BODY REFERENCE : SPEC1_BRAKES_REV F

TITLE :	SPECIFICATION OF IBERDROLA FUNCTIONS: "DETECT PROBABILITY OF FUTURE BRAKING SYSTEM DISTURBANCES (F10)"
SOURCE :	IBERINCO - UITESA
STATUS :	FINAL DOCUMENT
EDITOR :	JOSÉ MANUEL HERNÁNDEZ
VERIFIED BY :	
DISTRIBUTION :	ALL CONSORTIUM PARTICIPANTS
DATE :	5 SEPTEMBER 1997
NUMBER OF PAGES :	23
INTERNALREFERENCE:	J:\DOK\13\REMAFEX\WP4\F10.DOC

CONTENTS

0. OBJECTIVE OF THE FUNCTION

The objective of the function is to detect and report any anomalous situation concerning the braking system that can produce a future malfunction .

This objective is decomposed into another seven functions corresponding to the micro needs associated to the macro need this function describes. The final objective of this function will be to detect any malfunction from the results returned by the micro needs functions. Thus, the output of this function will be the seven micro needs (incidents) ordered by the certainty factors associated to each incident.

The incidents relative to braking system malfunction are:

_ Jacks wedging

_ Air leakage

_ Shoes wearing

_ Deformation / breaking of braking band

_ Braking jacks pistons scratching

_ Sudden braking with machine started

_ The machine does not stop

This information will be enough for the expert to take a decision about the preventive maintenance of the plant in a concrete item (braking system).

Also, the system will be able to make cause determination for each one of the possible incidents and to give a justification of the results.

1. FUNCTION ENVIRONMENT

The function will evaluate for each incident its own certainty factor showing the expert degree of confidence for that event to occur under the conditions given by the measures.

The information presented to the user will contain the list of incidents associated to the braking system, ordered by degree of certainty. Also, the degree of certainty will be showed for the user to evaluate the importance of each possible event.

This evaluation will be developed as a user request or cyclically, and will use data collected by the SCADA instrumentation in real time and data collected in periodical tests if they are available (or included by the user).

The sequence of requests and responses of the function is the following:

Event	Request/Response (RQ/RS)	From	To
Forecasting Request (user request)	RQ	Maintenance Operator (M.O.)	Function
Forecasting Request (cyclic execution)	RQ	Function	Function
Collect Data (continuous extern function with a fixed sample interval)	RQ	Function	Data Acquisition System (D.A.S.)
Data Collected (continuous function with a fixed sample interval)	RS	Data Acquisition System (D.A.S.)	Data Base
Select Data (on forecasting request)	RQ	Function	Function
Data Selected (on forecasting request)	RS	Data Base	Function
Perform Forecasting	RQ	Function	Function
Manual Data Request	RQ	Function	M.O.
Manual Data (Manual Tests)	RS	M.O.	Function/ Data Base
Report Results (ordered list of incidents with their associated certainty degree)	RS	Function	M.O.
Store Results (certainty degree incidents)	RS	Function	Data Base

There are two processes involved in the data handling. There is a process that collects data from the D.A.S. and includes it into the real time data base (D.B). The other process selects the data needed by the function in a range of time. The first function will operate independently from the second and in a continuous way. The second will respond to the maintenance operator requests and when expiring the hibernating period .

2. INPUT DATA DEFINITION

Input data to the function are divided into three types:

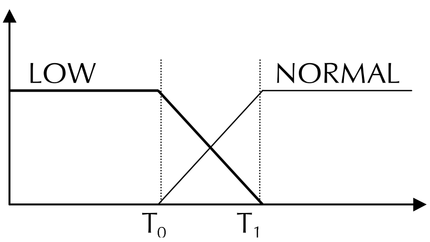
1. Digital signals from SCADA system: We will need the last updated value from the SCADA system contained in the real time DB.
2. Analogic signal from SCADA system: We will also need the last updated value from the SCADA system contained in the real time DB.
3. Digital and analogic data, inserted by the user into the real time DB: We will need the last value updated by the maintenance operator as well as the date when it was updated.

