','single');
 ItmPv138GMO2 = additem(grp_PvGdr1,'138GMO2_AI2.OUT.VALUE','single');
 ItmPv138GMO3 = additem(grp_PvGdr1,'138GMO3_ALM.OUT.VALUE','single');

Itens de Alarme

ItmAlm138GMM1 = additem(grp_AlmGdr1,'138GMM1_ALM.HI_LIM.VALUE','single');
 ItmAlm138GMM2 = additem(grp_AlmGdr1,'138GMM2_ALM.HI_LIM.VALUE','single');
 ItmAlm138GMM3 = additem(grp_AlmGdr1,'138GMM3_ALM.HI_LIM.VALUE','single');
 ItmAlm138GMO1 = additem(grp_AlmGdr1,'138GMO1_ALM.HI_LIM.VALUE','single');
 ItmAlm138GMO2 = additem(grp_AlmGdr1,'138GMO2_ALM.HI_LIM.VALUE','single');
 ItmAlm138GMO3 = additem(grp_AlmGdr1,'138GMO3_ALM.HI_LIM.VALUE','single');

Itens de Trip

ItmTrip138GMM1 = additem(grp_TripGdr1,'138GMM1_TRIP.HI_HI_LIM.VALUE','single');
 ItmTrip138GMM2 = additem(grp_TripGdr1,'138GMM2_TRIP.HI_HI_LIM.VALUE','single');
 ItmTrip138GMM3 = additem(grp_TripGdr1,'138GMM3_TRIP.HI_HI_LIM.VALUE','single');
 ItmTrip138GMO1 = additem(grp_TripGdr1,'138GMO1_TRIP.HI_HI_LIM.VALUE','single');
 ItmTrip138GMO2 = additem(grp_TripGdr1,'138GMO2_TRIP.HI_HI_LIM.VALUE','single');
 ItmTrip138GMO3 = additem(grp_TripGdr1,'138GMO3_TRIP.HI_HI_LIM.VALUE','single');

MCB01 Mancal Combinado (Guia/Escora)**Equipamentos**

Patins
 Sapatras do Mancal Guia
 Sapatras do Mancal Escora
 Cuba de Óleo
 Sensores de Temperatura (Sondas)

Se G163CS >60
 intertravamento no CKT de partida da máquina

condição para G171MC

TRIP/LOLO	15	gera alarme no campo e TRIP, através da SD G171MDA.
ALM / LO	30	intertravamento na pré-condição da partida da máquina por nível baixo de óleo na cuba.
Condição	>30	intertravamento na pré-condição para partida da máquina.
ALM / HI	280	gera alarme no campo, através da SD G171MCA.

TAG	Descrição	PV	ALM	TRIP	FAIXA
G163CS	Pressão de óleo de injeção do mancal escora	N	60	não	0-200 bar
G171MC	Nível de óleo na cuba	N	30	15	0-400 mmH2O
G138MG1	Temperatura do metal do mancal guia intermediário nº 1	51	75	85	0-200 °C
G138MG2	Temperatura do metal do mancal guia intermediário nº 2	51	75	85	0-200 °C
G138MG3	Temperatura do metal do mancal guia intermediário nº 3	51	75	85	0-200 °C
G138ME1	Temperatura do metal do mancal escora nº 1	72	85	90	0-200 °C
G138ME2	Temperatura do metal do mancal escora nº 2	72	85	90	0-200 °C
G138ME3	Temperatura do metal do mancal escora nº 3	72	85	90	0-200 °C
G138MI	Temperatura de óleo da cuba do mancal combinado	51	75	não	0-200 °C

Itens de PV

```

ItmPv138MG1 = additem(grp_PvGdr1,'138MG1_AI1.OUT.VALUE','single');
ItmPv138MG2 = additem(grp_PvGdr1,'138MG2_AI1.OUT.VALUE','single');
ItmPv138MG3 = additem(grp_PvGdr1,'138MG3_AI2.OUT.VALUE','single');
ItmPv138ME1 = additem(grp_PvGdr1,'138ME1_AI1.OUT.VALUE','single');
ItmPv138ME2 = additem(grp_PvGdr1,'138ME2_AI2.OUT.VALUE','single');
ItmPv138ME3 = additem(grp_PvGdr1,'138ME3_AI1.OUT.VALUE','single');
ItmPv138MI = additem(grp_PvGdr1,'138MI_AI2.OUT.VALUE','single');

```

Itens de Alarme

```

ItmAlm138MG1 = additem(grp_AlmGdr1,'138MG1_ALM.HI_LIM.VALUE','single');
ItmAlm138MG2 = additem(grp_AlmGdr1,'138MG2_ALM.HI_LIM.VALUE','single');
ItmAlm138MG3 = additem(grp_AlmGdr1,'138MG3_ALM.HI_LIM.VALUE','single');
ItmAlm138ME1 = additem(grp_AlmGdr1,'138ME1_ALM.HI_LIM.VALUE','single');
ItmAlm138ME2 = additem(grp_AlmGdr1,'138ME2_ALM.HI_LIM.VALUE','single');
ItmAlm138ME3 = additem(grp_AlmGdr1,'138ME3_ALM.HI_LIM.VALUE','single');
ItmAlm138MI = additem(grp_AlmGdr1,'138MI_ALM.HI_LIM.VALUE','single');

```

Itens de Trip

```

ItmTrip138MG1 = additem(grp_TripGdr1,'138MG1_TRIP.HI_HI_LIM.VALUE','single');
ItmTrip138MG2 = additem(grp_TripGdr1,'138MG2_TRIP.HI_HI_LIM.VALUE','single');
ItmTrip138MG3 = additem(grp_TripGdr1,'138MG3_TRIP.HI_HI_LIM.VALUE','single');
ItmTrip138ME1 = additem(grp_TripGdr1,'138ME1_TRIP.HI_HI_LIM.VALUE','single');
ItmTrip138ME2 = additem(grp_TripGdr1,'138ME2_TRIP.HI_HI_LIM.VALUE','single');
ItmTrip138ME3 = additem(grp_TripGdr1,'138ME3_TRIP.HI_HI_LIM.VALUE','single');

```

SLR01 Sist. de Lub. e Ref. do Man Guia do Gdor e Combinado

Sist. de Lub. e Resf. do Mancal Guia do Gerador

Equipamentos

Bomba 1
 Motor Elétrico da Bomba 1
 Bomba 2
 Motor Elétrico da Bomba 2
 Trocador de Calor
 Filtros 1 e 2

Se G180GMO maior 28
Intertravamento no CKT de partida máquina por baixa vazão

Se G180GMA maior 60
Intertravamento na pré-condição

Condição para G163B1 e G163B2

TRIP/LOLO	2	desliga bba prioritária e liga bba reserva na pressão baixa (estas condições valem para o E3 PLUS)
ALM / LO	2,5	
ALM / HI	5	
ALM / HIHI TRIP/HIHI	6	desliga bba prioritária e liga bba reserva na pressão alta (estas condições valem para o E3 PLUS)

TAG	Descrição	PV	ALM	TRIP	FAIXA
G180GMO	Vazão de óleo	N	28	19	0-200 l/min
G180GMA	Vazão de água	N	60	40	0-200 l/min
G126ATMGSS	Temperatura de água na saída do trocador de calor	66	35	não	20-50 °C
G126OTMGSE	Temperatura do óleo na entrada do trocador de calor	47	55	não	20-70 °C
G126OTMGSS	Temperatura do óleo na saída do trocador de calor	30	55	não	20-70 °C
G163B1	Pressão na saída da motobomba 01	4.8	Ver Condição		0-10 bar
G163B2	Pressão na saída da motobomba 02	4.8	Ver Condição		0-10 bar
G163FOMGS	Pressão diferencial do filtro de óleo	1.6	2	não	0-2.5 bar

Sist. de Lub. e Resf. do Mancel Combinado

Equipamentos

Bomba AI

Motor Elétrico da Bomba AI

Bomba AJ

Motor Elétrico da Bomba AJ

Filtro Duplo

Trocador de Calor

Painel Elétrico de Comando e Controle

Se G180GMM maior 20
intertravamento no CKT de partida da máquina por baixa vazão.
Se G180LP maior 25
intertravamento para comando de partida da bomba AH e AG.
Se G163LW = 0
intertravamento no CKT de partida da máquina.
Se G163LV = 0
intertravamento no CKT de partida da máquina.

TAG	Descrição	PV	ALM	TRIP	FAIXA
G180LP	Vazão de óleo	N	25	19	0-200 l/min
G180GMM	Vazão de água	N	20	19	0-200 l/min
G163LW	Pressão na saída da motobomba AH	N	50	35	0-150 bar
G163LV	Pressão na saída da motobomba AG	N	50	35	0-200 bar
G163LX	Pressão na saída da motobomba AI	3.1	2	1.2	1-10 bar
G163LY	Pressão na saída da motobomba AJ	3.1	2	1.2	1-10 bar
G163MB	Pressão diferencial dos filtros de óleo	1.9	0.5	não	0-2.5 bar
G126ATMCS	Temperatura de água na saída do trocador de calor	N	35	não	20-50 °C
G126OTMCE	Temperatura de óleo na entrada do trocador de calor	45	55	não	20-70 °C
DT302-1	Densidade do óleo no mancal combinado	0.865	0.87	não	0.5-1.2
DT302-T	Temperatura do óleo na tubulação na entrada da cuba	47	não	não	

ItmPvDT3021 = additem(grp_PvGdr1,'DT302-1_AI1.OUT.VALUE','single');
ItmPvDT302T1 = additem(grp_PvGdr1,'DT302-T_1.OUT.VALUE','single');

SPC01 SISTEMA DE PROTEÇÃO E CONTROLE

Equipamentos

Relé de Sincronismo
Relé de Subtensão do Gerador
Relé de Subtensão da LT6CR (Lado UHE)
Relé de Proteção Motor Gerador 1
Relé de Proteção Motor Potência Ativa Nula
Relé de Proteção Motor Potência Reativa Nula
Relé de Perda de Excitação
Relé de Desequilíbrio de Corrente R46.TI
Relé de Desequilíbrio de Corrente R46.YD
Relé Térmico do Estator - Fase A
Relé Térmico do Estator - Fase B
Relé Térmico do Estator - Fase V
Relé Sobrecorrente de Fase Inst/Temp.
Relé Sobrecorrente de Neutro Inst/Temp.
Relé Sobrecorrente Inst/Temp. de Terra TF.1
Relé Sobrecorrente Inst/Temp. Trans. Excit.
Relé Sobrecorrente Inst/Temp. de Terra TR.1
Relé Sobrecorrente Inst/Temp. - Lado UHE R50.1N

Relé Sobrecorrente Inst/Temp. - Lado UHE R50.1P
 Relé Sobrecorrente Inst/Temp. - Lado SE R50.2N
 Relé Sobrecorrente Inst/Temp. - Lado UHE R50.2P
 Relé Sobretensão Temp. Fase A (Lado UHE)
 Relé Sobretensão Temp. Fase B (Lado UHE)
 Relé Sobretensão Temp. Fase V (Lado UHE)
 Relé de Desequilíbrio de Tensão R60.V1
 Relé de Desequilíbrio de Tensão R60.V2
 Relé de Proteção de Terra no Estator
 Relé de Proteção de Terra no Rotor
 Relé de Supervisão - Fio Piloto
 Relé de Sobrefrequência
 Relé de Bloqueio p/ Def. Aut. H-M-E
 Relé de Bloqueio p/ Def. Aut. P/P
 Relé de Bloqueio p/ Defeitos Elétricos
 Relé de Bloqueio p/ Defeitos Hidráulicos
 Relé de Bloqueio p/ Defeitos Mecânicos
 Relé de Bloqueio p/ Defeitos Tipo P1
 Relé de Bloqueio p/ Defeitos Tipo P2
 Relé de Bloqueio p/ Falta de Tensão
 Relé Diferencial do Gerador
 Relé Diferencial da Unidade
 Relé Diferencial Fio Piloto
 Relé Temp. Diferencial Fio Piloto
 Relé de Sobre Excitação
 Sistema de Proteção - Deslig. Rápido - DDR1000
 Relés Auxiliares do Gerador
 Relés Principais do Gerador
 Relés Auxiliares do Regulador de Tensão
 Relés Auxiliares do Regulador de Velocidade
 Relés de Proteção do TF6.01
 Relés Auxiliares da Turbina Hidráulica
 QLC01 Quadro de Comando e Controle Local
 Chaves de Comando e Controle do Grupo
 Aparelhos de Medição
 Anunciadores e Alarmes
 PCC01 Proteção de Comando e Controle
 PTF01 Proteção do Transformador
 PRV01 Proteção do Regulador de Velocidade
 PRT01 Proteção do Regulador de Tensão
 PLT01 Proteção da Linha de Transmissão
 PTH01 Proteção da Turbina
 PGE01 Proteção do Gerador

THP01 TURBINA HIDRÁULICA PRINCIPAL

Equipamentos

Rotor
 Eixo
 Sistema de Água de Resfriamento dos Carvões
 Pás do Rotor
 Ogiva
 Servo Motor das Pás
 Cabeçote Kaplan

SVE01 Sistema de Vedação do Eixo da Turbina

Equipamentos

Junta de Manutenção
Junta de Carvão
Fluxostato/Pressostato
Tubulações/Válvulas

Se G180MP maior que 90
intertravamento na pré-condição de partida da máquina.

Se G163CW = 0
intertravamento na pré-condição de partida da máquina.

TAG	Descrição	PV	ALM	TRIP	FAIXA
G180MP	Vazão de água de vedação	N	90	não	0-200 l/min
G163MQ	Pressão de água de vedação do eixo	N	2.5	1	0-10 bar
G163AVE	Pressão diferencial da água de vedação do eixo	N	2	não	0-10 bar
G163CW	Pressão de água da junta de manutenção	N	2	não	0-10 bar
G126AVE	Temperatura da água de vedação do eixo da turbina				

SRV01 Sistema de Regulação de Velocidade

Equipamentos

Bomba AE
Motor Elétrico da Bomba AE
Bomba AF
Motor Elétrico da Bomba AF
Filtros de Óleo
Trocadores de Calor
Acumulador Ar/Óleo
Acumulador Ar
Tanque de Óleo Sem Pressão
Válvulas/Tubulações
Regulador Hidráulico
Regulador Eletrônico
Transdutores de Posição
Transdutores de Potência Ativa

Sistema de Resfriamento

TAG	Descrição	PV	ALM	TRIP	FAIXA
G180MN	Vazão de água do trocador de calor	N	35	não	0-200 l/min
G126AR	Temperatura da água de resfriamento	34	35	não	0-200 °C
G126LK	Temperatura do óleo de regulação	46	48	55	0-200 °C

ItmPv126LK = additem(grp_PvGdr1,'126LK_AI1.OUT.VALUE','single');
 ItmPv126AR = additem(grp_PvGdr1,'126AR_AI2.OUT.VALUE','single');

Sistema de Alimentação de Óleo

Se G163CR = 36.8

intertravamento para abertura da válvula de isolamento, através da SD G1 20BBOA, considerando a subida de pressão.

TAG	Descrição	PV	ALM	TRIP	FAIXA
G171LI	Nível de óleo do tanque sem pressão	N	400	350	0-1200 mmH2O
G163CR	Pressão de óleo de regulação	38	35		0-60 bar
G1DTOR	Densidade do óleo de regulação	N	Não definida ainda		

Sistema de Comando Hidráulico

Condições para G171LA

ALM / HIHI	1637	PV e ALM no supervisorio e gera alarme no campo através da SD G171LCA.
ALM / HI	1587	PV supervisorio , pré-condição para comandar a eletro-válvula de injeção de ar no balão de 40bar, através da SD G171CJA.
Condição	maior igual 1437	PV no supervisorio e intertravamento no CKT de partida da máquina (pré-condição) quando o nível está abaixo desta faixa.
ALM/LO	1287	PV e ALM no supervisorio e pre-condição para ligar a bba reserva (AE ou AF) de óleo de regulação via SMC.
ALM / LOLO	834	PV e ALM no supervisorio e gera alarme no campo, através da SD G171LD1A.
TRIP / LOLOLO	799	PV e ALM. no supervisorio, gera alarme no campo e TRIP na máquina, através da SD G171LD2A.

TRIP / LOLOLOLO	420	PV e ALM no supervisorio, gera alarme no campo e trip da máquina, através da SD G1LEA, comanda o fechamento da válvula de isolamento., através da SD G120BBCA e faz a partida de emergência das motobombas AE e AF, através do SMC.
--------------------	-----	---

Se G163LGF > 41

intertravamento no fechamento da eletro-válvula de injeção de ar no balão de 40 bar.

TAG	Descrição	PV	ALM	TRIP	FAIXA
G171LA	Nível de óleo do acumulador ar/óleo	N	Ver Condições		0-2000 mmH2O
G1TPA	Posição do nível de óleo do acumulador ar/óleo	N	Não definida ainda		
G163LGF	Pressão de óleo do acumulador ar/óleo	40	34	31	0-60 bar

SMD01 Sistema Distribuidor da Turbina

Equipamentos

Pré-Distribuidor

Servo-Motor

Anel Superior

Anel Inferior

Palhetas Móveis

Tampa Inferior

Pinos de Cizalhamento

Caixas de Passagens (Fins de Curso)

SAD01 Sistema de Adução e Descarga da Turbina

Caixa Espiral

Tubo de Sucção

Septos Divisores

Válvulas de Esgotamento e Drenagem

Poços das Válvulas de Esgotamento/ Drenagem

Se G163PEP ?? 3.2

intertravamento no ckt de comando de abertura das comportas de tomada água quando a pressão for de 3,2 bar.

Condições para PESU

ALM / HI	22000	
C5	6000	liga a segunda motobomba BS01 ou BS02 de esgotamento, através do SMC.
C4	5000	liga primeira a motobomba BS01 ou BS02 de esgotamento, através do SMC..

C3	3000	desliga as motobombas BS01 ou BS02 de esgotamento, através do SMC.
ALM / LO	2900	desliga as motobombas BS01 ou BS02 de esgotamento, através do SMC.
C2	2800	liga motobomba BS03 de esvaziamento, através do E3PLUS.
C1	920	desliga a motobomba BS03 de esvaziamento, através do E3PLUS.
ALM / LOLO	820	desliga motobomba BS03 de esvaziamento, através do E3PLUS.

TAG	Descrição	PV	ALM	TRIP	FAIXA
G163RJ	Pressão diferencial de água turbinada	N			0-7000 mmH2O
G163DJ	Pressão da água da caixa espiral	N			0-10 bar
G163PEP	Pressão estabelecida da caixa espiral normal	N			0-10 bar
G163TS	Pressão no tubo de sucção	N			0-5 bar
PESU	Nível do poço de esgotamento da turbina	N	Ver Condições		0-26000 mmH2O

SLF01 Sistema de Levantamento e Frenagem

Equipamentos

Macacos Pneumáticos

Reservatório de Ar (7,0 BAR)

Eletroválvula

Pista de Frenagem (Placas)

Bomba de Levantamento

Motor Elétrico da Bomba de Levantamento

Reservatório de Óleo

Painel Elétrico de Comando e Controle

Filtro de Ar (Sistema de Frenagem)

Se G163AF < 4

intertravamento no CKT de partida da máquina por pressão baixa.

TAG	Descrição	PV	ALM	TRIP	FAIXA
G163AF	Pressão de ar de frenagem	8.2	4	não	0-15 bar
G163FI	Pressão de frenagem indevida	N	0.5	não	0-10 bar

SDT01 Sistema de Drenagem da Tampa da Turbina

Equipamentos

Bomba AK

Motor Elétrico da Bomba AK

Bomba AL

Motor Elétrico da Bomba AL

Painel Elétrico de Comando e Controle

Chaves de Nível

Condição para G171SDT

TRIP / HIHI	300	gera alarme no campo e TRIP, através da SD G171LQA.
ALM / HI	250	gera alarme no campo, através da SD G171CHA e liga a 2ª motobomba (AK e AL), através do E3PLUS.
Condição 1	150	liga a 1ª motobomba AK ou AL, via E3PLUS, através do E3PLUS.
Condição 2	50	desliga as duas motobombas AK ou AL, através do E3PLUS.

TAG	Descrição	PV	ALM	TRIP	FAIXA
G171SDT	Nível de água do poço de drenagem da turbina	N	250	300	0-400 mmH2O
G163BAKL	Pressão de água na saída das motobombas AK/AL	N	2	1.5	0-10 bar

SVA01 Sistema da Válvula de Aeração

Equipamentos

Válvula de Aeração - Lado Jusante

Válvula de Aeração - Lado Montante

SME01 SISTEMA DE MEDIÇÃO

CTA01 Comportas da Tomada D'água

Equipamentos

Central Hidráulica

Painel de Comando e Controle Local

Chaves Fins-de-Curso

Motobomba Alta Pressão/Baixa Vazão

Motobomba de Baixa Pressão/Alta Vazão

Caixa de Terminais

Eletroválvulas

Bomba Manual

Indicador de Posição da Comporta - Sec.