All these data must be presented to the user before executing the function and the user must be able to modify any value to adjust the results obtained in the execution. These modifications must not be updated in the DB.

The input data will be the following SCADA signals and maintenance actions that could indicate possible malfunction incidents:

The following data are used in the algorithms for the forecasting function:

CODE:
" BRAKING_TIME_TOO_LONG ":
NAME:
Generator braking time too long
TYPE:
boolean, digital
RANGE:(limits)
on-off
ACQUISITION, SOURCE:
SCADA
XD-V0427 BRAKING_TIME_TOO_LONG (Pelton)
XD-V0805 BRAKING_TIME_TOO_LONG (Francis)
LABELS:
on (alarm)
off
Default value: off

CODE:
" TIME_BETWEEN_BR_COMPRESSOR_START "
NAME:
Lapsed time between compressor starting (without braking operations)
TYPE:
integer
RANGE:(limits)
0-1000 (hours)
ACQUISITION, SOURCE:
TIME_BETWEEN_BR_COMPRESSOR_START
The signal is acquired from the following signals:
SCADA, missing by SCADA:
AIR_BRAKING_COMPRESS_START
AIR_BRAKING_COMPRESS_STOP
This signals indicate when compressor starts and stop. The system has to calculate lapsed time since compressor is stopped until it starts again (time without working).
LABELS:
NORMAL
LOW
T_0 = **pending**
T_1 = **pending**

Default value: NORMAL

CODE:

" TIME_BR_COMPRESSOR_WORKING "

NAME:

compressor working time (without braking operations)

TYPE:

integer

RANGE:(limits)

0-10000 (min)

ACQUISITION, SOURCE:

TIME_BR_COMPRESSOR_WORKING

The signal is acquired from the following signals:

SCADA, missing by SCADA:

AIR_BRAKING_COMPRESS_START

AIR_BRAKING_COMPRESS_STOP

This signals indicate when compressor starts and stop. The system has to calculate lapsed time since compressor starts until it is stopped again (working time).

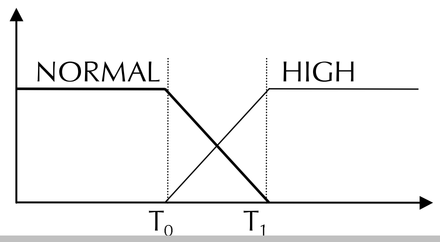
LABELS:

NORMAL

HIGH

T_0 = **pending**

T_1 = **pending**



Default value: NORMAL

CODE:

" CORRECT_AIR_BRAKE_PRESS "

NAME:

normal air braking pressure

TYPE:

boolean, digital

RANGE:(limits)

on-off

ACQUISITION, SOURCE:

SCADA

XD-S0211 CORRECT_AIR_BRAKE_PRESSURE (Pelton)

XD-S0510 CORRECT_AIR_BRAKE_PRESSURE (Francis)

LABELS:

on

off (alarm)

Default value: on

CODE:
" BRAKES_RELEASED "
NAME:
brakes released indicator
TYPE:
boolean, digital
RANGE:(limits)
on-off
ACQUISITION, SOURCE:
SCADA
XD-S0210 BRAKES_RELEASED (Pelton)
XD-S0509 BRAKES_RELEASED (Francis)
LABELS:
on
off
Default value: on

CODE:
" BRAKES_APPLIED "
NAME:
brakes applied indicator
TYPE:
boolean, digital
RANGE:(limits)
on-off
ACQUISITION, SOURCE:
SCADA
XD-S0209 BRAKES_APPLIED (Pelton)
XD-S0508 BRAKES_APPLIED (Francis)
LABELS:
on
off
Default value: off

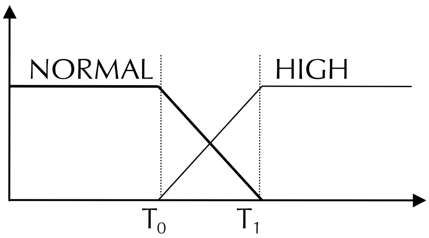
CODE:
" BRAKING_DEFECT "
NAME:
braking defect alarm
TYPE:
boolean, digital
RANGE:(limits)
on-off
ACQUISITION, SOURCE:
SCADA
XD-V0429 BRAKING_DEFECT (Pelton)
XD-V0807 BRAKING_DEFECT (Francis)
LABELS:
on (alarm)
off   Default value: off