Condições para G163PBA

ALM / LOLO TRIP/LOLO	30	Desliga a motobomba principal (MBCT1 ou MBCT2) e liga motobomba reserva (MBCT1 ou MBCT2).
ALM / LO	60	
ALM HI	120	
ALM / HIHI TRIP/HIHI	153	desliga a motobomba principal (MBCT1 ou MBCT2) ou a reserva (MBCT1 ou MBCT2).

TAG	Descrição	PV	ALM	TRIP	FAIXA
G163PC	Pressão diferencial de perda de carga das comportas	N	3	não	0-10 bar
G163PBA	Pressão de óleo na saída das motobombas das comportas de tomada d'água	N	Ver Condições		0-200 bar
G163TC	Pressão de óleo na tubulação das comportas de tomada d'água	N	80	não	0-200 bar
G171CB	Nível de óleo na centralina das comportas de tomada d'água	N	250	200	0-1200 mmH2O

TF601 Transformador Elevador

Equipamentos

Óleo Mineral Isolante

Motobombas de Circulação de Óleo

Motoventiladores

Painel de Supervisão

Indicador de Posição dos TAP'S

Transformador de Corrente TCH 0/1/2/3I

Transformadores de Corrente TCH 1/2/3J

Transformadores de Corrente TCH0J

Transformador de Corrente TCH2K

Transformador de Corrente TCX2J

TAG	Descrição	PV	ALM	TRIP	FAIXA
G149TP	Temperatura do enrolamento primário do trafo elevador	55	105	120	0-200 °C
G149TS	Temperatura do enrolamento secundário do trafo elevador	58	105	120	0-200 °C
G126F	Temperatura de óleo	49	85	95	0-200 °C

LT601 Linha de Transmissão 230 kV - (Usina/SE)

Equipamentos

Pára-raio 230kv - Fase A (Lado UHE)

Pára-raio 230kv - Fase B (Lado UHE)

Pára-raio 230kv - Fase V (Lado UHE)

Sistema de Proteção da LT6CR (Usina/SE)

Galeria de Auxiliares Comuns das UGH

Sistema de Tratamento e Distribuição de Água da Usina

Se G163FAL = 0.3

aciona a SD G163FALA do sistema de limpeza do filtro.

TAG	Descrição	PV	ALM	TRIP	FAIXA
G163FAL	Pressão diferencial da água do filtro auto-limpante	0.1	0.3	0.4	0-2.5 bar
SAS63	Pressão de saída das motobombas AN e AR	N	2.5	2	0-10 bar
MR63	Pressão diferencial do filtro de água de sucção das bombas AN e AR	N	0.4	2	0-2.5 bar
TQS71MO1	Nível de água do tanque de selagem 01	N			
TQS71MO2	Nível de água do tanque de selagem 02	N			

Sistema de Drenagem da Casa de Força

Condições para PDRU

ALM / HI	9200	
C3	9000	liga a segunda motobomba BDR01 ou BDR02 de drenagem, através do E3PLUS.
C2	8500	liga primeira motobomba BDR01 ou BDR02 de drenagem, através do E3PLUS.
C1	7500	desliga as motobombas BDR01 ou BDR02 de drenagem, através do E3PLUS.
ALM / LO	7300	desliga as motobombas BDR01 ou BDR02 de drenagem, através do E3PLUS.

TAG	Descrição	PV	ALM	TRIP	FAIXA
PDRU	Nível do Poço de drenagem da usina	3.5	Ver Condições		0-20000 mmH2O

Sistema de Ar de Regulação

Se AO63MU ou AP63MU ou AQ63MU < 0.2
desligar o compressor na baixa pressão.

Condições para BAR63

ALM / HI	51	
Condição 1	49,5-50,5	desliga os compressores (AO, AP e AQ), através do E3PLUS.
Condição 2	47	liga o 1º compressor AO ou AP ou AQ, através do E3PLUS.
Condição 3	46	liga o 2º compressor AO ou AP ou AQ, através do E3PLUS.
ALM / LO	45	liga o 3º compressor AO ou AP ou AQ, através do E3PLUS.
ALM / LOLO	41	

TAG	Descrição	PV	ALM	TRIP	FAIXA
AO63MU	Pressão de óleo de lubrificação do compressor AO	N	1	não	0-6 bar
AP63MU	Pressão de óleo de lubrificação do compressor AP	N	1	não	0-6 bar
AQ63MU	Pressão de óleo de lubrificação do compressor AQ	N	1	não	0-6 bar
BAR63	Pressão do balão de ar de regulação	N	Ver Condições		0-70 bar

Sistema Anti-Incêndio

ALM / HI	6	segurança para desligamento independente se está em local ou remota
Condição 1	5,5	desliga bba ACI01 de combate a incêndio, através do E3PLUS. Considerar uma temporização para que quando a linha pressurizar a 1 vez não veja esta condição.
Condição 2	2,4	liga bba ACI01 de combate a incêndio, através do E3PLUS. Considerar uma temporização para que quando a linha pressurizar a 1 vez não veja esta condição.
ALM / LO	2	

TAG	Descrição	PV	ALM	TRIP	FAIXA
ATCI63	Pressão da água na tubulação de combate a incêndio	N			0-15 bar

Sistema de Ar de Serviço

Condições para BAS63

ALM / HI	7,4	
Condição 1	7	desliga os compressores CAS01 e CAS02, através do E3PLUS.
Condição 2	6,2	liga o 1º compressor CAS01 ou CAS02 através do E3PLUS.
ALM / LO	5,9	liga o 2º compressor CAS01 ou CAS02 através do E3PLUS.
ALM /LOLO	4,5	

TAG	Descrição	PV	ALM	TRIP	FAIXA
BAS63	Pressão do balão de ar de serviço	6.5	Ver Condições		0-15 bar
CAS0126AR	Temperatura na descarga do compressor 1	N	93	95	0-120 °C
CAS0226AR	Temperatura na descarga do compressor 2	N	93	95	0-120 °C
CAS163F	Pressão diferencial do filtro de óleo no compressor 1	N	1.5	1.7	0-2.5 bar
CAS263F	Pressão diferencial do filtro de óleo no compressor 2	N	1.5	1.7	0-2.5 bar

5. Comunicação Via OPC do Matlab com o Servidor OPC da Usina Hidreletrica de Balbina

A continuação apresento o trabalho feito em entorno Matlab usando comunicação via Servidor OPC quem disponibiliza os dados dos transmissores da smar e fazendo o tratamento das sinais obtidas atravez das TAGs dos valores de Temperatura (Fig. 2).

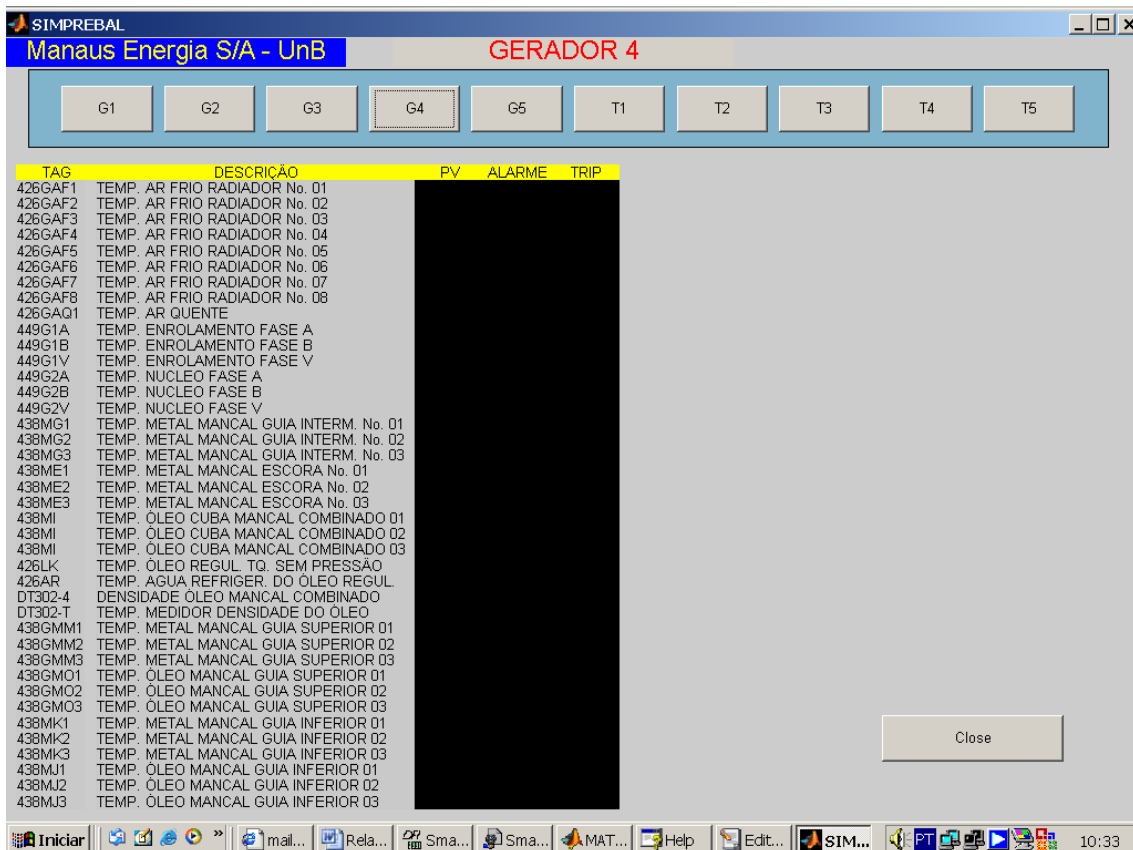


Fig. 2 – supervisorio em Matlab os Geradores e Transformadores da UGH de balbina

6. Desenvolvimento de aplicação em Java para leitura de Tags de um arquivo e escritura dos valores lidos no outro arquivo.

Esta aplicação desenvolvida em Java consiste em ler as tags gravadas no arquivo "tags.txt" enviar essas tags ao servidor OPC e esperar o retorno dos valores e salva-los em outro arquivo "dados.txt" (Fig. 3).

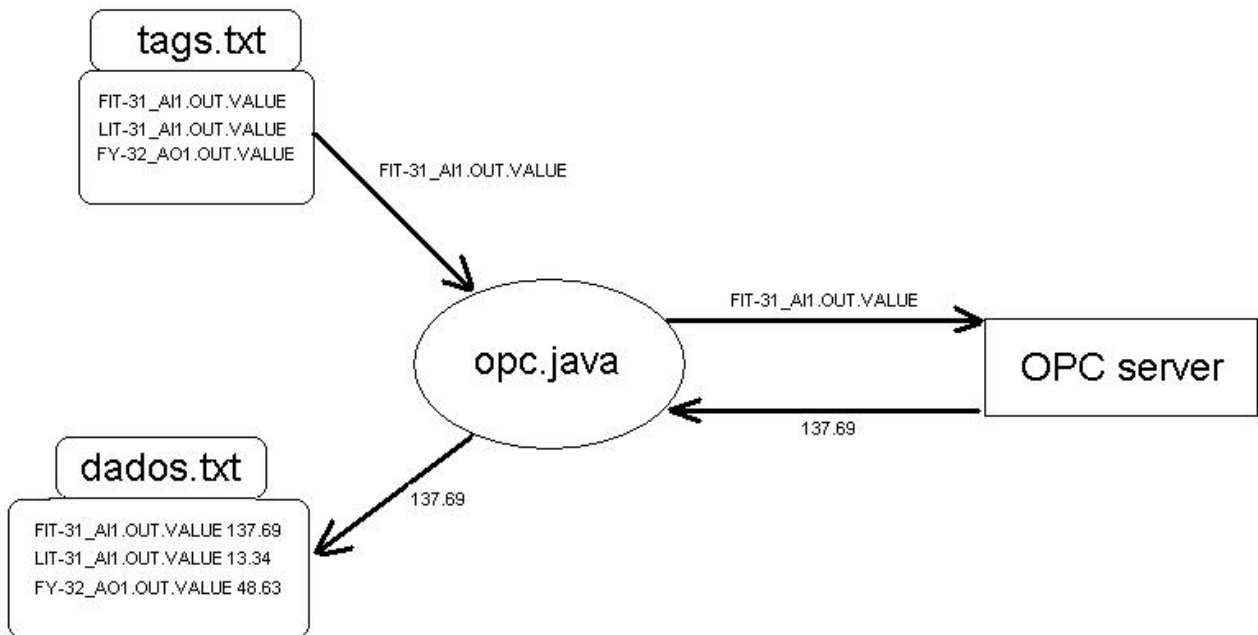


Fig. 3 – diagrama de fluxo da informação entre Tags e o OPC server

6.1 Leitura das Tags

Para a leitura das tags uso-se a seguinte método, onde os nomes das Tags do arquivo é convertido em forma de vetor para seu processamento, tendo como entrada o nome do vetor e dando como retorno o vetor das tags. Para chamar este método escreve-se uma linha de código `Vector v = fileToVector("tags.txt");` no método onde vai ser usado as tags a ser monitoradas.

```

public Vector fileToVector(String fileName) {
    Vector v = new Vector();
    String inputLine;
    try {
        File inFile = new File(fileName);
        BufferedReader br = new BufferedReader(new InputStreamReader(
            new FileInputStream(inFile)));
        while ((inputLine = br.readLine()) != null) {
            v.addElement(inputLine.trim());
        }
    }
}
  
```

```

    }
    br.close();
} // Try
catch (FileNotFoundException ex) {
    //
} catch (IOException ex) {
    //
}
return (v);

```

6.2 Escritura dos valores obtidas das Tags

Os valores das tags obtidas são salvas no arquivo “dados.txt”, para isso crie-se o seguinte método onde escreve as tags com os seus correspondentes valores. Para chamar a este método escreve-se a linha de código seguinte:

```
writeToFile("dados.txt",ci.name + " " + ci.getValueAt(col),true,true);
```

```

public boolean writeToFile(String fileName, String dataLine,
    boolean isAppendMode, boolean isNewLine) {
    if (isNewLine) {
        dataLine = "\n" + dataLine;
    }
    try {
        File outFile = new File(fileName);
        if (isAppendMode) {
            dos = new DataOutputStream(new FileOutputStream(fileName, true));
        } else {
            dos = new DataOutputStream(new FileOutputStream(outFile));
        }
        dos.writeBytes(dataLine);
        dos.close();
    } catch (FileNotFoundException ex) {

```

```

    return (false);
} catch (IOException ex) {
    return (false);
}
return (true);
}

```

6.3 Tela de apresentação dos dados monitorados

OS dados monitorados são apresentados em duas colunas Item e Value segundo mostrado na Fig. 4.

The screenshot shows a window titled "SIMPREGAL: Teste de OPC com PD3 da Smar" with a standard Windows interface. Inside the window, there is a table titled "Monitored items" with two columns: "Item" and "Value". The table contains 30 rows of data, listing various OPC tags and their corresponding numerical values.

Item	Value
FIT-31_AI1.OUT.VALUE	140.094482421875
LIT-31_AI1.OUT.VALUE	-15.108882904052734
FIT-31_INTG1.OUT.VALUE	169669.578125
TY-31_AO1.OUT.VALUE	4.0
FY-31_AO1.OUT.VALUE	5.0
TIT-31_AI1.OUT.VALUE	24.70465087890625
FIT-32_AI1.OUT.VALUE	0.0
FIT-32_INTG2.OUT.VALUE	20567.87109375
FY-32_AO1.OUT.VALUE	6.596503080800176E-4
TIT-32_AI1.OUT.VALUE	25.865843963623047
FIT-31_DIAG1.BLOCK_ALM.VALUE	0.0
FY-31_AO1.PV.VALUE	4.987347621917725
LIT-31_DIAG1.BLOCK_ALM.VALUE	0.0
TIT-31_DIAG1.BLOCK_ALM.VALUE	0.0
TIT-32_DIAG1.BLOCK_ALM.VALUE	0.0
TY-31_AO1.PV.VALUE	0.0
DFI-PD3_DIAG1.BLOCK_ALM.VALUE	0.0
DFI-PD3_DIAG1.BLOCK_ALM.ALARM_STATE	0.0
DFI-PD3_DIAG1.MODE_BLK.NORMAL	128.0
FIT-31_AI1.FIELD_VAL.VALUE	1.9626468420028687
FIT-31_AI1.HI_ALM.VALUE	0.0
FIT-31_AI1.LO_ALM.VALUE	0.0
FIT-32_AI1.HI_ALM.VALUE	0.0
FIT-32_AI1.LO_ALM.VALUE	0.0
TIT-31_AI1.HI_ALM.VALUE	0.0
TIT-31_AI1.LO_ALM.VALUE	0.0
TIT-32_AI1.HI_ALM.VALUE	0.0
TIT-32_AI1.LO_LO_ALM.VALUE	0.0

Fig. 4 – Tela mostrando as tags e os valores do OPC server

7. Integração da Comunicação OPC com I-kernel

Para a Integração da comunicação através de OPC com o I kernel, grava-se toda a configuração, Hostname(nome ou IP onde encontra-se o OPC server), PrgID(o nome do OPC server) e as Tags(nomes das grandezas reconhecidas pelo servidor OPC) do devicex a ser monitorados no arquivo “config.ini”, mostrado na Fig. 5.

O classe configuração do I-kernel leera toda a configuração do arquivo “config.ini” as tags lidas seram enviadas ao OPC server onde o OPC server retornara os valores das grandezas. Os valores das tags obtidas serão usados para o processamento inteligente por o JESS, Fuzzy-JESS ou Redes Neurais.

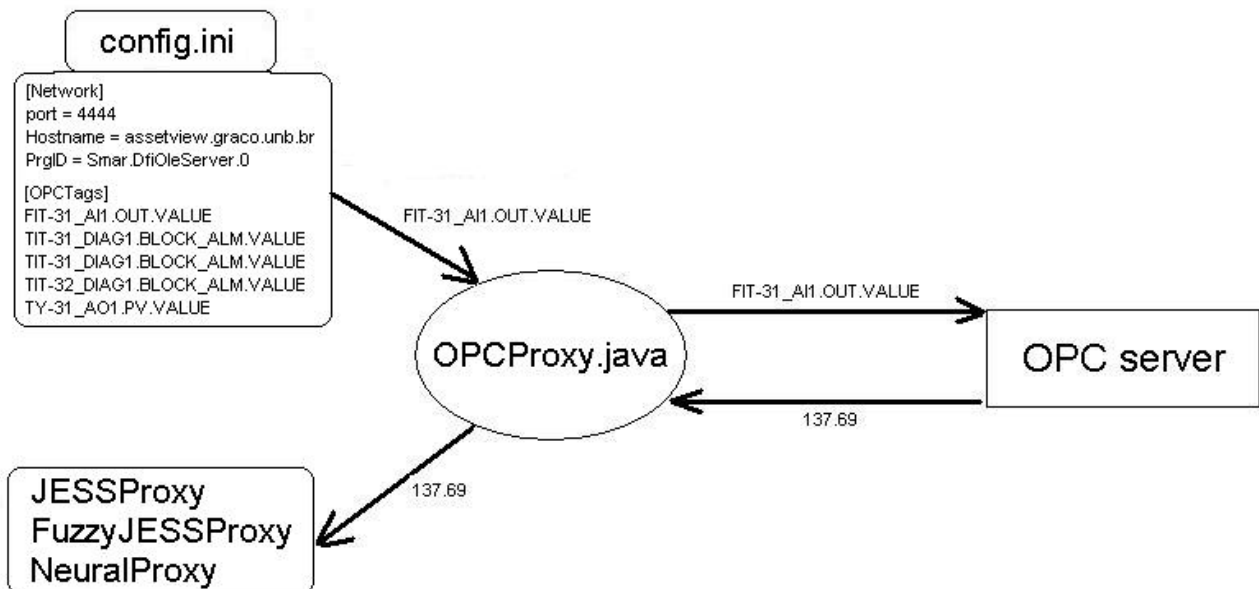


Fig. 5 – Diagrama de fluxo da informação na Classe OPCProxy do Simprebal

8. Integração do sistema IKernel com o Servidor OPC.

Para a Integração da comunicação através do servidor OPC com o Ikernel, grava-se no arquivo “config.ini” toda a configuração, host(nome ou IP do computador onde encontra-se o O servidor PC), progid(o nome do servidor OPC), domain(Domínio onde encontra-se o computador onde esta o servidor OPC), user(O nome do usuário do computador), password(a senha do usuário do computador) e as tags(nomes das grandezas reconhecidas pelo servidor OPC).

O classe “configuration” do I-kernel lê toda a configuração do arquivo “config.ini” as tags lidas serão processadas pela classe “OPCProxy” do Ikernel e enviadas ao servidor OPC

onde o servidor OPC retornara os valores das grandezas. Os valores das tags obtidas serão usados para o processamento inteligente da maquina de inferência das regras de produção usando o JESS, e futuramente por o Fuzzy-JESS ou Redes Neurais.