CODE:
" BRAKES_APPLIED_INCORRECT "
NAME:
alarm of incorrect application of brakes
TYPE:
boolean, digital
RANGE:(limits)
on-off
ACQUISITION, SOURCE:
SCADA XD-V0428 BRAKES_APPLIED_INCORRECT (Pelton) XD-V0806 BRAKES_APPLIED_INCORRECT(Francis)
LABELS:
on (alarm) off
Default value: off

CODE:
" BRAKING_SYST_AIR_NOISE "
NAME:
air noises in lines
TYPE:
boolean, digital
RANGE:(limits)
YES-NOT
ACQUISITION, SOURCE:
MANUAL INPUT, periodical routes reports BRAKING_SYSTEM_AIR_NOISES (Pelton) BRAKING_SYSTEM_AIR_NOISES (Francis)
LABELS:
YES NOT
Default value: NOT

CODE:
" BRAKE_APPLIC_NUMBER "
NAME:
brake applications number
(If brake applications number > defined value, it implies shoes wearing)
TYPE:
integer
RANGE:(limits)
0-pending (times)
ACQUISITION, SOURCE:
BRAKE_APPLIC_NUMBER (Pelton)
BRAKE_APPLIC_NUMBER (Francis)
The signal is acquired from the following signals:
SCADA:
YD-S0119 BRAKES_APPLY_ORDER (Pelton)
YD-S0302 BRAKES_APPLY_ORDER (Francis)
This signals indicate when brakes have to be applied. The system has to amount the number of brake applications in order to detect brake shoes wearing. This signal will be reset when brake shoes be replaced by maintenance.
LABELS:
NORMAL
HIGH

T_0 = **pending**
T_1 = **pending**



Default value: NORMAL

CODE:
" ABNORMAL_SHOE_NOISE "
NAME:
abnormal noises when shoes are applied
TYPE:
boolean, digital
RANGE:(limits)
YES-NOT
ACQUISITION, SOURCE:
MANUAL INPUT, periodical routes reports
ABNORMAL_SHOE_NOISE (Pelton)
ABNORMAL_SHOE_NOISE (Francis)
LABELS:
YES
NOT
Default value: NOT

CODE:
" ABNORMAL_SHOE_WEAR "
NAME:
abnormal shoes wearing
TYPE:
boolean, digital
RANGE:(limits)
yes-not
ACQUISITION, SOURCE:
MANUAL INPUT, periodical revision reports
ABNORMAL_SHOE_WEARING (Pelton)
ABNORMAL_SHOE_WEARING (Francis)
LABELS:
YES
NOT
Default value: NOT

CODE:
" BAD_BRAKE_BAND_SURFACE "
NAME:
band surface in bad state
TYPE:
boolean, digital
RANGE:(limits)
yes-not
ACQUISITION, SOURCE:
MANUAL INPUT, periodical revision reports
BAD_BRAKE_BAND_SURFACE (Pelton)
BAD_BRAKE_BAND_SURFACE (Francis)
LABELS:
YES
NOT
Default value: NOT

CODE:
" BRAKES_RELEASE_ORDER "
NAME:
order of release generator brakes
TYPE:
boolean, digital
RANGE:(limits)
on-off
ACQUISITION, SOURCE:
SCADA
YD-S0120 BRAKES_RELEASE_ORDER (Pelton)
YD-S0303 BRAKES_RELEASE_ORDER (Francis)
LABELS:
on
off Default value: off

P: Vibrations

The following data are used in the algorithms for the forecasting function:

CODE:

" SET_AXIAL_VIBR ":

NAME:

axial oscillations and/or vibrations

TYPE:

analogic

RANGE:(limits)

0-100 (mm/s)

ACQUISITION, SOURCE:

SCADA, missing by SCADA

AXIAL_VIBRATION (Pelton)

AXIAL_VIBRATION (Francis)

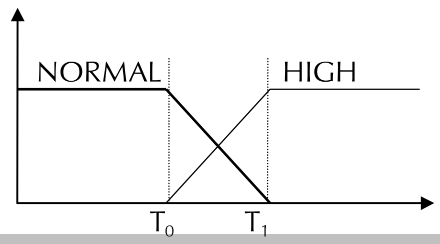
LABELS:

NORMAL

HIGH

T_0 = **pending**

T_1 = **pending**



Default value: NORMAL

The following data are not used in forecasting rules, but it could be used if rules are modified in test period. They will be shown by the function.

" LGB_HIGH_VIBR ":

XD-V0407 lower guide bearing high vibration alarm (Francis)

XD-V----- lower guide bearing high vibration alarm (Pelton)

" UGB_HIGH_VIBR ":

XD-V0408 upper guide bearing high vibration alarm (Francis)

XD-V----- upper guide bearing high vibration alarm (Pelton)

" LGB_VERY_HIGH_VIBR ":

XD-V0410 lower guide bearing very high vibration alarm(Francis)

XD-V--- lower guide bearing very high vibration alarm(Pelton)

" UGB_VERY_HIGH_VIBR ":

XD-V0411 upper guide bearing very high vibration alarm(Francis)

XD-V--- upper guide bearing very high vibration alarm(Pelton)

" UGB_VIBR ":

XD-V--- upper guide bearing vibration indicator (Pelton)

XD-V--- upper guide bearing vibration indicator (Francis)

" LGB_VIBR ":

XD-V--- lower guide bearing vibration indicator (Pelton)

XD-V--- lower guide bearing vibration indicator (Francis)

" SET_VIBR ":

XD-V--- set vibration indicator (Pelton)

XD-V--- set vibration indicator (Francis)

" SET_HIGH_VIBR ":

XD-V0319 set high vibration indicator (Pelton)

XD-V--- set high vibration indicator (Francis)

" SET_VERY_HIGH_VIBR ":

XD-V0322 set very high vibration indicator (Pelton)

XD-V--- set very high vibration indicator (Francis)

V: Group speed

The following data are used in the algorithms for the forecasting function:

CODE:
" SPEED_LESS_20 "
NAME:
speed equal or less than 20% or 25% nominal
TYPE:
boolean, digital
RANGE:(limits)
on-off
ACQUISITION, SOURCE:
SCADA
XD-S0122 SPEED_LESS_20 (Pelton)
XD-S0301 SPEED_LESS_25 (Francis)
LABELS:
on
off
Default value: off

CODE:
" VOID_SPEED_T "
NAME:
void speed
TYPE:
boolean, digital
RANGE:(limits)
on-off
ACQUISITION, SOURCE:
SCADA
XD-S0121 VOID_SPEED_T (Pelton)
XD-S0305 VOID_SPEED_T (Francis)
LABELS:
on
off
Default value: off

The following data are not used in forecasting rules, but it could be used if rules are modified in test period. They will be shown by the function.

" SPEED_LESS_50 ":
XD-S0123 indicator of speed equal or less than 50% nominal (Pelton)
(integer scalar / Boolean)
" SPEED_OVER_85 ":
XD-S0124 indicator of speed equal or more than 85% nominal (Pelton)
XD-S0302 indicator of speed more than 85% nominal (Francis)
(integer scalar / Boolean)

" GROUP_SPEED ":
XA-V0101 group speed indicator (Pelton)
XA-V0101 group speed indicator (Francis)
(analogic, rpm, real scalar)

CODE:
" BRAKES_APPLY_ORDER "
NAME:
order of apply generator brakes
TYPE:
boolean, digital
RANGE:(limits)
on-off
ACQUISITION, SOURCE:
SCADA YD-S0119 BRAKES_APPLY_ORDER (Pelton) YD-S0302 BRAKES_APPLY_ORDER (Francis)
LABELS:
on off
Default value: off