8.1 Arquivo de Tags

O arquivo "config.ini" tem a seguinte estrutura:

- Lista dos servidores OPC

```
[OPCServers]
dfi
df65
```

- Parâmetros de cada OPC server

```
[dfi]
progid = Smar.DfiOleServer.0
host = 164.41.17.129
domain = graco.unb.br
user = alvares
password = eletronorte2003
tags = dfiTags
```

```
[df65]
progid = Smar.DF65Server.1
host = 164.41.17.129
domain = graco.unb.br
user = alvares
password = eletronorte2003
tags = df65Tags
```

- Lista das tags que encontra-se em cada servidor OPC.

```
[dfiTags]
gl.t.metal.mguia.inf1 = FIT-31_AI1.PV.VALUE
gl.t.metal.mguia.inf2 = LIT-31_AI1.PV.VALUE
gl.t.metal.mguia.inf3 = TIT-31_AI1.PV.VALUE
gl.t.oleo.mguia.inf1 = FIT-32_AI1.PV.VALUE
gl.t.oleo.mguia.inf2 = TIT-32_AI1.PV.VALUE
gl.p.oleo.cuba = TIT-32_AI1.OUT.VALUE
gl.st.t.metal.mguia.inf1 = FIT-31_AI1.PV.STATUS
gl.st.t.metal.mguia.inf2 = LIT-31_AI1.PV.STATUS
gl.st.t.metal.mguia.inf3 = TIT-31_AI1.PV.STATUS
gl.st.t.oleo.mguia.inf1 = FIT-32_AI1.PV.STATUS
gl.st.t.oleo.mguia.inf2 = TIT-32_AI1.PV.STATUS
gl.st.p.oleo.cuba = TIT-32_AI1.OUT.STATUS
```

```
[df65Tags]
NivelBaixo = PD3.A.001.00
TempHiTqAquec = PD3.A.001.01
Emergencia = PD3.A.001.04
TempHiTqMist = PD3.A.002.05
LocalBBA1 = PD3.E.010.00
```

LocalBBA2 = PD3.E.010.01
Termostato = PD3.E.011.00
ChavedeNivel = PD3.E.011.01
SeloBBA1 = PD3.E.011.04
DefeitoBBA1 = PD3.E.011.05
SeloBBA2 = PD3.E.011.10
DefeitoBBA2 = PD3.E.011.11
AquecInhibido = PD3.S.012.00
BBA1Deslig = PD3.S.012.01
BBA2Deslig = PD3.S.012.02
BBA1 = PD3.S.012.03
BBA2 = PD3.S.012.04
Temp = PD3.S.012.05
Nivel = PD3.S.012.06
Sirene = PD3.S.012.10

8.2 Tela para escolher uma tag.

Na Fig. 6, mostra-se a tela de acordo a arvore hierárquica dos ativos da usina de Balbina.

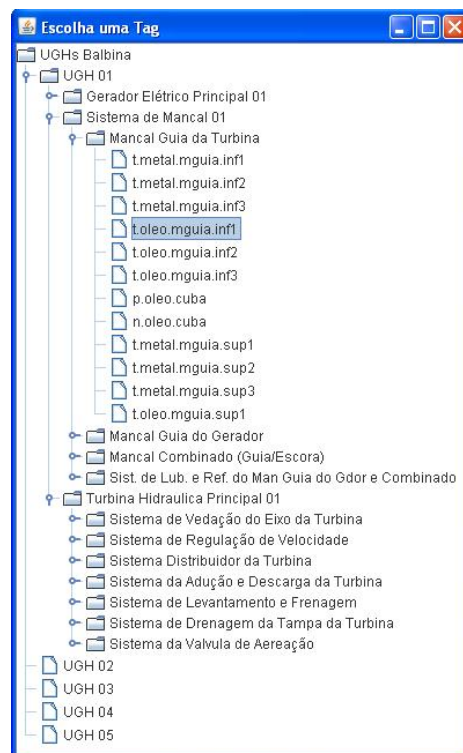


Fig. 6 – Tela para escolha de tags.

8.3 Tela de apresentação Inspeção de Variáveis.

A figura 7 mostra os valores obtidos do servidor OPC



Tag	Valor
clock	18:37:00
t.metal.mguia.inf1	128
t.metal.mguia.inf2	138.76472
t.metal.mguia.inf3	-14.89062
t.oleo.mguia.inf1	24,15805
t.oleo.mguia.inf2	4.0
t.oleo.mguia.inf3	1.0
p.oleo.cuba	325538.5
n.oleo.cuba	0.0
t.metal.mguia.sup1	24.603748
t.metal.mguia.sup2	1.0
t.metal.mguia.sup3	385612.28
t.oleo.mguia.sup1	24473.16

Fig. 7 - Tela dos valores das variáveis

9. Integração do sistema IKernel com o Servidor DB.

Esta aplicação desenvolvida em Java consiste em ler as tags gravadas no arquivo "config.ini", processar estas tags no programa em Java, enviar essas ao servidor DB e esperar o retorno dos valores para logo usar os valores obtidos no análise de tendência.

9.1 Arquivo de Tags

- Lista dos Servidores de Banco de Dados

```
[DBServers]
Assetview
```

- Parametros de cada Servidor DB

```
[Assetview]
user = sa
password = smar
DriveAddress = com.microsoft.sqlserver.jdbc.SQLServerDriver
ComAddress= jdbc:sqlserver://164.41.17.129:1092;databaseName=AssetView
tags = DBTags
```

- Lista das tags, pode ser real(do banco de dado especificado) ou simulado(valores armazenados no arquivo "values.ini").

```
[DBTags]
temp3 = real
temp4 = real
temp5 = real
temp6 = real
temp7 = real
```

```
temp8 = real
temp9 = real
temp10 = real
temp11 = real
temp12 = real
temp13 = real
temp14 = real
temp15 = real
temp16 = real
temp17 = real
temp18 = real
temp19 = real
temp20 = real
```

- Lista SELECT das tags a serem lidas do banco de dados na tabela e na posição.

```
[SELECT]
```

```
temp3 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='13'
temp4 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='14'
temp5 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='15'
temp6 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='16'
temp7 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='17'
temp8 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='18'
temp9 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='19'
temp10 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='20'
temp11 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='21'
temp12 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='22'
temp13 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='23'
temp14 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='24'
temp15 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='25'
temp16 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='26'
temp17 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='27'
temp18 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='28'
temp19 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='29'
temp20 = SELECT * FROM ParametersHistory WHERE CodParametersHistory='30'
```

```
[INSERT]
```

```
DBinsert = INSERT INTO DBTagsInformation VALUES ('>?', '>?')
```

```
[UPDATE]
```

```
temp4 = UPDATE DBTagInformation SET Value = '>?' WHERE Indice = '2'
```

10. Ferramenta de configuração e monitoramento.

- Esta ferramenta desenvolvida em Java Fig. 8 é lado cliente do SIMPREBAL, pegando dados do lado servidor (ikernel).

The screenshot displays the SIMPREBAL V2.8 software interface. At the top, there is a menu bar with options: Arquivo, Help, Sinótico, iKernel, and Inspeção. Below the menu are three buttons labeled 'Atalho I', 'Atalho II', and 'Atalho III'. The main workspace is divided into several panels. On the left, there are three panels showing 3D models of turbine components, labeled 'Unidade Geradora Hidráulica I', 'Unidade Geradora Hidráulica IV', and 'Unidade Geradora Hidráulica V'. Each panel has labels for various parts: 'cabeçote', 'tampas', 'eixo', 'cgive do rotor', and 'pa'. In the center, a 'Escolha uma Tag' dialog box is open, showing a hierarchical tree of components. The selected tag is 'Loleo.mguia.inf1'. On the right, an 'Inspeção de Variáveis' dialog box is open, displaying a table of variable values.

Tag	Valor
clock	18:34:11
t.metall.mguia.inf1	128
t.metall.mguia.inf2	138.99384
t.metall.mguia.inf3	-14.89062
Loleo.mguia.inf1	24.168234
Loleo.mguia.inf2	4.0
Loleo.mguia.inf3	1.0
p.oleo.cuba	325528.47
n.oleo.cuba	0.0
t.metall.mguia.sup1	24.576086
t.metall.mguia.sup2	1.0
t.metall.mguia.sup3	385602.28
Loleo.mguia.sup1	24473.16

At the bottom of the window, there is a 'Painel Geral' section with the following text:

Valores TAGS: 139.13808
Valores TAGS: 139.13461
Valores TAGS: 138.77165

Fig. 8 – ferramenta configuração e monitoramento

Anexo 1
Paper Submetido e ACEITADO no CAIP2007

Sistema de Manutenção Baseada Em Condição Para Usina Hidrelétrica de Balbina

Alberto J. Álvares¹, Edgar J. Amaya¹, Rosimarci Tonaco¹

(1) Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Mecânica, Curso de Sistemas Mecatrônicos. Campus Universitário Darcy Ribeiro, 70910-900 – Asa Norte – Brasília – DF, Brasil.(alvares@AlvaresTech.com, eamaya@unb.br, rosimarci@hotmail.com)

RESUMO

A metodologia foi desenvolvida no contexto do projeto Modernização da Área de Automação de Processos das Usinas Hidrelétricas de Balbina e Samuel, onde o objetivo é o desenvolvimento de um sistema inteligente de manutenção preditiva para a usina de Balbina. A metodologia SIMPREBAL é baseada em conceitos de manutenção centrada em confiabilidade, sendo utilizada para analisar os modos e efeitos de falhas das Unidades Geradoras Hidráulicas de Balbina a partir das grandezas monitoradas pelo sistema de supervisão e controle da Usina, focando a sua análise no sistema da Turbina. O objetivo é conceber uma metodologia para coleta e análise de dados das unidades geradoras hidráulicas da usina, e a implementação de um sistema computacional com vistas à produção de diagnósticos que auxiliem a tomada de decisão quanto às ações operacionais e de manutenção das máquinas, contribuindo desta maneira ao aumento da disponibilidade dos equipamentos e sugestões para a tomada de decisões dos operadores e o pessoal de operação.

INTRODUÇÃO

O modelo OSA-CBM (*Open System Architecture for Condition Based Maintenance*) que consiste em sete camadas mostrado na Fig. 1, será utilizado como referência para o desenvolvimento do sistema de manutenção inteligente baseado em condição, maiores detalhes sobre o modelo, podem ser encontrados no *site* <http://www.osacbm.org>. A noção de uma arquitetura estendida em camadas usada aqui é consistente com o conceito usado em Buschman (1996). As camadas hierárquicas representam uma transição lógica ou um fluxo da saída dos sensores para a camada de tomada de decisão, através das camadas intermediárias assim ate chegar à camada de



Fig. 1: Arquitetura OSA-CBM e suas 7 camadas

A seguir será apresentado um exemplo das sete camadas estudando o caso do mancal combinado do sistema da turbina e como estas se ajustam ao sistema SIMPREBAL que é baseado em manutenção centrada em confiabilidade(MCC). segundo Fleming et al. (1997), a MCC

envolve uma consideração sistemática das funções do sistema, a maneira como essas funções falham e um critério de priorização explícito baseado em fatores econômicos, operacionais e de segurança para a identificação das tarefas de manutenção aplicáveis tecnicamente e custos eficientes no combate a essas falhas (p.53).

ESTUDO DE CASO: MANCAL COMBINADO DO SISTEMA DA TURBINA

Esta seção apresenta uma breve explicação sobre os mancais do grupo turbina-gerador, bem como, um estudo de caso utilizando como exemplo o mancal combinado. O mancal combinado do sistema é responsável por suportar o empuxo radial do eixo da turbina e o empuxo hidraulico vertical. O arranjo dos mancais do grupo turbina-gerador mostrado na figura 2 é o seguinte: o mancal de guia inferior ou mancal da turbina é colocado próximo à extremidade inferior do eixo, na tampa superior e compatível com o arranjo e o acesso à vedação do eixo; o mancal combinado ou mancal guia-escora é suportado por uma estrutura cônica apoiada na tampa superior; e o mancal de guia superior ou mancal do gerador é locado imediatamente acima do rotor do gerador, no centro da aranha superior do gerador, cujo peso deve ser suportado pela carcaça do estator, mas as forças radiais do mancal devem ser suportadas pela estrutura de concreto.

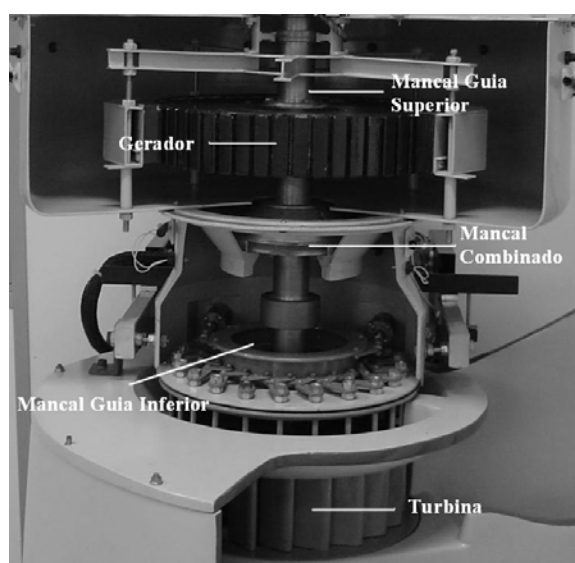


Fig. 2: Maquete de uma Unidade Geradora Hidráulica

Uma vez definida as funções do sistema utilizando as sete camadas do modelo OSA-CIM, pode-se definir os módulos das camadas para o sistema em estudo do mancal combinado.

Módulo de sensor: A aquisição de dados necessários para o processamento da informação relativo ao mancal combinado, a obtenção dos TAGs referentes a cada equipamento é feita via OPC e/ou SQL para TAGs da instrumentação Fieldbus Foundation da Smar da serie 302 é feita segundo Smar (2001), e os banco de dados da Rockwell via JDBC mostrado na Fig. 3.

Processamento de sinal: O sistema deve acessar os dados da Usina de Balbina a partir dos Bancos de Dados SQL utilizados pelo Sistema de Monitoramento de Balbina, Sistema de Gestão de Operação e Manutenção ou diretamente da instrumentação Fieldbus e Rockwell por meio do servidor OPC mostrado na Fig. 3, sendo que estes disponibilizam as informações on-line dos equipamentos. Para obtenção das variáveis on-line via OPC de acordo as configurações dos blocos funcionais segundo Smar(2005), esta-se desenvolvendo um programa para aquisição das TAGs necessárias pra avaliação dos equipamentos em C/C++ e uma interface C/C++ com Java através de JNI (Java Native Interface).

Monitoração de Condição: Esta camada determina a condição do sistema atual, subsistema, ou indicadores de condição de componente baseando-se em algoritmos, sistemas inteligentes e na

saída dos módulos de sensores e processamento de sinal. É possível fazer o uso de histórico de condição local e fornecer parâmetros para o modelo. Duas dimensões devem ser consideradas: primeira, o sistema *Assetview* da Smar e o sistema *MES Maximo* fornecem análises relativa à condição da instrumentação *FieldBus*; segunda dimensão, o sistema *SIMPREBAL* fornece a condição do sistema monitorado. Um exemplo de regras de produção pode ser visto na Tabela 1.

Tabela 1: Regras de produção e Possíveis dificuldades de funcionamento

Condição	Falha	Causa provável	Solução
Pressão de Oleo > 60bar	Trocador de calor sujo	Sujera nas placas do trocador de calor ativo.	Fazer o Bypass para o outro trocador de calor e Limpar as placas do trocador sujo.
Nível de Oleo na cuba <15mmH2O	Falta de óleo na cuba	Vazamento de oleo	aumentar oleo na Cuba
Temp. do metal mancal > 75 °C	Vibração crescente	Funcionamento com cargas abaixo do nominal.	Evitar que a unidade geradora funcione por tempo demasiado com cargas baixas.

Avaliação de saúde: Na primeira dimensão o conversor *FieldBus* e o sistema *Assetview* da Smar fornecem a avaliação da saúde da instrumentação *FieldBus*. Na segunda dimensão o sistema *SIMPREBAL* fornece a avaliação da saúde do sistema monitorado. A avaliação da saúde do equipamento ocorre através da extração das características de cada equipamento e posterior detecção de anomalias e diagnósticos dos mesmos. Para tanto, deve-se criar um conjunto de regras de produção mostrado na Tabela 1, que seja capaz de expressar e avaliar a condição atual do equipamento, evitando desta forma uma possível falha/falta.

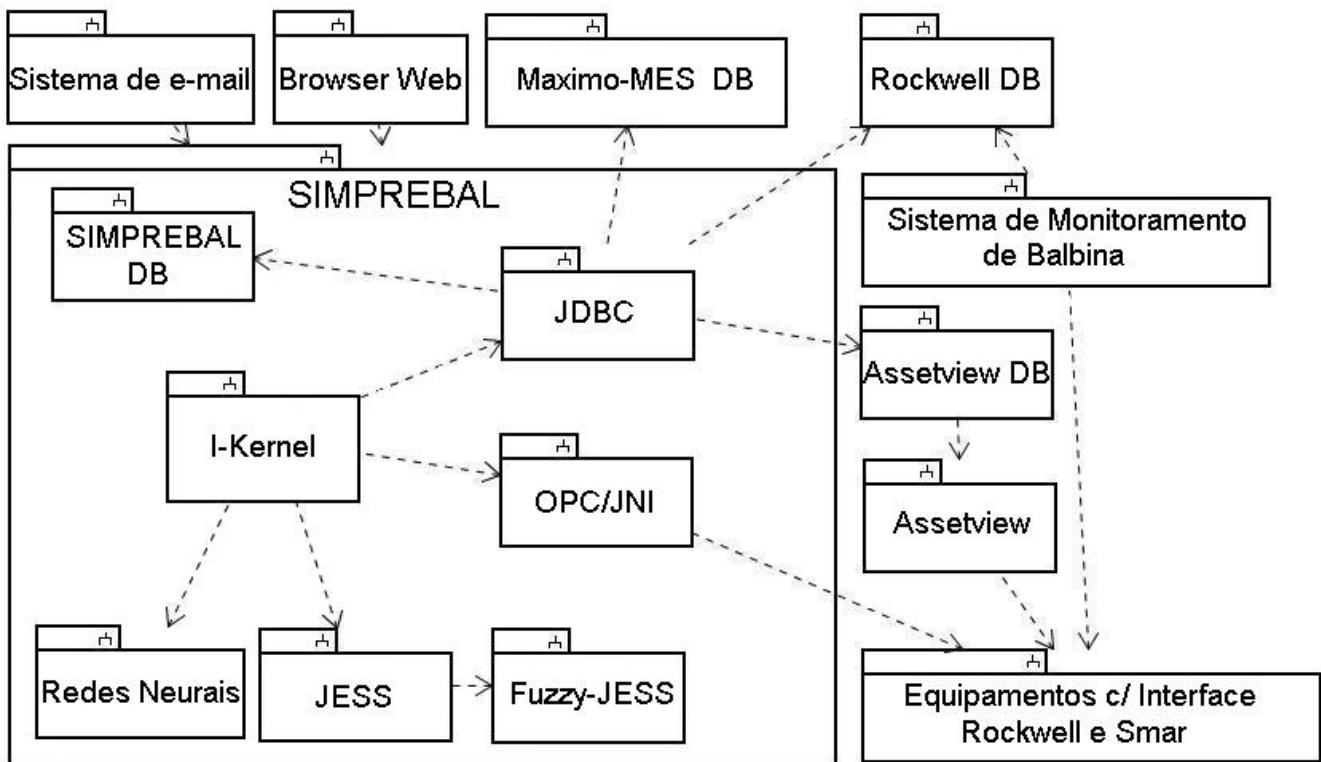


Fig. 3: Visão Geral do Projeto SIMPREBAL, demonstrando o I-Kernel e sua interação.

Prognósticos: Na tarefa de prognóstico, considera-se a avaliação da saúde do sistema, subsistemas, componentes, o escalonamento empregado (predição de uso) e a capacidade de raciocínio do modelo para prever o estado de saúde do equipamento. O sistema *SIMPREBAL* irá utilizar abordagem baseada em Sistemas Inteligentes enfocando a utilização de Sistemas Especialistas,

Lógica Fuzzy e Redes Neurais, para o qual temos que ter um modelo fuzzy, já que segundo Tanaka & Wang (2001), para se construir um sistema fuzzy, é preciso obter um modelo fuzzy de Takagi-Sugeno do sistema. Um prognóstico é sugerido a partir das informações dos módulos anteriores. Este pode ser baseado em modelos pré-estabelecidos ou feito considerando as características encontradas na avaliação de saúde.

Tomada de decisão: Integra informações necessárias para a tomada de decisão, considerando informação sobre a saúde dos equipamentos e para predizer a saúde de um sistema e/ou subsistema. O sistema SIMPREBAL irá realizar a tomada de decisão baseando-se na sua base de conhecimento gerada a partir da árvore de faltas/falhas, da árvore de sintomas, e pelas informações de inspeções realizadas pelo sistema TPM (Manutenção Produtiva Total), segundo Wyrebski (1998) hoje é considerada uma metodologia que auxilia na redução de ocorrência de falhas em instalações de máquinas e equipamentos nas empresas. Nesse módulo o sistema inteligente apresenta algumas sugestões para a solução do problema, cabe ao usuário tomar a decisão mais aplicada.

Apresentação: Suporta a apresentação de informação para o controle das entradas dos usuários de sistemas (por exemplo, manutenção e operação). As saídas incluirão qualquer informação produzida pelas camadas mais baixas e as entradas incluirão qualquer informação requerida pelas camadas mais baixas. Fornece também a interface homem/máquina, a interface será desenvolvida para GUI (Interface Gráfica com o Usuário) baseada em browser (Netscape, Mozilla, Iexplore, entre outros) usando html, XML, javascript e applets (Java).

CONCLUSÃO

Com a proposta apresentada no presente artigo, espera-se mostrar sugestões de manutenção para os operadores e o pessoal de manutenção antes que o equipamento apresente problema. Esse processo de manutenção é conhecido como manutenção preditiva e é baseada em confiabilidade. O sistema SIMPREBAL será desenvolvido usando técnicas de IA (Inteligência Artificial) como regras de produção, Lógica Fuzzy e redes neurais, o que facilitará o processo de tomada de decisão, visto que um dos objetivos do sistema é avaliar e monitorar o equipamento para oferecer prognósticos confiáveis. Permitindo dessa maneira que o operador possa tomar decisões de manutenção com base em informações geradas pelo sistema. O Sistema Inteligente SIMPREBAL auxiliará na tomada de decisão sugerindo prognósticos, para que a manutenção do equipamento possa ocorrer antes que a falha no mesmo ocorra. Isto implica em ganho de tempo e maior confiabilidade do sistema da usina. Vale ressaltar que a metodologia aqui proposta ainda está em fase de elaboração, resultados concretos serão obtidos posteriormente.

AGRADECIMENTOS

Agradecemos à Eletronorte, Manaus Energia e ao Eng. Antonio Araújo da Eletronorte pelas informações cedidas que foram de extrema importância para o desenvolvimento do presente trabalho.

REFERÊNCIAS

- Buschman, Frank, *“Oriented Software Architecture A System of Patterns”*, England; 1996
- Fleming, P. V.; França, S. R.R. O., *“Considerações Sobre a Implementação Conjunta de TPM e MCC na Indústria de Processos”*, Abraman 12º Congresso Brasileiro de Manutenção: TT044, São Paulo, (1997).
- Smar, *“Equipamentos de Campo série 302 Foundation. Manual de instalação, operação e manutenção”*, (2001)
- Smar, *“Manual de instruções dos blocos funcionais Fieldbus Foundation”*, (2005).
- Tanaka, K. Wang, H.O., *“Fuzzy control systems design and analysis a linear matrix inequality approach”*, New York: John Wiley & Sons (2001).
- Wyrebski, J., *“Manutenção Produtiva Total um Modelo Adaptado”*, Dissertação (Mestrado em Engenharia de Produção) Programa de Pós-Graduação em Engenharia de Produção, UFSC, (1998).

Anexo 2
Paper Submetido e ACEITADO no COBEM2007

DIFFERENT CONTROL STRATEGIES USED IN DIDACTIC PLANT PD-3 OF SMAR THROUGH OPC TECHNOLOGY

Edgar J. Amaya Simeón, edgar.amaya@gmail.com

Alberto José Álvares, alvares@AlvaresTech.com

Victor Rafael R Celestino, cvictor@uol.com.br

Carlos Frederico Maciel Silveira, fredericofurlan@yahoo.com.br

Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Inovação em Automação Industrial (GIAI), CEP 70910-900, Brasília, DF, Brazil

Abstract. *In this work, three different control strategies, neural network, fuzzy logic and PID (Proportional Integral Derivative), were implemented in Matlab, acting on a Smar PD3 didactic plant by means of OPC (OLE - Object Linking and Embedding - for Process Control) technology. The PD3 plant employs Foundation Fieldbus protocol and configuration tools of Smar System 302. The PD3 process chosen for testing control strategies consists to control the temperature on the mix water tank, which has a constant flow input of constant temperature hot water and an input consisting of ambient temperature water controlled by a pneumatic valve. The PID control was developed using differences equation and adjusting the parameters using a Ziegler- Nichols method. The Neural Network, that has a structure (3-20-1) capable to learn the dynamic of the PID controller, was trained using the values saved in vectors from the response of the PID controller. The training was accomplished using a backpropagation method that is least Square Method (LMS). The LMS method, search the minimum valor of the function square error. The Fuzzy Logic Controller was developed using the toolbox of the Matlab, membership function was edited in the FIS editor. In addition to the controllers a supervisory was developed using GUI of Matlab, acting on PD3 plant independently of SCADA Process View (Smar System 302). Controllers obtained satisfactory results for the combinations of input variables values, the Neural Network and PID controller have similiar response actuation and the Fuzzy Logic Controller has a less accuracy response, in other words the results highlight differences in performance among control strategies.*

Keywords: *Control Systems, Neural Networks, Fuzzy Logic, PID, Fieldbus Foundation.*

1. INTRODUCTION

In this work, the temperature control level in the mix water tank of Smar PD-3 didactic plant was executed, acting through different control strategies, implemented in Matlab. Plant access was achieved using OPC technology (OLE - Object Linking and Embedding - for Process Control). The plant allows the implementation of several control loops, by using industrial devices with sensing, actuating and control functions. These devices are directly managed through a Fieldbus Foundation protocol. PD3 Plant also includes a System 302 software suite.

The objective of this work was to implement and compare three different control strategies, Neural Network, Fuzzy Logic and PID (Proportional Integral Derivative), to communicate with the Fieldbus Foundation Smar instruments located in the PD3 didactic plant, using the Matlab opc toolbox that use OPC technology (OLE - Object Linking and Embedding - for Process Control), also to develop the supervisory system in order to have the HMI (Human Machine Interface).

The rest of this paper is organized as follows: Section 2 presents the description of the PD3 didactic plant; Section 3 presents the three different control strategies and describes the supervisory system; Section 4 describes the tests, results and discussions; and finally Section 5 concludes the paper.

2. SMAR PD3 DIDACTIC PLANT

The Smar PD3 didactic plant aim is to demonstrate didactically the operation of several control loops, using the same equipment and configuration tools (System 302 software suite), developed for the application in industrial control. With a compact arrangement, this plant makes accessible to instructors and students all components in the loop, allowing not only the observation of its behavior, but also its manipulation (Smar, 2004).

In the implementation of such loops, the same characteristics and situations found by instrumentation professionals are provided, with the resources of advanced technology available in the market. Besides the proposed control loops, others can be generated from the same physical structure assembled, without the need of any mechanical modification, but only by modifying the configuration of its devices through functional blocks (Smar, 2005).

The plant applies Fieldbus Foundation technology, an industrial communication bus used to connect field devices such as sensors, actuators, indicators and controllers, which allow the control of temperature, flow and associated levels, helping in industrial instrumentation and control systems learning.

The Smar PD3 didactic plant, presented in the Fig. 1, is monitored and operated by a station, constituted of a PC microcomputer and supervisory software, which executes equipment data acquisition and presents those by screen ani-

mations. It also allows actuating directly on registries, modifying equipment internal values and control loop operational modes.



Figure 1. SMAR PD3 Didactic Plant.

Figure 2 show the PD3 didactic plant contains two flow loop: The flow of the loop 1 is generated by pump 1, from water reservoir tank to tank 1 (hot water tank), all the hot water is transferred to the mix water tank. The Flow measurement is obtained by a Fieldbus LD302D differential pressure transducer FIT-31. The flow control is made by a Fieldbus FY302 positioner acting through a globe type valve FY-31. Temperature measure is obtained by a Fieldbus TT302 temperature transducer TIT-31. The resistance current control is made by Fielbus FI302 converter Fielbus Foundation for current acting in the static converter FY-31.

The TIC-31 is a single PID controller, that's located in the FY-31, is commissioned to control the water temperature in the tank 1 (hot water tank) acting in the resistance current.

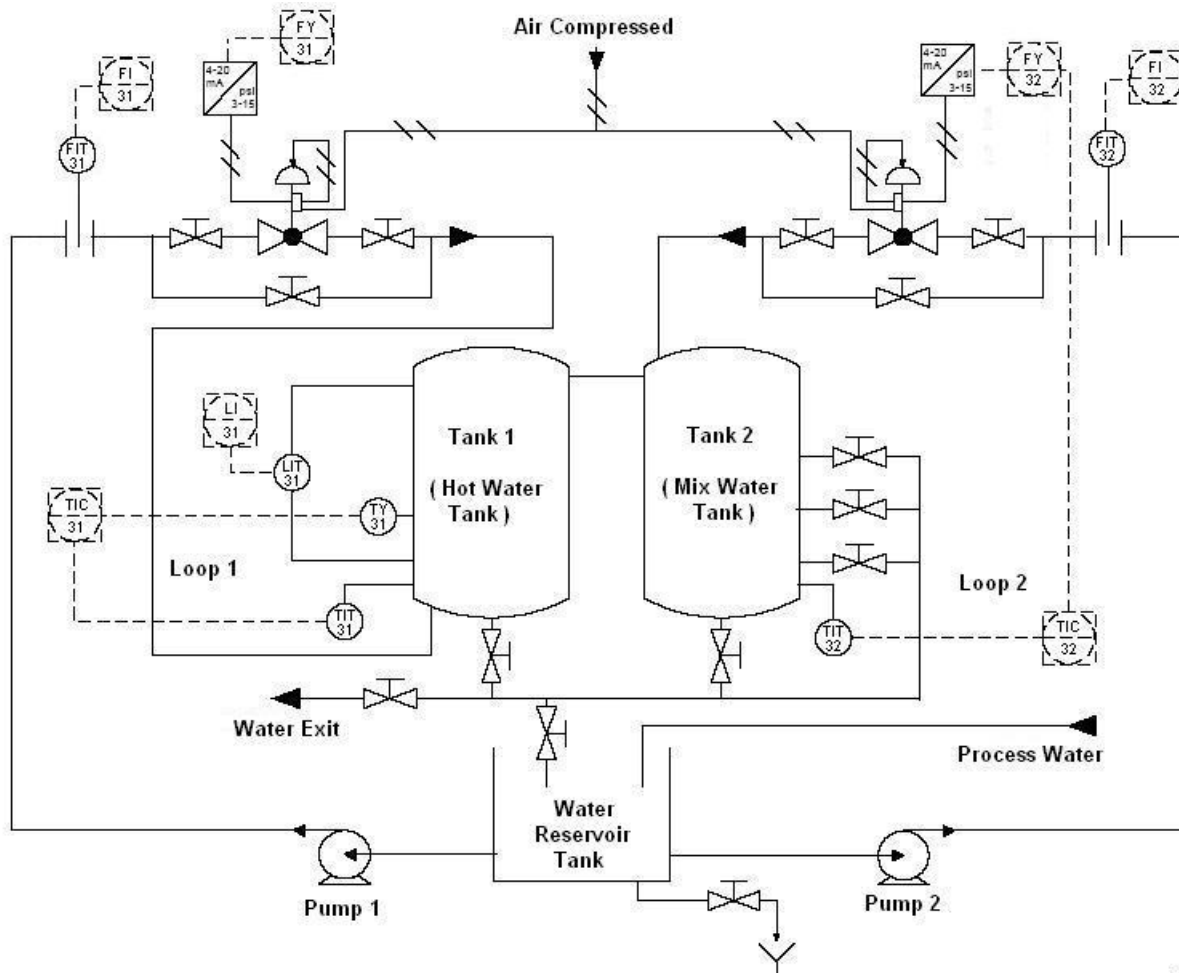


Figure 2. Controls Loops in the Smar PD3 Didactic Plant.

The flow of the loop 2 is generated by pump 2, from water reservoir tank to tank n 2 (mix water tank). Flow measurement is obtained by a Fieldbus LD302D differential pressure transducer FIT-32. Flow control is made by a Fieldbus FY302 positioner acting through a globe type valve FY-32. Temperature measure is obtained by a Fieldbus TT302 temperature transducer TIT-32.

The TIC-32 can be any of the control strategies implemented in Matlab like Neural Network, Fuzzy Logic or PID controller, that's located in the micro computer, is commissioned to control the water temperature in the tank 2 (mix water tank) acting in the control valve that control the flow of cold water.

3. CONTROL STRATEGIES

3.1 Artificial Neural Networks (ANN)

The need to control systems and processes exist since remote times, while methodologies to enhance a ANN training are recent (Bakshi, 1993), (Qin, 1992), (Leonard, 1992). The analysis and design of neural controllers is not trivial, due to the fact that ANN is not directly modeled by differential equations, what preclude the use of classical methods for analysis and design.

In general, it is difficult to obtain general analytical expressions due to the variety of network structures and activation functions which can be used. Most frequent modes to use neural networks in system control are: obtaining system inverse model through learning, behavior mapping of a known controller in the network, adaptation characteristics learning or reference models. Thus, online training of neural networks, until recently, demanded a significant processing speed and required very efficient convergence methods, what also demands large computational processing. With technological advances and creation of new optimization methods, online training became possible, create a neural networks applications, where a large number of practical applications became possible.

ANN are generic nonlinear approximators. Neither the structure nor the parameters are needed. If training data is rich in information, if neurons number are sufficient, and if there is an adequate training algorithm, a plausible solution can be expected. A global optimum solution is seldom needed for practical problems. When a large range of operations is expected, it is more difficult for just one ANN to have a good performance in the entire range. In this article we propose an ANN trained for a defined operation range.

In the Fig. 3 is show the artificial neuron model, each input receives a stimulus, which is weighted by synaptic weights. All inputs are summed, generating a response, which is later modulated by a mathematical function (activation function).

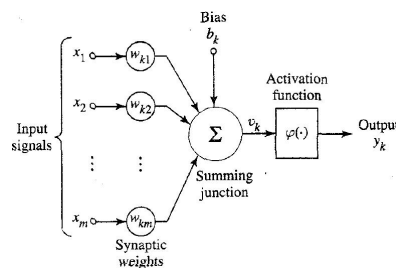


Figure 3. No Linear Model of a Neuron.

Neural networks are composed of several elements connected among them in some way, allowing its parallel operation (Gabriel, 1996). The way neurons are organized defines the network architecture. The architecture used is a perceptron type with multiple layers as depicted in the Fig. 4.

In the Eq. (1) is illustrated the operation performed by an artificial neuron when the weights vector $[w_0 \dots w_n]$ is multiplied with the input vector $[x_0 \dots x_n]$, each element can have a bias b_k plus an extra constant input $b_k = + 1$. The linear output y_k is finally converted in a nonlinear function, as a relay, sigmoid, hyperbolic tangent, etc. The relay functions are appropriated for binary systems, while the sigmoid an hyperbolic tangent functions can be employed for both continuous and binary systems (Lotufo, 2007).

$$y_k = f\left(\sum_{k=0}^n x_k w_k + b_k\right) \quad (1)$$

The neural network structure: 3 inputs, 20 neurons in the hidden layer and 1 neuron in the output layer. The activation function in the hidden layers is the hyperbolic tangent function and in the output layer is the sigmoid function.

The backpropagation method, utilized in this work, belongs to the family of Least Mean Square (LMS) methods, in which the minimum value of the quadratic error function is sought, based in the gradient method. The training algorithm basically consists in correcting in the output layer and in the hidden layers synaptic weights, on the basis of the error

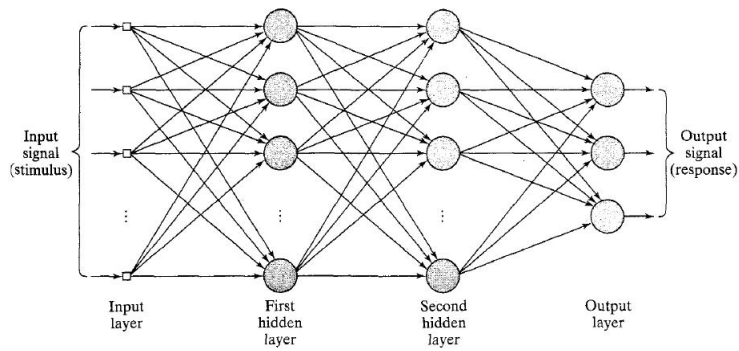


Figure 4. Architectural graph of a multilayer perceptron with two hidden layers.

occurred in each output of the neural network. Weight correction is done by means of the minimum quadratic means method, aiming to find values for the weights which minimize the network output error (Haykin, 99).

For the training of the neural network controller, input and output values obtained from the PID controller response were used according to the Fig. 5.

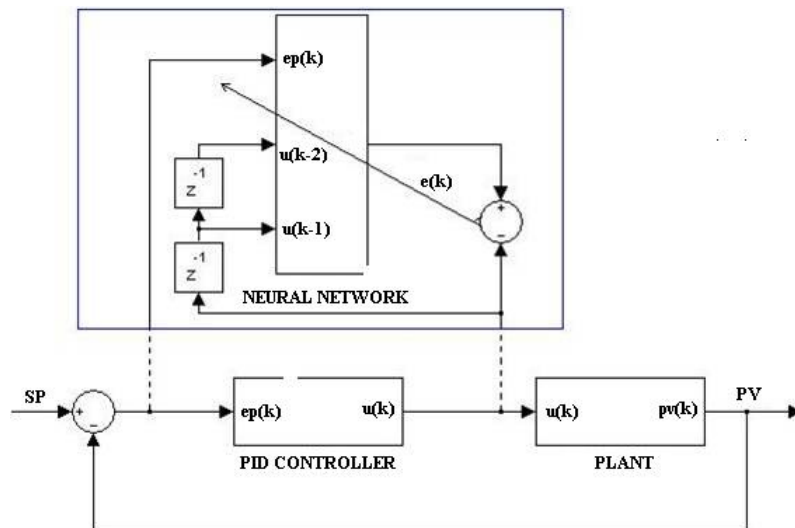


Figure 5. Neural Network Training Model.

The neural network controller showed in the Fig. 6, that has as inputs the error $e(k)$ signal and the past values of the control signal $u(k-1)$ and $u(k-2)$, make it follow a reference signal $r(k)$. The neural network controller generate control signals $u(k)$ act on plant input for control purposes.

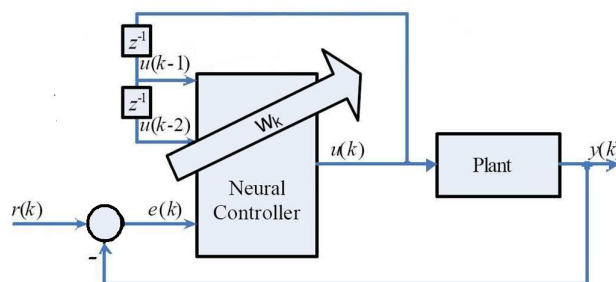


Figure 6. Neural Network Controller.

3.2 PID (proportional integral derivative)

The utility of PID controllers is its general applicability to the majority of control systems. In particular, when plant mathematical model is unknown, PID is the most useful control method. Although in many applications, PID can't provide optimal control. PID control has the following inputs variables, SP (Set Point) and CV (Controlled Variable). The difference between SP value and CV value constitutes the error; this error is the input for PID controller, which will generate a control action seeking to error zero. After some actuating time, the SP value is attained.

Error processing happens in accordance with schematics presented in the Fig. 7. The $e(t)$ term represents error as a time function, the control signal is the result of the sum of the proportional, integral and derivative terms, acting in the process to be controlled, finally, Output term represents Controlled variable.

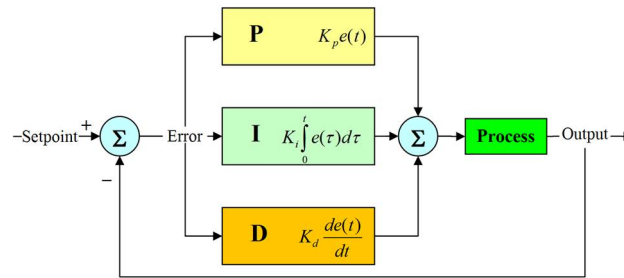


Figure 7. PID controller basic schematics.

Control action over process is a result of summing individual actions of the three components. The first, on top of the Fig. 7, has as individual error action multiplied by the constant K_p . The second, the of integral error multiplied by the constant K_i . And the third, the multiplied of derivative error by the constant K_d . Considering K_i as K_p/T_i , K_d as $K_p \cdot T_d$. The transfer function of the PID controlled in the frequency domain is expressed in the Eq. (2), the same transfer function in the discrete domain showed in the Eq. (3). In order to implement the PID controller in a program computer it's necessary to represent in its discrete form showed in the Eq. (4).

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (2)$$

With application of z transform:

$$\frac{U(z)}{E(z)} = \left(K_p + \frac{K_p T z}{T_i (z-1)} + \frac{K_p T_d (z-1)}{T z} \right) \quad (3)$$

Given that multiplication by $z-1$ represents a delay of one sampling interval (T), the following difference equation is obtained for PID controller in its digital form:

$$u(k) = \left(K_p + \frac{K_p T}{T_i} + \frac{K_p T_d}{T} \right) e(k) + K_p T_d T e(k-1) + \frac{K_p u(k-1)}{T_i} \quad (4)$$

Equation (4) applied in a recursive manner, was the core of the remote PID controller algorithm proposed in this work. The algorithm was implemented in Matlab in a remote client computer, which access to the PD3 didactic plant through OPC protocol.