CODE:
" DEFLECTOR_CLOSED "
NAME:
turbine deflector (Pelton) or guide vanes (Francis) in closed position
TYPE:
boolean, digital
RANGE:(limits)
on-off
ACQUISITION, SOURCE:
SCADA XD-S0116 DEFLECTOR_CLOSED (Pelton) XD-S0206 GUIDE_VANES_CLOSED (Francis)
LABELS:
on off
Default value: off

Z: Group breaker position

The following data are used in the algorithms for the forecasting function:

CODE:
" STAT_BREAKER_CLOSED "
NAME:
Group breaker in closed position
TYPE:
boolean, digital
RANGE:(limits)
on-off
ACQUISITION, SOURCE:
SCADA
XD-S0302 STAT_BREAKER_CLOSED (Pelton)
XD-S0705 STAT_BREAKER_CLOSED (Francis)
LABELS:
on
off
Default value:off

The following data are not used in forecasting rules, but it could be used if rules are modified in test period. They will be shown by the function.

" STAT_BREAKER_OPEN ":
XD-S0301 indicator of group breaker in open position (Pelton)
XD-S0704 indicator of group breaker in open position (Francis)

3. OUTPUT DATA DEFINITION

The output of this function will be used to report directly to the user the evaluation results of the different incidents relative to the braking system malfunction. These results are the ordered list of the related incidents with the certainty factor associated to each one. The output will be then for example:

1. *Jacks wedging (0.7) HIGH*
2. *Air leakage (0.7) HIGH*
3. *Shoes wearing (0.2) LOW*
4. *Deformation / breaking of braking band (0.1) VERY LOW*
5. *Braking jacks pistons scratching (0.3) LOW*
6. *Sudden braking with machine started (0.1) VERY LOW*
7. *The machine does not stop (0.4) MEDIUM*

As in the input, the output interface for the function will be a file, that must be created by the function when finishing execution and will contain the name of the incident, the certainty factor and the label associated to the certainty factor.

As mentioned earlier, the list of the possible causes for each incident must be displayed if the user requests it. This information will be also given by the function in the output file.

The list of possible causes is the following:

Jacks wedging:

By jamming cylinder -piston (when brakes are applying)
- foreign particles (piece of seal, particles, dust, water)
- dragging
- scratches
- points of strong friction
- oxidation
- friction between guides and cylinder

By jamming cylinder -piston (when brakes are releasing)
- the same as in the case of applying brakes
- bad working of recuperation spring:
  · lack of strength
  · breaking

Air leakage:

Leakage in the equipment that produces braking air
Leakage in pipes and collector
Leakage in braking jacks:
- coupling input to cylinder
- breaking of torodial seal

Shoes wearing:

Number of manoeuvres during the operative life
Improper material (temperature/resistance/wearing)
Sudden application of brakes
Braking band in bad state
High turbine torque (bad closing of deflector)
Intermittent applications (accelerations between brakes)

Deformation / breaking of braking band:

Improper material for the shoes
Sudden application of brakes
Mechanical stress (abrasion, wearing, scratches, etc.)
Thermal stress (hot points, deformations, etc.)

Braking jacks pistons scratching:

By jamming cylinder -piston (in applying)
- foreign particles (pieces of seal, particles, dust, water)
- dragging
- scratching
- strong friction points
- oxidation
- friction between guides and cylinder

Sudden braking with machine started:

Failures in:
- control position of breaker group (open)
- control position of turbine deflector (closed)
- control of velocity measurement (application)
- air electric valve control (braking)
- time counting control (before braking)

The machine does not stop:

- Jacks wedging (brakes does not apply)
- Air leakage (loss of pressure)
- Shoes wearing (minor friction)
- Deformation of band (minor friction)
- Insufficient time of application (control failure)
- High turbine torque (leakage in deflector closing)
- Machine breaker is closed "dragging"

The output file will also contain a justification text of the deductions made by the fuzzy module to obtain the conclusions.