For the tuning of the PID controller, a Ziegler-Nichols method was used, where an experimental adjustment of PID parameters was done. The method works in the following manner: A step function is placed as set point and PID is turned-on, with T_r equal to infinity and T_i equal to zero; K_p gain is adjusted until reaching marginal stability condition. This gain, de K_{pc} , is named critical gain. The resulting sinusoid period as marginal stable output is indicated by P_{pc} . The final K_p value is set as $0.6 K_{pc}$, T_i as $0.5 P_{pc}$, and T_d as $0.125 P_{pc}$ (Ogata, 2004). Using the algorithm above and applying a small empirical adjustment on top of values found, the following parameters were used for the implementation: $K_p = 3.7$; $T_i = 9$; $T_d = 1$. For all tests and results obtained, it was used $T = 2$ s.

3.3 Fuzzy Logic Controller

Fuzzy set theory and fuzzy logic establish the rules of a non-linear mapping. The use of fuzzy sets provides a basis for a systematic way for application of uncertain and indefinite models. Fuzzy control is based on a logical system called fuzzy logic. It is much closer in spirit to human thinking and natural language than classical logical systems. Nowadays, fuzzy logic is used in almost all sectors of industry and science. One of them is plant control. According to many

researchers, there are some reasons for the present popularity of fuzzy logic control. First of all, fuzzy logic can be easily applied for most applications in industry. Besides, it can deal with intrinsic uncertainties by changing controller parameters. Finally, it is appropriate for rapid applications. Therefore, a fuzzy logic system has been applied to industrial systems as a controller. Human experts prepare linguistic descriptions as fuzzy rules. These rules are obtained based on experiments of the process's step response, error signal and its time derivative (Cam, 2007).

Basically, fuzzy rules provide a convenient way for expressing control policy and domain knowledge. Furthermore, several linguistic variables might be involved in the antecedents (before then) and the conclusions (after then) of these rules. A fuzzy logic system mainly consists of three steps: fuzzification, fuzzy inference and defuzzification. In the fuzzification step, the real variables are translated into linguistic variables by using fuzzy set theory. In the fuzzy inference step, 'If-Then' rules that define the system behavior are evaluated. The defuzzification step translates the linguistic result obtained from the fuzzy inference into a real value by using the rule base provided.

The fuzzy controller applied in this work is composed of a set of inferencing rules of a type such as If <expression> AND <expression> Then <result>, which define control actions as a function of the several value ranges, which plant input variables can assume. 225 inference rules were implemented.

Using the Fuzzy Logic Toolbox (Jang and Gulley, 1999), the controller was implemented using FIS editor, a Matlab toolbox. The controller was built based on Mamdani model and max-min composition was used for operations with fuzzy numbers: min operator for AND connector, max for OR connector, besides min operator for implying and max for aggregation. The centroid method was used for defuzzification.

The controller uses three input variables, which are extracted from plant output: water temperature in the hot water tank TIT-31, water temperature in the mix water tank TIT-32 and set point of the water temperature in the mix water tank TIT-32SP. The output is the open percent value of the control valve FY-32.

Input variables assume values: cold (C1), warm (W1, W2, W3), hot (H1, H2, H3), and very hot (VH1, VH2, VH3). The membership functions have the form presented in the Fig. 8.

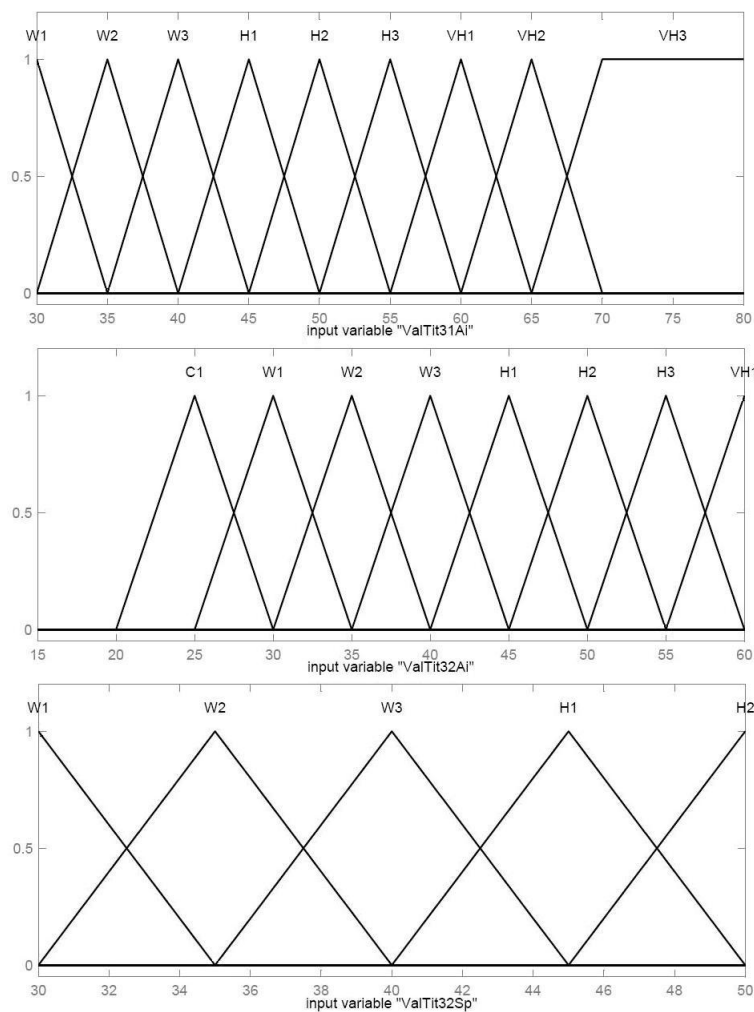


Figure 8. Input Membership Functions.

Output variable assume values: closed, 1/10, 1/8, 1/6, 1/5, 1/4, 3/10, 1/3, 3/8, 2/5, 1/2, 5/8, 2/3, 3/4, 5/6 and open. Its membership function showed in the Fig. 9 was discretized based on the plant linearized model.

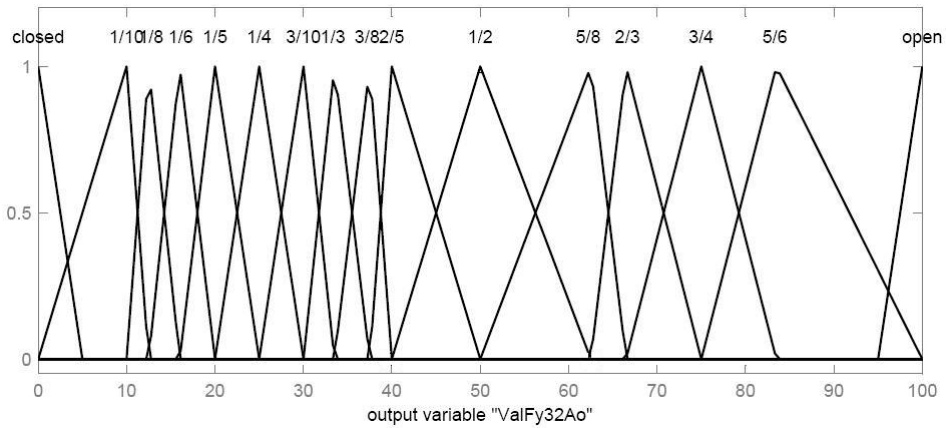


Figure 9. Output Membership Function

Some of the rules implemented can be observed in the Tab. 1, rules havind as inputs TIT-31, TIT-32 and TIT-32SP, the out of the rules is FY-32. The same rules is showed in the three dimensional view surface rules as illustrated in the Fig. 10.

Table 1. Inference Rules.

TIT-31	TIT-32	TIT-32SP	FY-32
VH	C	H	0
VH	W	H	1/4
VH	H	H	3/4
H	C	H	0
H	W	H	1/4
H	H	H	3/4
VH	C	W	3/4
VH	W	W	1
H	C	W	1/4
H	W	W	3/4

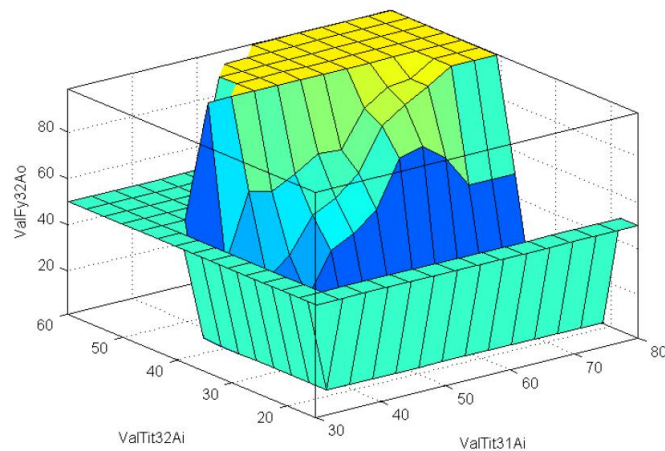


Figure 10. Surface Rules.

3.4 Supervisory System

In order to have simultaneous access to as much information as possible in the least time achievable, it was necessary to centralize information through the development of an appropriate HMI (Human machine interface). Figure 11. illustrate the Supervisory System that was implemented in Matlab using GUIDE and OPC toolbox (Matlab, 2006), using two active X for displaying process variables like a hot water tank level and the plot of process variables.

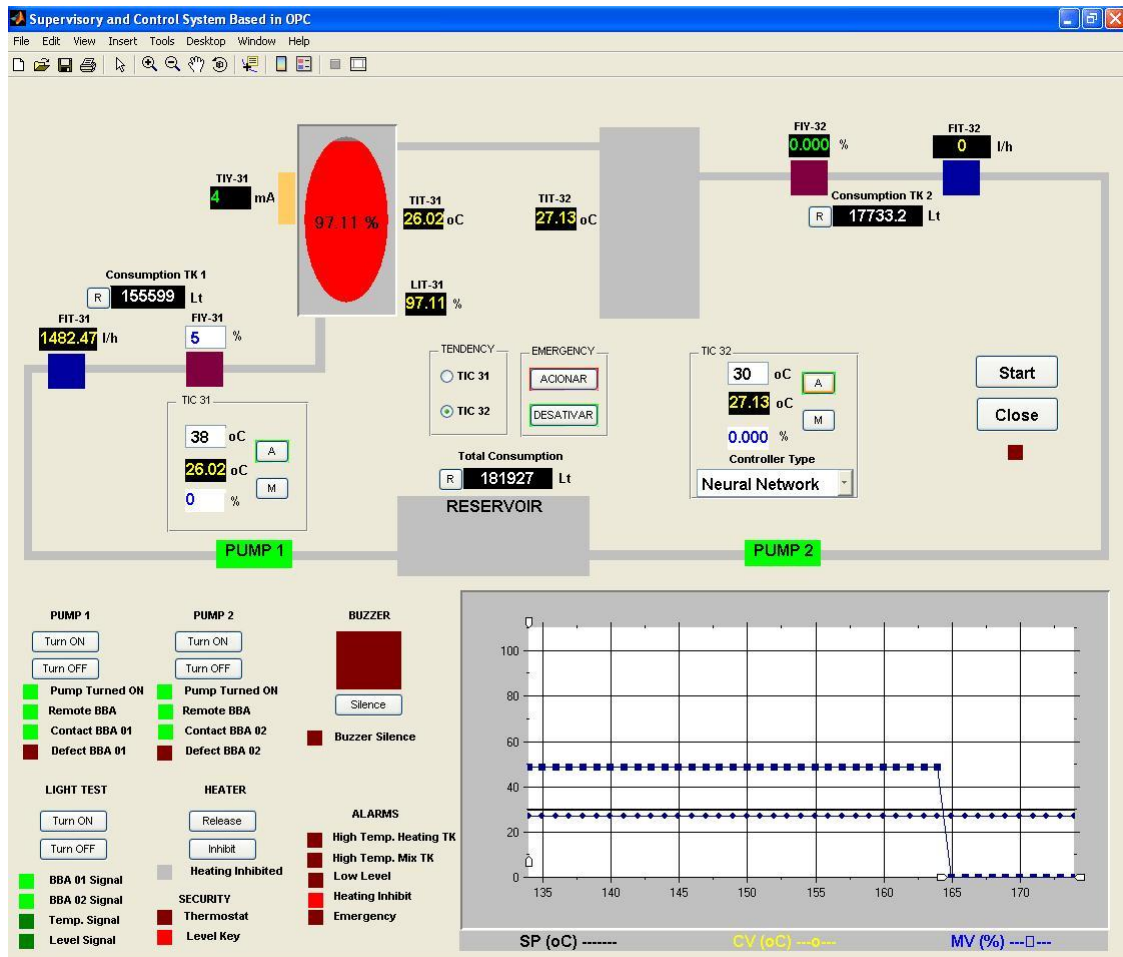


Figure 11. OPC Supervisory System developed in Matlab.

The supervisory system contain the controllers algorithms implemented, and allow the complete control of the plant like the turned on of the pumps, heating, buzzer, control valves; switch the control strategies. OPC communication protocol was used for connecting the client to the server that contain all the process data. The communication occurs in a client server architecture, the server computer stores the two Smar OPC servers, Smar.DfiOleServer.0 and Smar.DF65Server.1. In the Client Side is located the supervisory system.

4. TESTS, RESULTS AND DISCUSSIONS

Defined the test for functional analyzes of these controllers, the temperature of the hot water tank with its automatic temperature control loop around 38°C and the temperature of the cold water from the reservoir around 26°C, the set point of the controllers were in 32°C, the process start with the water temperature in 38°C, the mission of the controllers is to arrive to 32°C and maintain in this value.

The Neural Network controller with its weights learned starts the control loop, its action is observed in the Fig. 12, the PID controller with its parameters syntonized starts and the effected of the control action is looking in the Fig. 13, and the response of the Fuzzy Logic Controller is showed in the Fig. 14.

Table 2. Comparison of the three Control Strategies Response.

	Neural Network	PID	Fuzzy Logic
Overshoot[%]	1.75	0.78	8.6
Overshoot Time[s]	34	54	110
Settling Time[s]	90	85	>120
Rise Time[s]	20.4	21.3	20

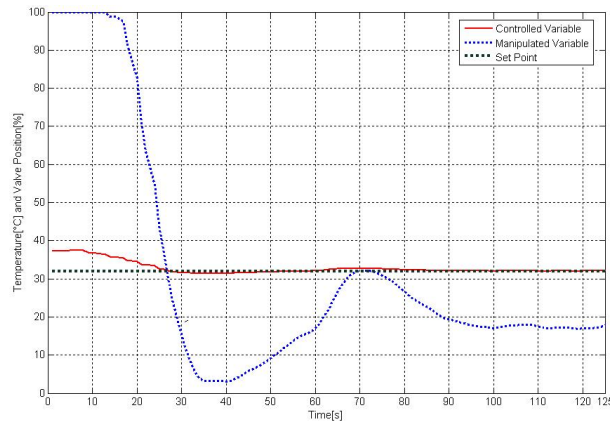


Figure 12. Neural Network Controller response

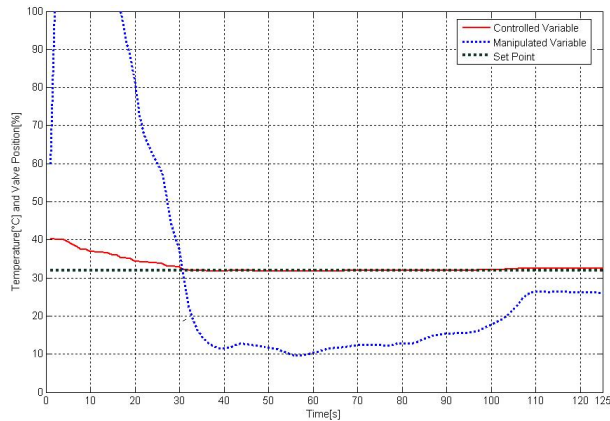


Figure 13. PID Controller response

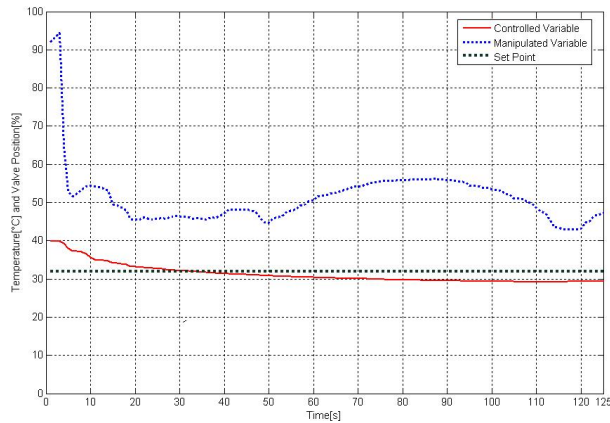


Figure 14. Fuzzy Logic Controller response

5. CONCLUSIONS

In this paper, a comparative study was presented using a neural network, fuzzy logic and a conventional proportional integral derivative controller, PID. The controllers were applied to a didactic plant of Smar. In the tests were observed that the Fuzzy Logic controller arrive to the set point slowly because were considered few rules and membership function. The neural network envelop learning and training in this case was necessary to have as a teacher to the PID controller. The PID require mathematical equation and tests to tuning the constants K_p , K_i and K_d . The fuzzy logic controller demand human experience that know how to interact with the plant dynamic. The employed industrial protocol Fieldbus Foundation to communicate the supervisory system and Smar instrumentation permit to control the plant remotely. The Matlab OPC and Fuzzy Toolbox (Matlab, 2006) was helpful in the development of the experiments. Future work includes the increasing dynamic to the fuzzy controller and to made neuro fuzzy controller in order to had better response behavior.

6. ACKNOWLEDGEMENTS

We acknowledge the support of the Eletronorte and Manaus Energia provided by the Reserch and Development Program under contract number 4500052325 project number 128 "Modernization of Processes Automation Area of the Hydroelectric power stations of Balbina and Samuel", that has as a technical responsible to the Prof. Alberto José Álvares of the UnB.

7. REFERENCES

- Bakshi, B.R. and Stephanopoulos, G., 1993, "Wave-Net: A Multiresolution, Hierachial Neural Network with Localized Learning," in *AIChe J.*, Vol. 39, No. 1, 57-8 1.
- Cam, E., 2007, "Application of fuzzy logic for load frequency control of hydroelectrical power plants", *Energy Conversion Manage* 48 (4), pp. 1281-1288.
- Chen, F.C. and Khalil, H.K., 1995, "Adaptive Control of a Class of Nonlinear Discrete-Time Systems Using Neural Networks", *IEEE Trans. On Automat. Control*, 40, pp. 791-801.
- Gabriel, O., 1996, "Um Esquema de Controle Adaptativo Neural com Treinamento em TempoReal", *Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte-UFRN. Natal, Rio Grande do Norte. Brasil.*
- Haykin, S., 1999, "Neural Networks a Comprehensive Foundation", Second Edition, pp. 183-197.
- Hunt, K.J. and Sbarbaro, D., 1991, "Neural Networks for nonlinear internal model control", *IEE Proc., Part D*, 138, pp. 431-438.
- Jang, J.S.R. and Gulley, N., 1999, "Matlab fuzzy logic toolbox User's Guide. v.1", Natick: The Mathworks Inc, 235 p.
- Leonard J.A., Kramer, M.A. and Ungar, L.H., 1992, "A Neural Network Architecture That Computes its Own Reliability," in *Comp. and Chem. Eng.*, Vol. 16, No. 9, 1992,819-835.
- Lotufo A.D.P., Lopes M.L.M. and Minussia C.R., 2007, "Sensitivity analysis by neural networks applied to power systems transient stability", *Electric Power Systems Research* 77 (7), pp. 730-738.
- Matlab, 2006, "OPC Toolbox for use with Matlab and Simulink User's Guide. v.2.", Natick: The Mathworks Inc, 373 p.
- Ogata, K., 2004, "Engenharia de Controle Moderno". Ed. LTC, Rio de Janeiro, 710 p.
- Qin, S.J. and McAvoy, T.J., 1992, "Nonlinear PLS Modeling Using Neural Networks," in *Comp. and Chem. Eng.*, Vol. 16, NO. 4, 1992,379-391.
- Smar, 2004, "User's manual Didatic Plant III", 127 p.
- Smar, 2005, "Function Blocks Instruction Manual Fieldbus Foundation", 334 p.

Anexo 3
Paper Submetido e ACEITADO no COBEM2007

AN INTELLIGENT KERNEL FOR THE MAINTENANCE SYSTEM OF A HYDROELECTRIC POWER PLANT

Edgar J. Amaya Simeón, edgar.amaya@gmail.com

Alberto José Álvares, alvares@AlvaresTech.com

Rosimarci Pacheco Tonaco, rosimarci@hotmail.com

Rodrigo de Queiroz Souza, rodrigoqsouza@hotmail.com

Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Inovação em Automação Industrial (GIAI), CEP 70910-900, Brasília, DF, Brasil

Ricardo Ribeiro Gudwin, gudwin@dca.fee.unicamp.br

Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica e de Computação, Departamento de Engenharia de Computação e Automação Industrial, Campinas, São Paulo, Brasil.