4. DYNAMIC BEHAVIOUR

As mentioned in section 1. , there are two processes involved in the forecasting functions. The first of them is the responsible of gathering data at sample intervals (given by the availability of SCADA signals) and inserting them into the D.B. in real time. This function allows us to dispose of all the data needed to carry out the forecasting and cause determination needs.

As mentioned above, we will have 2 types of executions:

1. As a user request.
2. Cyclically.

For the configuration of the cyclic execution, the system must provide the way to define:

- The event/s (a group of SCADA signals that satisfy some conditions) that starts the execution of the functions under a certain type of cycle.
- The period of activation for each type of cycle.

Thus, when the event defined by the user for a type of cycle is true, the functions will be executed cyclically within a period of activation. This period could be null, in the case of the group stop event, for example.

5. DATA PROCESSING (ALGORITHMS)

For the data processing, we have to consider the deduction mechanism used to determine the possibility of any braking system malfunction due to a concrete incident of the seven listed previously in section 0.

This mechanism uses the certainty associated to each event of the system to make deductions for each incident. That is, we will have a deduction tree for each incident going from lower to upper nodes, evaluating the rules and spreading the certainty to upper levels. In those cases where we have independent variables for a common conclusion, we will accumulate the certainty for the conclusion. Where we have dependent variables (for example: A defined as A1 AND A2) we must not.

The implementation of the function is based on the deduction rules and their probabilities. The deduction must be made with forward chaining going from the signals to the incidents certainty. All the rules for an incident will be applied accumulating the resulting certainty because they represent independent events.

The rules to apply are listed below grouped by incidents:

Jacks wedging:

IF (“ **BRAKING_TIME_TOO_LONG** “) IS ON THEN
JACK_WEDGING
(CERTAINTY=MEDIUM, RELIABILITY=MEDIUM)

IF (“ **BRAKING_DEFECT** “) IS ON THEN
JACK_WEDGING
(CERTAINTY=HIGH, RELIABILITY=HIGH)

IF (“ **BRAKES_APPLY_ORDER** “) IS ON AND
(“ **BRAKES_APPLIED** “) IS OFF (temporised) THEN
JACK_WEDGING
(CERTAINTY=HIGH, RELIABILITY=HIGH)

IF (“ **BRAKES_RELEASE_ORDER** “) IS ON AND
(“ **BRAKES_RELEASED** “) IS OFF (temporised) THEN
JACK_WEDGING
(CERTAINTY=HIGH, RELIABILITY=HIGH)

Air leakage:

IF (“ **TIME_BETWEEN_BR_COMPRESSOR_START** “) IS LOW THEN
AIR_LEAKAGE
(CERTAINTY=HIGH, RELIABILITY=HIGH)

IF (“ **TIME_BR_COMPRESSOR_WORKING** “) IS HIGH THEN
AIR_LEAKAGE
(CERTAINTY=HIGH, RELIABILITY=HIGH)

IF (" **CORRECT_AIR_BRAKE_PRESS** ") IS OFF THEN
AIR_LEAKAGE
(CERTAINTY=HIGH, RELIABILITY=HIGH)

IF (" **BRAKING_TIME_TOO_LONG** ") IS ON THEN
AIR_LEAKAGE
(CERTAINTY=MEDIUM, RELIABILITY=MEDIUM)

IF (" **BRAKING_DEFECT** ") IS ON THEN
AIR_LEAKAGE
(CERTAINTY=MEDIUM, RELIABILITY=MEDIUM)

IF (" **BRAKING_SYST_AIR_NOISE** ") IS YES THEN
AIR_LEAKAGE
(CERTAINTY=VERY HIGH, RELIABILITY=VERY HIGH)

Shoes wearing:

IF (" **BRAKE_APPLIC_NUMBER** ") IS HIGH THEN
SHOES_WEARING
(CERTAINTY=HIGH, RELIABILITY=HIGH)

IF (" **ABNORMAL_SHOE_NOISE** ") IS YES THEN
SHOES_WEARING
(CERTAINTY=LOW, RELIABILITY=LOW)

IF (" **BRAKING_TIME_TOO_LONG** ") IS ON THEN
SHOES_WEARING
(CERTAINTY=MEDIUM, RELIABILITY=MEDIUM)

Note: shoe wearing detector can be installed in shoes

Deformation / breaking of braking band:

IF (" **BRAKES_APPLIED** ")IS ON AND
(" **SET_AXIAL_VIBR** ") IS HIGH THEN
BRAKING_BAND_DEFORM/BREAK
(CERTAINTY=MEDIUM, RELIABILITY=MEDIUM)

IF (" **ABNORMAL_SHOE_NOISE** ") IS YES THEN
BRAKING_BAND_DEFORM/BREAK
(CERTAINTY= MEDIUM,RELIABILITY=MEDIUM)

IF (" **BRAKING_TIME_TOO_LONG** ") IS ON AND
(" **ABNORMAL_SHOE_NOISE** ") IS YES THEN

BRAKING_BAND_DEFORM/BREAK
(CERTAINTY= MEDIUM,RELIABILITY=MEDIUM)

IF (" **ABNORMAL_SHOE_WEAR** ") IS YES OR
(" **BAD_BRAKE_BAND_SURFACE** ") IS YES THEN
BRAKING_BAND_DEFORM/BREAK
(CERTAINTY= HIGH, RELIABILITY=HIGH)

Braking jacks pistons scratching:

IF (" **BRAKING_TIME_TOO_LONG** ") IS ON THEN
BRAKING_JACKS _PISTONS_SCRATCHING
(CERTAINTY= MEDIUM,RELIABILITY=MEDIUM)

IF (" **BRAKES_APPLY_ORDER** ") IS ON AND
(" **BRAKES_APPLIED** " temporised) IS OFF THEN
BRAKING_JACKS _PISTONS_SCRATCHING
(CERTAINTY= HIGH, RELIABILITY=HIGH)

IF (" **BRAKES_RELEASE_ORDER** ") IS ON AND
(" **BRAKES_RELEASED** " temporised) IS OFF THEN
BRAKING_JACKS _PISTONS_SCRATCHING
(CERTAINTY= HIGH, RELIABILITY=HIGH)

IF (" **BRAKES_APPLY_ORDER** ") IS ON AND
(" **BRAKING_SYST_AIR_NOISE** ") IS YES THEN
BRAKING_JACKS _PISTONS_SCRATCHING
(CERTAINTY= HIGH, RELIABILITY=HIGH)

Sudden braking with machine started:

IF (" **BRAKES_APPLIED_INCORRECT** ") IS ON THEN
SUDDEN_BRAKING
(CERTAINTY= VERY HIGH, RELIABILITY=HIGH)

IF (" **BRAKES_APPLIED** ") IS ON AND
(" **SPEED_LESS_20** ") IS OFF AND
(**" VOID_SPEED_T "**) IS OFF THEN
SUDDEN_BRAKING
(CERTAINTY= VERY HIGH, RELIABILITY=HIGH)

IF (" **BRAKES_APPLIED** ") IS ON AND
(" **DEFLECTOR_CLOSED** ") IS OFF THEN
SUDDEN_BRAKING
(CERTAINTY= VERY HIGH, RELIABILITY=HIGH)

IF (“ **BRAKES_APPLIED** “) IS ON AND
(“ **STAT_BREAKER_CLOSED** “) IS ON THEN
SUDDEN_BRAKING
(CERTAINTY= VERY HIGH, RELIABILITY=HIGH)

The machine does not stop:

IF (“ **BRAKES_APPLIED** “) IS ON AND
(“ **DEFLECTOR_CLOSED** “) IS OFF AND
(“ **VOID_SPEED_T** “) IS OFF (temporised) THEN
MACHINE_DOES_NOT_STOP
(CERTAINTY= VERY HIGH, RELIABILITY=HIGH)

IF (“ **BRAKING_DEFECT** “) IS ON AND
(**“ VOID_SPEED_T** “) IS OFF THEN
MACHINE_DOES_NOT_STOP
(CERTAINTY= HIGH, RELIABILITY=HIGH)