Abstract. *In this paper, we present the conceptualization of an intelligent system to be used on the predictive maintenance of a hydroelectric power plant. It is a real application, being developed in the context of the project "Modernization Processes of Automation Area of the Hydroelectric power stations of Balbina and Samuel", with the support of Eletronorte. The work described in this paper corresponds to the development of an intelligent kernel, which is planned to be the heart of a major methodology, called SIMPREBAL (Predictive Maintenance System of Balbina), to be implanted in the plant of Balbina, near the city of Manaus in Brazil. The overall specification and design of this intelligent kernel is described here using UML (Unified Modeling Language) diagram. This intelligent kernel, called here I-kernel, will be providing expert systems functionality, besides fuzzy logic rules processing and neural networks in an operational cycle where data is collected from equipments and databases, and will be used to both give support to the operational team and to the maintenance team at the plant. I-Kernel will get the data from supervision system by means of both JDBC access to databases and JNI (Java Native Interface) getting data from the OPC (OLE - Object Linking and Embedding - for Process Control) server. In order to provide expert systems and fuzzy logic functionality to the system, we will be using the tools JESS (Java Expert Systems Shell) and Fuzzy-JESS. The SIMPREBAL methodology is based on RCM (Reliability centered maintenance) concepts, being used to analyze the manners and effects of fails on the Hydraulic Generator Units of Balbina, focusing its analysis in the Turbine system. In this work, the main details of the I-Kernel system are detailed. The concept of an intelligent kernel is that will be useful for the construction of similar applications in the future.*

Keywords: *Unified Modeling Language, Predictive Maintenance, Expert Systems, Fieldbus Foundation.*

1. INTRODUCTION

The motivation for this work started with the development of the SIMPREBAL (Alvares, 2006) methodology, under the context of the project "Modernization Processes of Automation Area of the Hydroelectric power stations of Balbina and Samuel", with the support of Eletronorte. The main goal was to implement ideas from RCM (Reliability Centered Maintenance) (Smith, 1993)(Mobray, 1997) and implant them in the plant of Balbina, a hydroelectric power plant located near the city of Manaus in Brazil. The SIMPREBAL project conceived an intelligent system collecting data directly from equipments in the power plant, storing this data in a database, and making use of all this information in order to predict maintenance interventions in order to minimize the impact of (future) failures in the operation of the plant. The main strategy for achieving this goal is by means of the prediction of future failures, based on the measuring of some process variables and analyzing their operational conditions in order to detect anomalous situations that may indicate that a potential failure is possible, in a near future. This may allow for the maintenance team to anticipate this failure and take the appropriate actions in order to minimize the problems that this future failure may impose in the operation of the power plant. In order to develop the SIMPREBAL methodology, there was a need for an intelligent kernel, a software component responsible for getting the data both directly from the equipments used in the power plant, and from databases with historical data, processing all this data in order to analyze them, making predictions, and an integration with the maintenance system of the power plant, so the predicted anomalies may take their role in the organizational process of the maintenance team. From this demand, there appeared the main subject theme of this paper, the I-Kernel Subsystem. We conceived I-Kernel to be a software component, to be developed in Java language, which should be integrated to the main computer system in Balbina, providing the way for implementing SIMPREBAL in the operational (supervision and control) and maintenance organizational process already working in the plant of Balbina.

The role of I-Kernel was, then, to provide the general functionalities of expert system, fuzzy rule based system and neural networks, which were required in order to generate the predictive analysis and decision-making demanded by the SIMPREBAL methodology. In this paper, we detail how we conceived this main component and how it can be integrated to other sub-systems in order to allow for the SIMPREBAL methodology to be applied in a power plant. This will be

developed in the next sections, by means of a detailed object-oriented specification and design.

The OO (object-oriented) paradigm offers several benefits, such as encapsulation, abstraction, and reusability to improve the quality of software. The UML (Unified Modeling Language) has emerged as the standard for analysis and design of OO systems. UML provides a variety of diagramming notations for capturing design information from different perspectives. In recent years, researchers have realized the potential of UML models as a source of information in software development (Sommerville 2003).

The rest of this paper is organized as follows: Section 2 presents the I-kernel subsystem requirements as one of the main subsystems of the SIMPREBAL system; Section 3 describes the general SIMPREBAL system configuration, showing how the I-Kernel should be integrated to other sub-systems; Section 4 presents the design of the I-kernel application ; Section 5 presents the Configuration and Monitoring Application Tool that works together with the I-kernel application; and finally Section 6 concludes the paper.

2. I-KERNEL SUBSYSTEM SPECIFICATION

The I-Kernel subsystem is a software component, to be developed in Java language, responsible for the data acquisition from databases and process equipments through OPC, and its processing through conventional rules, fuzzy logic and neural network, in N layers. As much of the functionalities required for I-Kernel are already available as out-of-thebox software components, instead of developing the whole system from scratch, our proposal is to use the available components and to integrate them in order to generate I-Kernel. In the same way, I-Kernel, as a software component, may be reused in future projects that demand intelligent processing solutions involving the processing of rules, fuzzy logic or neural networks.

The components to be used are the following: JESS - Java Expert System Shell, developed by Sandia National Laboratories, responsible by the rule processing and main inference machine to the expert system part of I-Kernel; Fuzzy-JESS, developed by the NRC Institute for Information Technology, responsible for the processing of fuzzy rules, in a way similar to JESS; JDBC responsible for the access to the databases and JNI, responsible for the access to the OPC servers which give direct access to the equipments. The I-Kernel application needs to access different databases, in order to integrate to the computational environment of Balbina's power plant. This includes the database for the Rockwell's Monitoring System, Smar's Asset Management System - Assetview, IBM/MRO's Operating and Maintenance Management System MAXIMO-MES(Manufacturing Execution System), and also the own SIMPREBAL database which will be storing all the knowledge required by the different intelligent algorithms. The Neural Network module will be developed almost from scratch, based on some code available at UNICAMP, built according to the scientific literature. A general view of this architecture, in terms of a UML diagram, is available at Fig. 1.

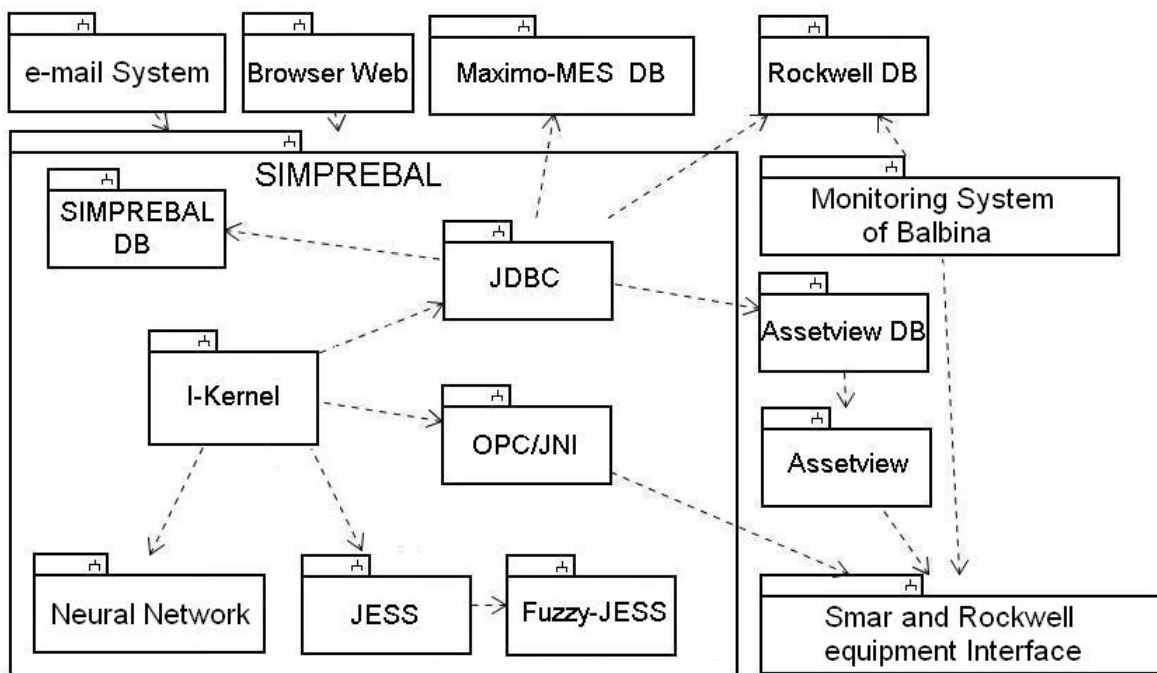


Figure 1. SIMPREBAL System Architecture and Interactions

According to (Sommerville, 2003), a useful way to structure the requirements is to split them into user requirements and system requirements.

2.1 User Requirements

The user requirements should capture the more general demands of system's users. Normally, these requirements are described in natural language and shouldn't enter too much in technical details. In this document we will be adopting this format recommended by Sommerville, listing the user requirements as a requirements list in natural language. The user requirements are divided into "Functional Requirements" as illustrated in Table 1, and Non-Functional Requirements listed in Table 2.

Table 1. Functional Requirements.

	Functional Requirement
FU1	The system should access the Plant's data, either from databases used by the Monitoring System of Balbina, or directly from the plant's equipments, by means of an OPC server, which should make available the on-line information from the equipments
FU2	The system should process those data in the form of: a rule-based expert system, a fuzzy logic inference engine, Neural Networks
FU3	The system should alert the user through e-mail messages when anomalous situations could be diagnosed
FU4	The system should alert the user through a visual alert, when anomalous situations could be diagnosed
FU5	The system should permit the edition of a synoptic, containing some variables to be monitored, chosen by the user and composing a possible presentation screen
FU6	The system should exhibit the on-line value of the variables being monitored, which were selected to compose a certain synoptic, showing them in a proper screen previously developed
FU7	The system should implement some learning mechanism, in such a way that the report of failures and previous defects could be used to prevent the appearance of new failures
FU8	The information processing should happen as a closed operational cycle, that should follow the following sequence: Verification of the Data to be acquired; Acquisition of the Data from Database; Acquisition of the Data through OPC; Temporary storage of all the data in JESS variables; [For each one of N possible layers: Processing of the rules through JESS, the Fuzzy Logic rules through Fuzzy-JESS, the data through Neural Network;] Updating of the Data in the Database; Updating of the Data through OPC.

Table 2. Not-Functional Requirements.

	Not-Functional Requirement
NFU1	The system should be developed in Java language
NFU2	The system rules should not be stored directly in the source-code, but should be editable and available externally in an editable text file
NFU3	The system should possess a web interface access, through which should be possible to the user to configure the system, to edit rules and parameters and to monitor the variables being processed by the system
NFU4	The system should be compatible to SQL generic databases, given that a JDBC driver for the database is available
NFU5	The system should use the JESS package as an inference engine
NFU6	The system should use the Fuzzy-JESS package to process the fuzzy logic rules.
NFU7	The system should be conceived such as classic rules, fuzzy logic rules and neural network may be used in an interchangeable way for each one of the SIMPREBAL processing layers (3-condition monitor, 4-health assessment, 5-prognostics, 6-decision support) according to the OSA-CBM (Open System Architecture for Condition Based Maintenance)

2.2 System Requirements

The system requirements will be detailed in a more technical way, using UML diagram. Following the Unified Development Process (Jacobson et.al. 1999), system requirements should be specified by means of use cases. In Fig. 2, we have an UML use-case diagram, which provides us a general overview of the system. The system is composed of two applications that are executed in an independent way: the I-Kernel application and the Configuration and Monitoring Application (Gudwin 2006).

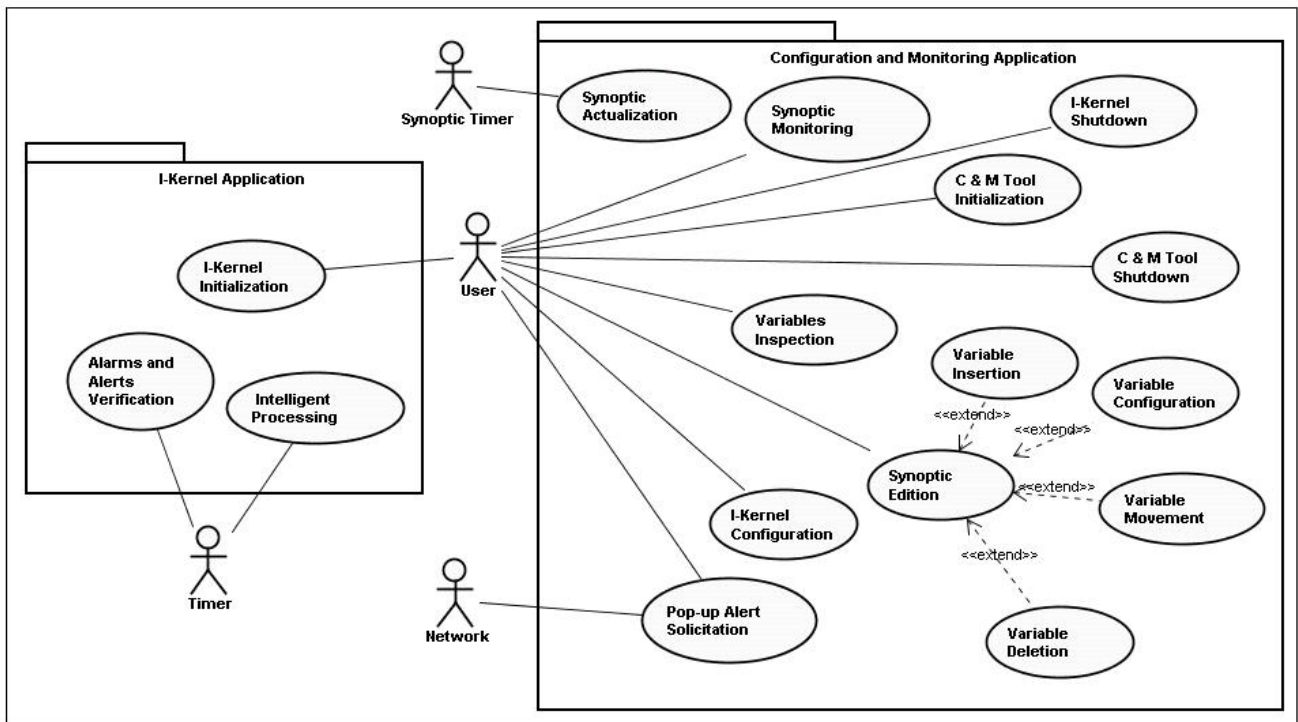


Figure 2. SIMPREBAL Use-Case Diagram

The overall use-case diagram represents an overview of the system’s functionality by pointing out who are the main users of the system and to which use-cases they take part. The whole system is partitioned in two sub-systems: The I-Kernel Application and the Configuration and Monitoring Application. The I-Kernel Application is responsible for the main operational cycle, where data is collected, processed and dispatched. The Configuration and Monitoring Application, as its name suggests, is responsible for configuring and monitoring I-Kernel.

We developed basically 16 use-cases: I-Kernel Initialization, Intelligent Processing, Alarms and Alerts Verification, I-Kernel Shutdown, C & M Tool Initialization, C & M Tool Shutdown, Synoptic Edition, Variable Insertion, Variable Deletion, Variable Movement, Variable Configuration, Synoptic Monitoring, Synoptic Actualization, Variables Inspection, I-Kernel Configuration, and Pop-up Alert Solicitation.

3. System Configuration

In Fig. 3 we present a UML deployment diagram, showing a possible configuration for the final system. This is not the only possible configuration, but it is a prototypical one. In the case of Fig. 3, there might be 1 client machine and 3 different server computers, where the different software modules should be installed. A different configuration might use only one server computer, where all the service modules are installed. The only restriction presented is that the ConfMonitToolApp module and the I-kernelApp module necessarily need to be installed in the same computer. This restriction is because the ConfMonitToolApp module is a java applet, which uses the network for communicate to the I-kernelApp module. For security restrictions, a java applet only can communicate with the same computer from where it was loaded. For that reason, it is necessary that the I-kernelApp module is installed in the same computer where the web server is installed. Besides the I-kernelApp and the ConfMonitToolApp, other services to be required are: a web server, a database compatible to JDBC and an OPC server to be accessed through JNI. The two software components which were developed were the I-kernelApp and ConfMonitToolApp. These applications should be available as JAR files which should be executed independently through a JVM (Java Virtual Machine). The I-KernelApp module is a standalone java application. The ConfMonitToolApp is a java applet, loaded from a website, and stored in the web server.

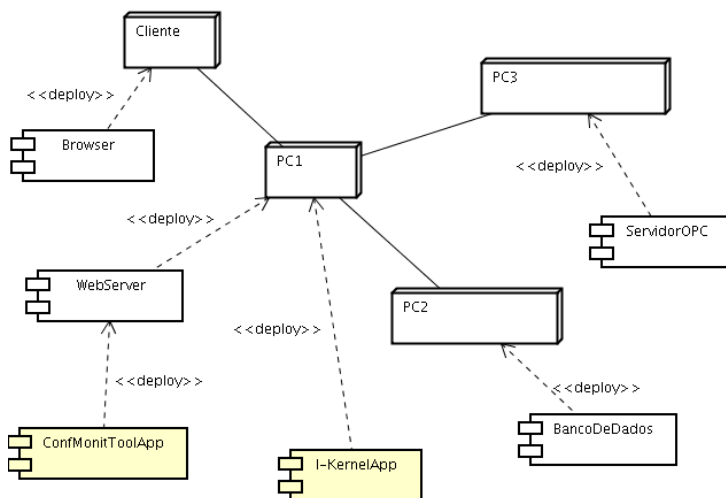


Figure 3. System Configuration

4. I-Kernel Application

The I-Kernel application is a server that doesn't have an user interface. In this way, the only thing that the user can do is to start the application. After that, a timer previously programmed sends periodic "ticks" that start each operational cycle of intelligent processing and make alarms and alerts verification. The details can be observed in Fig. 4.

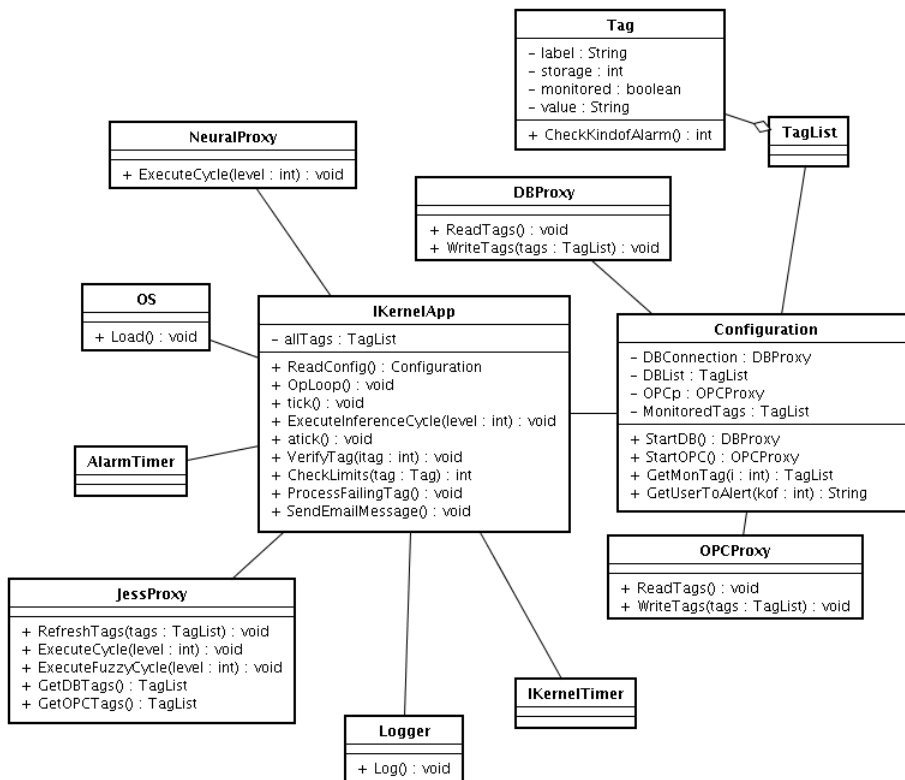


Figure 4. I-Kernel Application Use-Case Diagram

4.1 Use-Case 1: I-Kernel Initialization

In order to initialize the I-kernel application, the user asks the operational system to load the application. Once the application is loaded, it creates an IKernelApp object, which reads the configuration file, creates a new AlarmTimer, a new IKernelTimer, ask for the starting of the database proxy and to the OPC proxy, and after that creates the JESS proxy

(which should include the Fuzzy-JESS processing) and creates the neural networks proxy. Once created, the AlarmTimer and the IKernelTimers start doing "ticks", which will trigger the main operational cycle. This sequence is shown in Fig. 5.

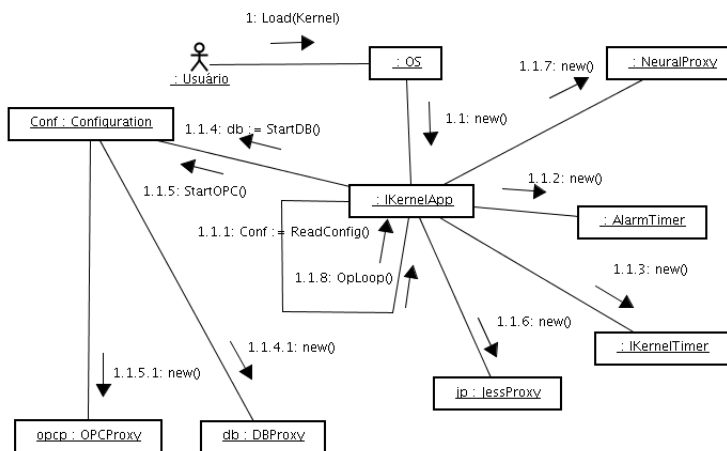


Figure 5. I-Kernel Application Initialization Collaboration Diagram

4.2 Use-Case 2: Intelligent Processing

The intelligent processing is the heart of the application. It starts with a "tick" from the IKernelTimer, which trigger a sequence of operations: first it asks the DBProxy for the tags to be read from the database, then it asks the OPCProxy for the tags to be read directly from the Process Equipments, then bot tags from database and from OPC servers are fed into JESS. After that, the system starts an n-level loop, where for each loop, the system asks the JESS engine to perform one operational cycle, which first execute JESS rules, then Fuzzy JESS rules and then the neural network. The whole process is shown in a collaboration diagram in Fig. 6.

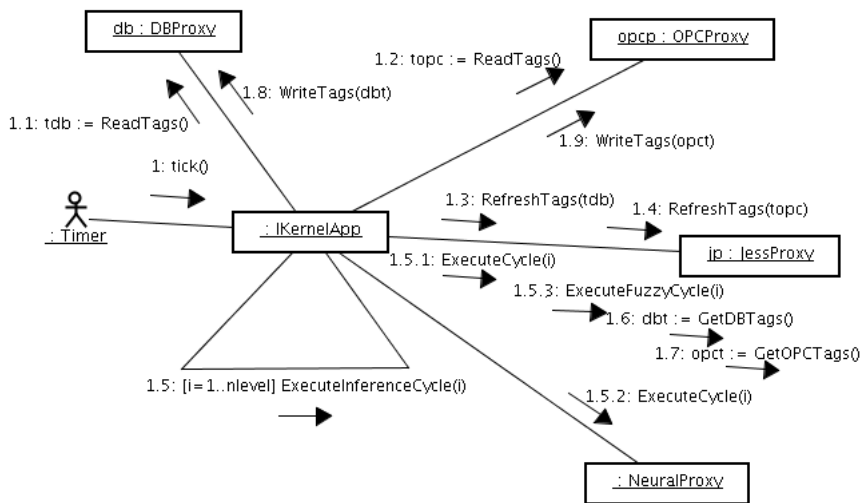


Figure 6. Intelligent Processing Collaboration Diagram

4.3 Use-Case 3: Alarms and Alerts Verification

The alarms and alerts sub-system works in parallell to the intelligent processing, and the mechanism is somewhat the same. The AlarmTimer perform periodic "ticks" to the IKernelApp, which in response perform a verification of alarms and alerts. The system verify a certain number of monitored tags, and for each tag it checks if the current value is out of bounds. If the tag value is out of normal, the system checks if the monitored tag should send an e-mail alert or a popup to the operator, as defined in configuration. After that, it logs the fact with the Logger object and performs either the sending of an e-mail message or a popup alert message. This behavior is described in Fig. 7.

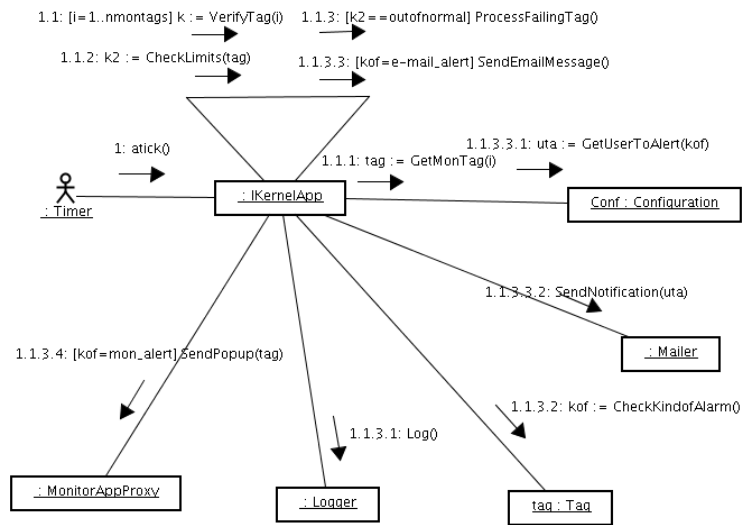


Figure 7. Alarms and Alerts Verification Collaboration Diagram

5. Configuration and Monitoring Application

The Configuration and Monitoring application is a web application, destined to promote the configuration of parameters of the I-Kernel, as well as the Monitoring of the variables under control of the system and the request of shutdown of the I-Kernel. This monitoring can be a direct supervision of the state of some variable (by the selection of the variable among all those which are available), or it can be a monitoring by synoptic. In the direct variables monitoring, the user chooses the variable to be monitored, and the system exhibits its state directly.

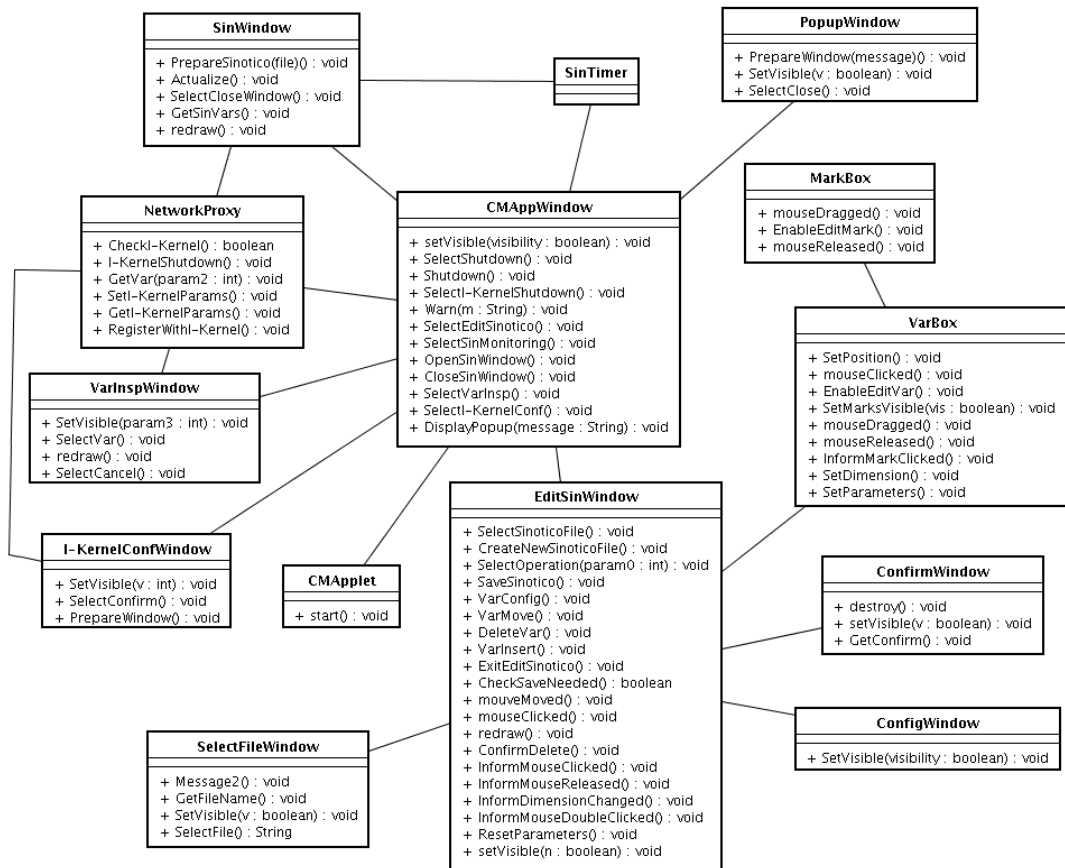


Figure 8. Configuration and Monitoring Application Collaboration Diagram

This monitoring type is targeted to an isolated inspection in the state of some variable of the system. In the synoptic monitoring, it's first required the definition of a synoptic, through the configuration of a screen with a group of variables that one want to monitor, and once a synoptic has been created, the user can choose it for synoptic monitoring. Besides these use cases, the system must treat request from the network, usually originated by the I-Kernel application, so that an alarm and alert pop-up is exhibited for the user. That pop-up request is usually generated from the verification of alarms and alerts use case at the I-Kernel. All the parameters and classes are shown in Fig. 8.

5.1 Use-Case 4: I-Kernel Shutdown

For this use case to start, the C & M tool should be active. To shutdown the I-Kernel application, the user first asks the I-kernel to shutdown. The system then verifies the I-kernel status. If it's off, then it sends an error message, but if it's on, the system starts the shutdown of the active I-Kernel.

5.2 Use-Case 5: C & M Tool Initialization

The user loads an URL from the Web browser, which corresponds to the address of the Configuration and Monitoring Tool. The web page loads the CMApplet which should initialize the C & M tool and open its main user interaction window.

5.3 Use-Case 6: C & M Tool Shutdown

Once the C & M Tool is open and operational, the user requests the system to close this tool.

5.4 Use-Case 7: Synoptic Edition

This is a complex use-case, which is extended by 4 other use-cases. Through this use-case, the user edits a synoptic to be used in another use case: Synoptic Monitoring. Basically, the user informs the synoptic name to be edited (it can be a new synoptic or a synoptic previously edited), and the system opens that synoptic for edition. With the synoptic opened, the user may have a series of options that can be chosen in any order: Variable Insertion, Variable Deletion, Variable Movement, Variable Configuration, Save synoptic, and conclude synoptic edition. If the user made all of the necessary editions, he can decide to save or close the synoptic. The details of this use case are in Fig. 9.

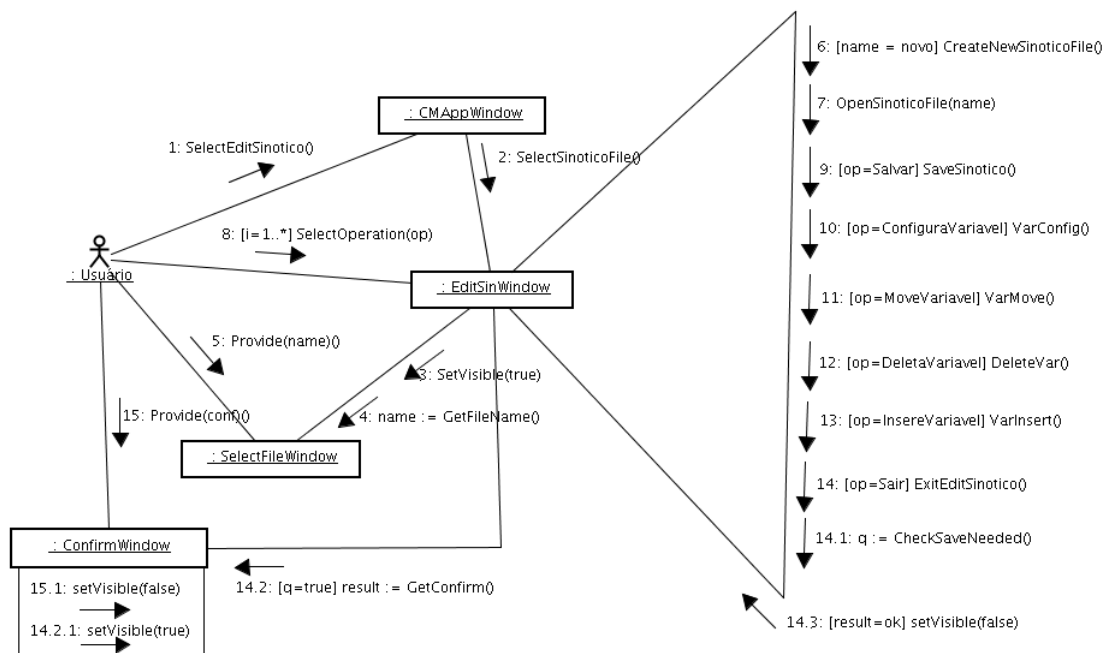


Figure 9. Synoptic Edition Collaboration Diagram

5.5 Use-Case 8: Variable Insertion

This use-case is an extension of the "Synoptic Edition" use-case, allowing the user to insert a new variable to be monitored in the synoptic being edited. Basically the user selects the new variable to insert, positioning the mouse where he wants it to be inserted. The system inserts the new variable being monitored and upgrades the screen.

5.6 Use-Case 9: Variable Deletion

This use-case is an extension of the "Synoptic Edition" use-case, allowing the user delete a variable being monitored in the synoptic. Basically, the user selects the variable to be deleted and asks the system to delete it.

5.7 Use-Case 10: Variable Movement

This use-case is an extension of the "Synoptic Edition" use-case, allowing the user to move around a variable being monitored in the synoptic. Basically, the user clicks on the variable to mark it. After that he clicks in it again, without releasing the mouse button and then moves it for the final position. When releasing the mouse button, the system moves the variable to the wanted position.

5.8 Use-Case 11: Variable Configuration

Permit the user to configure the variable monitored in the synoptic edition. Basically the user has two options: he can either resize the variable, or modify the variable's configuration parameters. If he wants to resize the variable, he clicks in one of the variable marks and moves it to the new wanted point. The system will resize the variable considering the new point. Alternatively, the user can give a double click in the variable, and a new window with the configuration parameters will appear. In it, the user can edit all the variable parameters which he wants, and after finishing the configuration, he can confirm or cancel the alterations.

5.9 Use-Case 12: Synoptic Monitoring

After a synoptic is edited in use case 7: Synoptic Edition, it may be used by the user to obtain the monitoring of a group of variables in a window, defined in the synoptic template. The user basically chooses the file with the synoptic template, and the system opens the monitoring window and it begins the timer that will make the updating of the screen. After that, the system will be waiting to the user request for closing the window, and when this happens, it deletes the timer and the monitoring window.

5.10 Use-Case 13: Synoptic Actualization

This use-case complements the use-case 12: Synoptic Monitoring. Basically, when a synoptic window is being monitored, it is associated with a synoptic timer that sends periodic "ticks" of clock. In each "tick", the system captures the synoptic variables through OPC server and database (it has been configured through the use-case Variable Configuration), updates its values and update the screen to see the new values.

5.11 Use-Case 14: Variables Inspection

The user has the option of inspecting one or more variables. The user selects the variables to inspection, and the system opens a variables inspection window. The user can select which variable he wants to inspect, and the system show that variable in the appropriate place and exhibits the variable value. The user then can decide to inspect another variable or to conclude the operation.

5.12 Use-Case 15: I-Kernel Configuration

The user can update the parameters of the I-Kernel configuration. The user requests to the system the I-Kernel configuration, and the system opens a configuration window, containing all of the parameters configured. The user then can make the alterations that he intends, and confirm the changes or to leave canceling the operation.

5.13 Use-Case 16: Pop-up Alert Solicitation

This use-case is used together with use-case 3: Alarms and Alerts Verification. The I-Kernel subsystem sends a request through the network asking for the alert pop-up to appear. In the present use-case, the request arrives through the network, and the C & M subsystem creates the pop-up window, showing the message wanted to the user. After reading the message the user can request the closing of the pop-up window.

6. CONCLUSIONS

We briefly presented in this paper the requirements and design of an intelligent kernel (I-Kernel) to be used in the construction of a Predictive Maintenance System of a Power Plant. The modelling complies with RCM (Reliability centered maintenance) concepts. The system basically consist of the ConfMonitToolApp module and the I-kernelApp module. This application is actually going to a coding phase, and the next steps are to fully test the application within the power plant. Despite its use in this particular application, the concept of an intelligent kernel goes beyond simply this project, being useful for the construction of similar applications in the future.

7. ACKNOWLEDGEMENTS

We acknowledge the support of the Eletronorte and Manaus Energia provided by the Reserch and Development Program under contract number 4500052325, project number 128 "Modernization of Processes Automation Area of the Hydroelectric power stations of Balbina and Samuel", that has as a technical responsible Prof. Alberto José Álvares of the UnB, and we acknowledge too to the engineer Antonio Araujo from Eletronorte, who played a significant role in this work for the information provided that was import to develop this project.

8. REFERENCES

- Alvares Alberto, 2006, "SIMPREGAL: Metodologia do Sistema de Manutenção Preditiva da Usina de Balbina Baseado dos dados Monitorados do Sistema De Supervisão e Controle Smar e Rockwell", Relatório Técnico de Pesquisa UnB.
- Gudwin Ricardo, 2006, "Especificação do Sistema - Sistema I-Kernel: Um Kernel Inteligente para o SIMPREBAL - Sistema de Manutenção Preditiva de Balbina", Relatório Técnico.
- Jacobson Ivar, Grady Booch, James Rumbaugh, 1999, "The Unified Software Development Process", Addison Wesley.
- Moubray John, 1997, "RCM II - Reliability-Centered Maintenance" 2nd edition, New York: Industrial Press Inc.
- Smith Anthony M., 1993, "Reliability-Centered Maintenance" - McGraw-Hill, USA.
- Sommerville Ian, "Engenharia de Software", 6a. edição, Addison-Wesley/Pearson

Anexo 4
Paper Submetido e ACEITADO no CIBIM2007



8º CONGRESSO IBEROAMERICANO DE ENGENHARIA MECANICA

Cusco, 23 a 25 de Outubro de 2007

SISTEMA INTELIGENTE DE MANUTENÇÃO BASEADA EM CONDIÇÃO PARA USINA HIDRELÉTRICA DE BALBINA

Amaya E. J.*, Tonaco R.*, Souza R. Q.* Álvares A. J.*

*Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Inovação em Automação Industrial (GIAI), CEP 70910-900, Brasília, DF, Brasil.

*edgar.amaya@gmail.com, *rosimarci@gmail.com, *rodrigoqsouza@gmail.com, *alvares@AlvaresTech.com

RESUMO

Neste artigo é apresentada a metodologia que foi desenvolvida no contexto do projeto de Modernização da Área de Automação de Processos das Usinas Hidrelétricas de Balbina e Samuel, onde o objetivo é o desenvolvimento de um sistema inteligente de manutenção preditiva da usina de Balbina. A metodologia SIMPREBAL é baseada em conceitos de manutenção centrada em confiabilidade (MCC), desenvolvendo um sistema chamado de I-Kernel, um kernel inteligente, que vai obter dados dos bancos de dados do sistema supervisorio da Rockwell, do sistema de gestão da manutenção e operação Maximo, do sistema Assetview e do proprio Simprebal, usando JDBC (Java Database Connectivity) e dados do Servidor OPC (OLE – Object Linking and Embedding – for Process Control) da instrumentação Fieldbus Foundation da Smar usando a biblioteca openscada do projeto Utgard, além de obter dados o I-Kernel processará informações usando sistemas especialistas baseados em regras usando JESS (Java Expert System Shell) e técnicas de Inteligência Artificial tais como, redes neurais e lógica fuzzy usando Fuzzy JESS. Os dados dos defeitos e falhas obtidos vão ser armazenados num banco de dados do Simprebal para serem usados por um sistema de aprendizagem baseado em redes neurais e predizer falhas usando essas informações. O sistema é concebido para coletar e analisar os dados monitorados das unidades geradoras da usina de Balbina, e a implementação de um sistema computacional cliente-servidor com vistas à produção de diagnósticos de estados de funcionamento e de dados que auxiliem a tomada de decisão quanto às ações operacionais e de manutenção das máquinas gerando OS (ordens de serviço) com sugestões das ações de manutenção, visando o aumento da disponibilidade dos equipamentos.

PALAVRAS CHAVE: Manutenção Preditiva, Sistemas Especialistas, Redes Neurais, Lógica Fuzzy, Fieldbus Foundation.

INTRODUÇÃO

O modelo OSA-CBM (Open System Architecture for Condition Based Maintenance) que consiste em sete camadas mostrado na Fig. 1, será utilizado como referência para o desenvolvimento do sistema de manutenção inteligente baseado em condição, maiores detalhes sobre o modelo, podem ser encontrados no site <http://www.mimosa.org/downloads/39/specifications/index.aspx>. A noção de uma arquitetura estendida em camadas usada aqui é consistente com o conceito usado em [Buschman, 1996]. As camadas hierárquicas representam uma transição lógica ou um fluxo da saída dos sensores para a camada de tomada de decisão, através das camadas intermediárias até chegar à camada de apresentação.



Fig. 1: Arquitetura OSA-CBM e suas 7 camadas.

A seguir será apresentado um exemplo das sete camadas estudando o caso do mancal guia da turbina e como estas se ajustam ao sistema SIMPREBAL que é baseado em MCC (Manutenção Centrada em Confiabilidade). Segundo [Fleming et al., 1997], a MCC envolve uma consideração sistemática das funções do sistema, a maneira como essas funções falham é um critério de priorização explícito baseado em fatores econômicos, operacionais e de segurança para a identificação das tarefas de manutenção aplicáveis tecnicamente e custos eficientes no combate a essas falhas.