IF (“ **BRAKES_APPLIED_INCORRECT** “) IS ON AND
(“ **VOID_SPEED_T** “) IS OFF THEN
MACHINE_DOES_NOT_STOP
(CERTAINTY= HIGH, RELIABILITY=HIGH)

IF (“ **BRAKES_APPLY_ORDER** “) IS ON AND
(“ **VOID_SPEED_T** “) IS OFF (temporised) THEN
MACHINE_DOES_NOT_STOP
(CERTAINTY= HIGH, RELIABILITY=HIGH)

6. INTERFACES

6.1 OPERATOR INTERFACES

The operator interface is defined by the input and output data listed in the previous sections:

•Input:

- The user must be able to view and modify all input data.
- Also, the user must be able to define and modify the different types of cycles for the cyclic execution.

- ♦ The user must be able to define and modify the thresholds for the resulting certainty factors that will produce the triggering of an alarm in the monitoring system. So, the result of every incident must be defined and used in the DB as SCADA inputs.

• Output: The user must be able to view the list of possible incidents, related certainty factors and certainty labels. Also the list of possible causes for each incident, and justification of the deductions must be listed is the user requests it. This information must be displayed :

- ♦ When the user executes the function.
- ♦ When the user retrieves an alarm report or a certainty factor evolution graph.
- ♦ When the highest certainty incident is greater than its associated threshold (alarm detected).

The implementation of this function can be done in C or C++. This would allow us to incorporate to the program some libraries already implemented by IBERDROLA. We can also use an inference engine for the resolution of the rule handling and any tools to build the knowledge base and to fuzzyficate variables.

The simulation of the system for testing purposes can be done easily by including in the D.B. some historical incident data. So, the mechanisms to update the DB with simulation values must be provided by the system.

6.2 SYSTEM INTERFACES

The system interface will be the mentioned input and output files and the parameters given in the function call.

• The parameters will indicate the number of the function to be executed and the type of turbine (Pelton/Francis).

• The input file will contain the data listed in the input data section.
• The output file always contains the incidents, certainty factors of the incidents, labels associated with the certainty factors, lists of possible causes and justification of the deductions.
• The certainty factors must always be inserted into the DB by the system.
• The rest of the information must only be inserted when an alarm is detected.

So, the system must provide the way to access to the forecasting data stored in the DB to study the tendencies of the incidents.

7. ERROR MANAGEMENT

- Input data into normal limits (for analogic data it's specified in the fuzzy sets, for alarms 0/1).
- The resulting certainty accumulated or inferred from the application of any rule must be into normal values [0,1] during the inference process.
- To control null values or not existent (for a given period) in D.B.
- Errors must be included in separate files/tables and identified by a key.
- The tuning of the fuzzy sets and the adjusting of the certainty factors associated to each rule will be done according to the special conditions of the equipment in the plant.
- The grouping , partition or including of any rule could be done while testing if the results are not the most accurate to the working conditions of the plant, so the system must be flexible in this aspect.
- All kind of error signals from the computer must be captured in the function.

8. CONSTRAINTS

The only time constraint is the availability of data into the D.B. for the chosen period of time of *M* minutes. This means that the process for the data gathering from SCADA must insert data into the D.B. almost continuously (with a sample rate to determine)

9. HARDWARE AND SOFTWARE REQUIREMENTS

The mentioned above for the building of the Knowledge Base and the inference engine, C,C++ (Borland), Oracle, PC architecture, Windows-NT.

10. TEST PLAN

The testing of this function will be specified in the WP6 IBERDROLA documents for the Adaptation and Experimentation Specifications of the System.

Some of the features we will try to test are the following:

- Control of incorrect input data.

- To prove that for a set of symptoms related to an incident the probability of the incident is high enough.
- To prove that for a set of symptoms related to an incident the probability of the incident is higher than the rest (conclusion is clear).
- To prove that for a set of symptoms related to several incidents the probability is higher for the all the incidents implicated.
- To prove that for a set of symptoms indicating normal working there are not high probability values for any incident.
- To prove that exist any symptoms set that returns a high probability for a given incident.
- To prove that the fuzzyficated symptoms describe precisely the existence of a fault. In that item.