ESTUDO DE CASO: MANCAL GUIA DA TURBINA

Esta seção apresenta uma breve explicação sobre os mancais do grupo turbina-gerador, bem como, um estudo de caso utilizando como exemplo o mancal guia da turbina. O mancal guia da turbina é responsável por suportar o empuxo radial do eixo da turbina. O arranjo dos mancais do grupo turbina-gerador mostrado na figura 2 é o seguinte: o mancal guia inferior é colocado próximo à extremidade inferior do eixo, na tampa superior e compatível com o arranjo e o acesso à vedação do eixo; o mancal combinado ou mancal guia-escora é suportado por uma estrutura cônica apoiada na tampa superior; e o mancal de guia superior ou mancal do gerador é locado imediatamente acima do rotor do gerador, no centro da aranha superior do gerador, cujo peso deve ser suportado pela carcaça do estator, mas as forças radiais do mancal devem ser suportadas pela estrutura de concreto.

Uma vez definida as funções do sistema utilizando as sete camadas do modelo OSA-CBM, pode-se definir os módulos das camadas para o sistema em estudo do mancal guia da turbina.

Módulo de sensor: O módulo do sensor deve ser desenvolvido de acordo com o padrão IEEE Std 1451 segundo [Bengtsson, 2004]. A aquisição de dados necessários para o processamento da informação relativa ao mancal guia da turbina, a obtenção das Tags referentes a cada equipamento é feita via OPC para Tags da instrumentação Fieldbus Foundation da Smar da serie 302 é feita segundo [Smar, 2001], e as Tags do banco de dados da Rockwell via JDBC mostrado na Fig. 3.

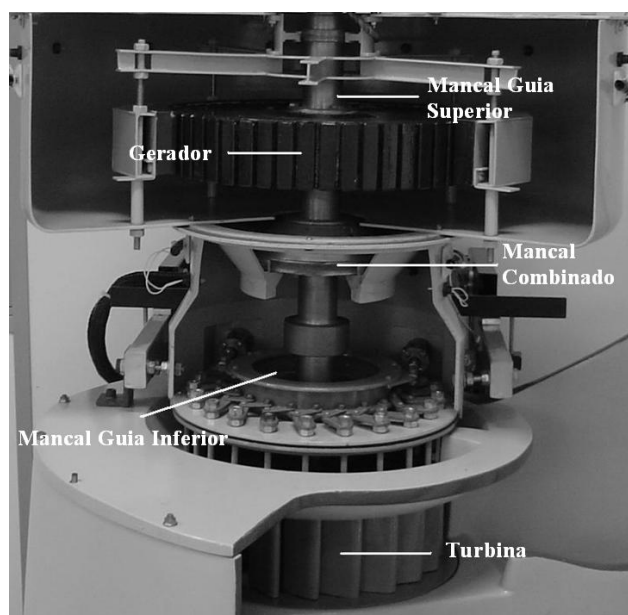


Fig. 2: Maquete de uma Unidade Geradora Hidraulica.

Processamento de sinal: O sistema deve acessar os dados da Usina de Balbina a partir dos Bancos de Dados SQL utilizados pelo Sistema de Monitoramento de Balbina, Sistema de Gestão de Operação e Manutenção Máximo ou diretamente da instrumentação Fieldbus Foundation da Smar através do servidor OPC mostrado na Fig. 3, sendo que estes disponibilizam as informações on-line dos equipamentos. Para obtenção das variáveis on-line via OPC de acordo com as configurações dos blocos funcionais segundo [Smar, 2005], está em desenvolvendo um programa para aquisição das Tags necessárias pra avaliação dos equipamentos usando a biblioteca openscada do projeto Utgard [Utgard, 2007]. Cada Tag tem um valor de Status que é um número de 8 bits, onde os 2 bits mais significativos representam a qualidade do signal do dispositivo (0=Bad, 1=Uncertain, 2=Good Non Cascade, 3=Good Cascade), os 2 bits menos significativos são os limites (0=Not Limited, 1=Low Limited, 2=High Limited, 3=Constant), os quatro bits intermediário representam o substatus, onde para cada valor da qualidade existem valores de substatus e se ocorrer um defeito ou falha é indicado com números diferentes, assim pode-se saber se foi no sensor, equipamento ou outros, segundo a tabela 1. As regras para esta camada mostra-se na tabela 2 onde na primeira coluna compara-se o status e o subStatus com número e na segunda coluna declara-se um código para essa condição da Tag.

Tabela 1: Valores de Sub-status e qualidade no atributo de Status.

		Qualidade			
		0 = Bad	1 = Uncertain	2 = Good (Non Cascade)	3 = Good (Cascade)
Sub-status	0	Não específico	Não específico	Não específica (baixa prioridade)	Não específica
	1	Erro na configuração	Último valor usável	Alarme ativo de bloco	Inicialização aprovada (IA)
	2	Não conectado	Substituto	Alarme ativo de consulta	Requisição de inicialização (IR)
	3	Falha no Equipamento	Valor inicial	Alarme ativo critico	Não solicitado (NI)
	4	Falha no Sensor	Conversão do sensor não exato	Alarme de bloco desconhecido	Não selecionado (NS)
	5	Sem Comunicação, com último valor usavel	Violação de limite de unidade técnica	Alarme de consulta desconhecido	Cancelamento local (LO)
	6	Sem Comunicação, com valor não usável	Sub-normal	Alarme crítico desconhecido	Estado de falha ativo (FSA)
	7	Fora de serviço (alta prioridade)			Estado de falha iniciado (IFS)

Tabela 2: Regras para a camada Processamento de sinal.

Status e subStatus das Tags	Código
status = "2" ou status = "3"	(assert (signal-GOOD ?label))
status = "0" e subStatus = "0"	(assert (signal-BAD-0 ?label))
status == "0" e subStatus = "1"	(assert (signal-BAD-1 ?label))
status == "0" e subStatus = "2"	(assert (signal-BAD-2 ?label))
status == "0" e subStatus = "3"	(assert (signal-BAD-3 ?label))
status == "0" e subStatus = "4"	(assert (signal-BAD-4 ?label))
status = "1" ou (status = "0" e (subStatus = "5" ou subStatus = "6" ou subStatus = "7"))	(assert (signal-UNCERTAIN ?label))

Monitoração de Condição: Esta camada determina a condição do sistema atual, subsistema, ou indicadores de condição de componente baseando-se em algoritmos, sistemas inteligentes e na saída dos módulos de sensores e processamento de sinal. Este módulo de monitoração deve ser desenvolvido usando o padrão ISO 13373-1 segundo [Bengtsson, 2004]. É possível fazer o uso de histórico de condição local e fornecer parâmetros para o modelo. Duas dimensões devem ser consideradas: primeira, o sistema Assetview da Smar e o sistema MES Máximo fornecem análises relativa à condição da instrumentação FieldBus; segunda dimensão, o sistema SIMPREBAL fornece a condição do sistema monitorado. Um exemplo de regras de produção nesta camada para a Tag da pressão de óleo na cuba pode ser visto na tabela 3, onde os valores da tag são comparados com valores da faixa de operação do dispositivo e gera-se um código para a condição do dispositivo.

Tabela 3: Regras para a camada Monitoração de Condição.

Faixa de Valores (mm bar)	Codigo	Condição
(400 < G163MS_AI.PV.VALUE) e signal-GOOD	G163MS-normal	Nível normal de óleo na cuba do mancal guia inferior
(300 < G163MS_AI.PV.VALUE < 400) e signal-GOOD	G163MS-baixo	Nível baixo de óleo na cuba do mancal guia inferior
(200 < G163MS_AI.PV.VALUE < 300) e signal-GOOD	G163MS-alarme	Nível muito baixo de óleo na cuba do mancal guia inferior - 1º estágio
(G163MS_AI.PV.VALUE < 200) e signal-GOOD	G163MS-trip	Nível muito baixo de óleo na cuba do mancal guia inferior - 2º estágio

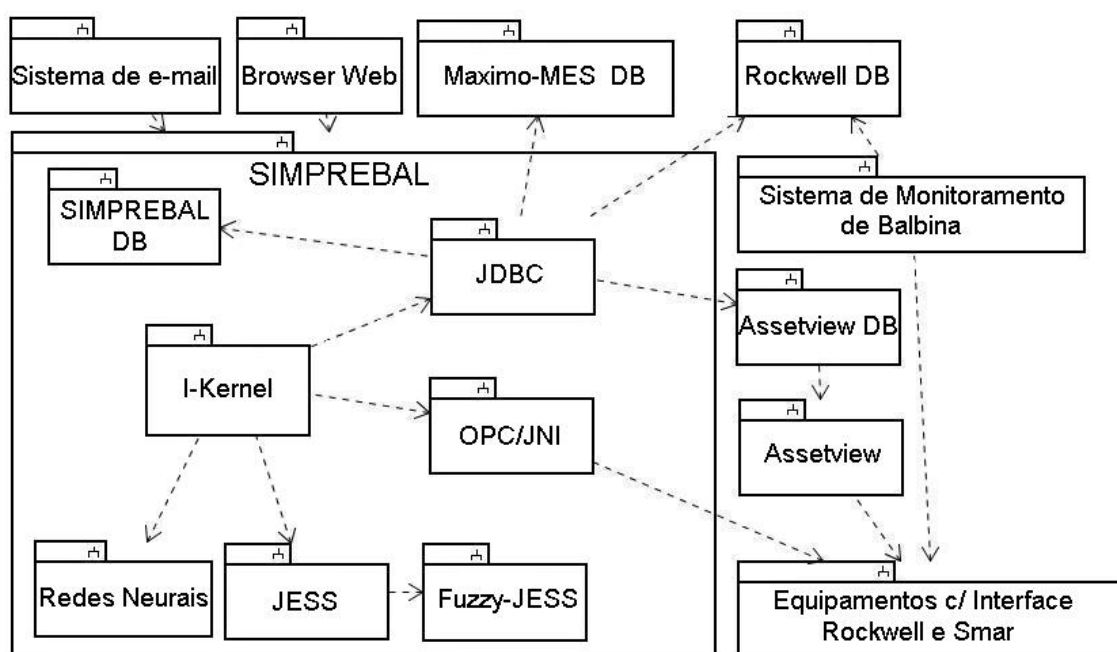


Fig. 3: Visão Geral do Projeto SIMPREBAL, demonstrando o I-Kernel e sua interação.

Avaliação de saúde: Esta camada deve ser desenvolvida a partir das normas IEEE 1232 e a ISO 13373-1 segundo [Bengtsson, 2004]. Na primeira dimensão o conversor FieldBus e o sistema Assetview da Smar fornecem a avaliação da saúde da instrumentação FieldBus. Na segunda dimensão o sistema SIMPREBAL fornece a avaliação da saúde do sistema monitorado. A avaliação da saúde do equipamento ocorre através da extração das características de cada equipamento e posterior detecção de anomalias e diagnósticos dos mesmos. A análise será feita usando os valores históricos das Tags, cálculo do KPI (Key Performance Indicators), os valores das tags serão obtidos do banco de dados do supervisor da Rockwell. No encerramento das OS (Ordens de Serviço), o usuário deverá encerrar as OS fornecendo todos os dados para registro histórico e cálculo de KPI.

Prognósticos: Na tarefa de prognóstico, considera-se a avaliação da saúde do sistema, subsistemas, componentes, o escalonamento empregado (predição de uso) e a capacidade de raciocínio do modelo para prever o estado de saúde do equipamento. Um prognóstico é sugerido a partir das informações dos módulos anteriores. Este pode ser baseado em modelos pré-estabelecidos ou feito considerando as características encontradas na avaliação de saúde. O prognóstico vai consistir de dois passos, o diagnóstico que será feita com os valores atuais e o prognóstico que vai ser feito, usando tendência de tags armazenadas no banco de dados ou através de redes neurais que vão prever falhas ou defeitos futuros baseadas em dados históricos ou acontecimentos passados. As regras para diagnóstico são baseadas em valores reais e são mostradas na tabela 4.

Tabela 4: Regras de produção para Diagnóstico.

Sinal de "g1.st.p.oleo.cuba"	Codigo de Diagnostico
signal-BAD-0	assert(diagnostic G163MS-signal0)
signal-BAD-1	assert(diagnostic G163MS-signal1)
signal-BAD-2	assert(diagnostic G163MS-signal2)
signal-BAD-3	assert(diagnostic G163MS-signal3)
signal-BAD-4	assert(diagnostic G163MS-signal4)
signal-UNCERTAIN	assert(diagnostic G163MS-signal5)
not (condition G163MS-normal)	assert(diagnostic G163MSF1)

Tomada de decisão: Tem o objetivo de integrar informações necessárias para a tomada de decisão, considerando informações sobre a saúde dos equipamentos e para prever a saúde de um sistema e/ou subsistema. O sistema SIMPREBAL irá realizar a tomada de decisão baseando-se na sua base de conhecimento gerada a partir da árvore de faltas/falhas, da árvore de sintomas, e pelas informações de inspeções realizadas pelo sistema TPM (Manutenção Produtiva Total), segundo [Wyrebski, 1998] hoje é considerada uma metodologia que auxilia na redução de ocorrência de falhas em instalações de máquinas e equipamentos nas empresas. Nesse módulo o sistema inteligente apresenta algumas sugestões para a solução do problema, cabe ao usuário tomar a decisão mais aplicada. As regras para esta camada com as sugestões de manutenção são mostradas na tabela 5, a decisão é enviada do servidor i-kernel para o cliente onde é mostrada na tela que interage com o pessoal da operação, gerando OS para manutenção.

Tabela 5: Regras de produção para Tomada de Decisão.

Codigo	Decisão
G163MS-signalD0	Falha no processamento de sinal do tag G163MS: Não específica
G163MS-signalD1	Falha no processamento de sinal do tag G163MS: Erro na configuração
G163MS-signalD2	Falha no processamento de sinal: verificar conexão do sensor de pressão à cuba e ao transmissor G163MS
G163MS-signalD3	Verificar falha no transmissor de pressão G163MS
G163MS-signalD4	Verificar falha no sensor de pressão diferencial da cuba do mancal guia inferior
G163MS-signalD5	Verificar falha de comunicação na aquisição do tag G163MS
G163MSD1	Verificar estanqueidade das tubulações, válvulas e tampas de acesso
	Verificar aperto dos parafusos de fixação da tampa e suporte da cuba
	Realizar alinhamento e ajuste de folga das sapatas
	Restituir a carga de óleo

Apresentação: Suporta a apresentação das informações para o controle das entradas dos usuários do sistema (por exemplo, manutenção e operação). As saídas incluirão qualquer informação produzida pelas camadas mais baixas e as entradas incluirão qualquer informação requerida pelas camadas mais baixas. A figura 4 apresenta a interface homem-máquina associada ao lado do Cliente, sendo desenvolvida em Java para ser usada como uma aplicação Web ou um Applet chamado via browser, independentemente da plataforma computacional usada pelo cliente. A GUI (Interface Gráfica com o Usuário) mostrada na Fig. 4 apresenta a camada de Apresentação (sétima camada) do sistema Simprebal mostrando a Tomada de Decisão (sexta camada) sugerida pelo sistema SIMPREBAL. A GUI apresenta também os valores das Tags em inspeção de variáveis e a estrutura hierárquica dos ativos.

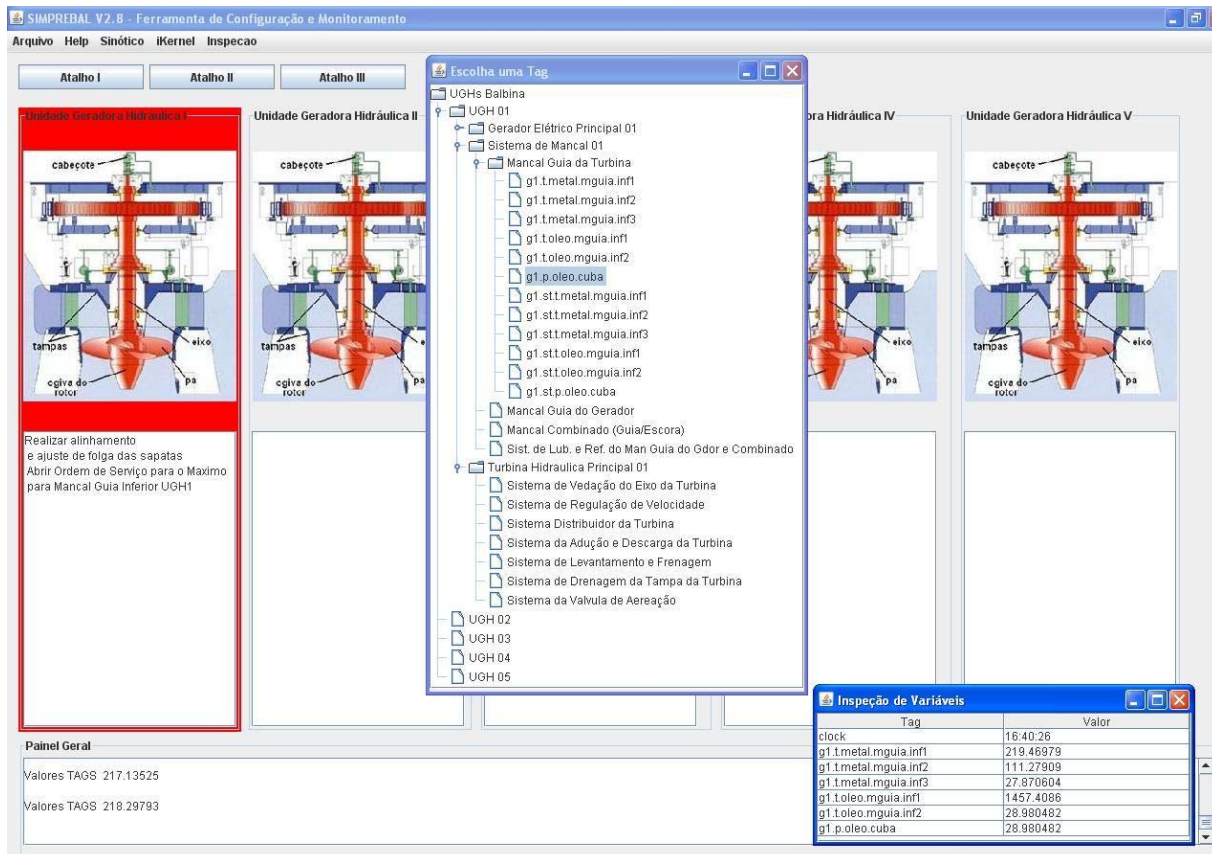


Fig. 4: GUI SIMPREBAL Configuração e Monitoração: Cliente Web referente à camada de Apresentação da metodologia SIMPREBAL.

CONCLUSÃO

Com a proposta apresentada no presente artigo, espera-se mostrar sugestões de manutenção para os operadores e o pessoal de manutenção antes que o equipamento apresente problema. Esse processo de manutenção é conhecido como manutenção preditiva e é baseada em confiabilidade. O sistema SIMPREBAL será desenvolvido usando técnicas de IA (Inteligência Artificial) como regras de produção, Lógica Fuzzy e redes neurais, o que facilitará o processo de tomada de decisão, visto que um dos objetivos do sistema é avaliar e monitorar o equipamento para oferecer prognósticos confiáveis. Permitindo dessa maneira que o operador possa tomar decisões de manutenção com base em informações geradas pelo sistema. O Sistema Inteligente SIMPREBAL auxiliará na tomada de decisão sugerindo prognósticos, para que a manutenção do equipamento possa ocorrer antes que a falha ocorra. Isto implica em ganho de tempo e maior confiabilidade do sistema da usina. Vale ressaltar que a metodologia aqui proposta ainda está em fase de elaboração, resultados concretos serão obtidos posteriormente.

AGRADECIMENTOS

Agradecemos à Eletronorte, Manaus Energia e ao Eng. Antonio Araújo da Eletronorte pelas informações cedidas que foram de extrema importância para o desenvolvimento do presente trabalho.

REFERÊNCIAS

1. F. Buschman, *Oriented Software Architecture A System of Patterns*, England, 1996.
2. OSA-CBM, *Open Systems Architecture for Condition-Based Maintenance*.
<<http://www.mimoso.org/downloads/39/specifications/index.aspx>>, acessado em agosto de 2007.
3. P. V. Fleming e S.R.R.O. França, *Considerações Sobre a Implementação Conjunta de TPM e MCC na Indústria de Processos*, Abraman 12º Congresso Brasileiro de Manutenção: TT044, São Paulo, pp.53, 1997.
4. M. Bengtsson, *Condition Based Maintenance System Technology –Where is Development Heading?*, *Proceedings of the 17th European Maintenance Congress*, Spain, 2004.
5. Smar, *Equipamentos de Campo série 302 Foundation. Manual de instalação, operação e manutenção*, 2001.
6. Smar, *Manual de instruções dos blocos funcionais Fieldbus Foundation*”, 2005.
7. Utgard, *Projeto Utgard*, <<http://openscada.org/UtgardProject>>, acessado em agosto de 2007.
8. J. Wyrebski, *Manutenção Produtiva Total um Modelo Adaptado*”, *Dissertação (Mestrado em Engenharia de Produção) Programa de Pós-Graduação em Engenharia de Produção, UFSC*, 1998.