BODY REFERENCE : SPEC1_DIRT_REV E

TITLE :	SPECIFICATION OF IBERDROLA FUNCTIONS: "DETERMINATE DIRT LEVEL IN REFRIGERATION COILS (S23)"
SOURCE :	IBERINCO - UITESA
STATUS :	FINAL DOCUMENT
EDITOR :	JOSE MANUEL HERNANDEZ
VERIFIED BY :	
DISTRIBUTION :	ALL CONSORTIUM PARTICIPANTS
DATE :	5 SEPTEMBER 1997
NUMBER OF PAGES :	6
INTERNAL REFERENCE:	J:\DOK\13\REMAFEX\WP4\S23.DOC

CONTENTS

0. OBJECTIVE OF THE FUNCTION

The objective of the function is to detect and report the refrigeration coils obstruction level that can produce a future malfunction .

For this purpose, we will define an empirical model of the physical features of the refrigeration coils. Thus, following this model we will be able to determine in every moment the dirt level of the coils and to avoid non safety situations.

This information will be useful to evaluate the probability of some incidents.

1. FUNCTION ENVIRONMENT

The function will evaluate the confidence degree of the incident, that is, the probability of coils obstruction. So, the information presented to the user contains the level of obstruction computed by means of the model.

This evaluation will be executed every time the input data described below is selected from SCADA sensors by the monitoring system, and will use the most recent data collected by the instrumentation in real time. The results obtained from the execution of the function will be stored into the real-time DB.

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

Event	Request/Response (RQ/RS)	From	To
Monitoring Request	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 monitoring request)	RQ	Function	Function
Data Selected (on monitoring request)	RS	Data Base	Function
Perform Monitoring	RQ	Function	Function
Store Results (dirt degree)	RS	Function	Data Base

2. INPUT DATA DEFINITION

The input data will be the following signals for each refrigeration coil:

P_1 : Inlet water pressure (analogic, kg/cm^2).
P_2 : Outlet water pressure (analogic, kg/cm^2)
Q: water flow (analogic, l/s).
K_T : Theoretical constant of pressure losses
(number, kg·s^2 / cm^2·l^2 , manual input)

The different coils where we can apply this model are:

- **stator** air refrigeration coils
- **generator upper carter** oil refrigeration coil
- **generator lower carter** oil refrigeration coil
- **turbine carter** oil refrigeration coil

Input data are the following signals:

STAT_COIL_WAT_FLOW (analogic, l/s, missing by SCADA)
STAT_COIL_WAT_INLET_PRESS (analogic, kg/cm^2, missing by SCADA)
STAT_COIL_WAT_OULET_PRESS (analogic, kg/cm^2, missing by SCADA)
TURB_GB_COIL_WAT_FLOW (analogic, l/s, missing by SCADA)
TURB_GB_COIL_WAT_INLET_PRESS (analogic, kg/cm^2, missing by SCADA)
TURB_GB_COIL_WAT_OULET_PRESS (analogic, kg/cm^2, missing by SCADA)
LGB_COIL_WAT_FLOW (analogic, l/s, missing by SCADA)
LGB_COIL_WAT_INLET_PRESS (analogic, kg/cm^2, missing by SCADA)
LGB_COIL_WAT_OULET_PRESS (analogic, kg/cm^2, missing by SCADA)
UGB_COIL_WAT_FLOW (analogic, l/s, missing by SCADA)
UGB_COIL_WAT_INLET_PRESS (analogic, kg/cm^2, missing by SCADA)
UGB_COIL_WAT_OULET_PRESS (analogic, kg/cm^2, missing by SCADA)
TURB_COIL_DIRT_LEVEL_KT (number constant, kg·s^2 / cm^2·l^2 , manual input)
UGB_COIL_DIRT_LEVEL_KT (number constant, kg·s^2 / cm^2·l^2 , manual input)
STAT_COIL_DIRT_LEVEL_KT (number constant, kg·s^2 / cm^2·l^2 , manual input)
LGB_COIL_DIRT_LEVEL_KT (number constant, kg·s^2 / cm^2·l^2 , manual input)

3. OUTPUT DATA DEFINITION

The output of this function will be used to estimate the probability of a future malfunction in any of the parts whose operation depends on the working of the coils (stator, generator upper and lower carters, turbine carter). So, the dirt level is a good way to evaluate the state of the coils (percentage value for each one of them):

The results will be the following analogic signals:

- % of dirt in coil.

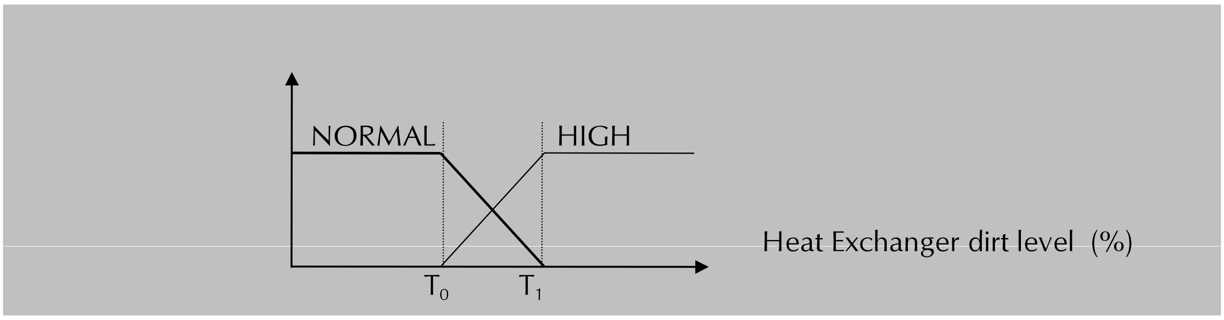
  STAT_COIL_DIRT_LEVEL(%) (Pelton)
  STAT_COIL_DIRT_LEVEL(%) (Francis)
  LGB_COIL_DIRT_LEVEL(%) (Pelton)
  LGB_COIL_DIRT_LEVEL(%) (Francis)
  UGB_COIL_DIRT_LEVEL(%) (Pelton)
  UGB_COIL_DIRT_LEVEL(%) (Francis)
  TURB_COIL_DIRT_LEVEL(%) (Pelton)
  TURB_COIL_DIRT_LEVEL(%) (Francis)

- real pressure losses constant.

  STAT_COIL_DIRT_LEVEL_KR (Pelton)
  STAT_COIL_DIRT_LEVEL_KR (Francis)
  LGB_COIL_DIRT_LEVEL_KR (Pelton)
  LGB_COIL_DIRT_LEVEL_KR (Francis)
  UGB_COIL_DIRT_LEVEL_KR (Pelton)
  UGB_COIL_DIRT_LEVEL_KR (Francis)
  TURB_COIL_DIRT_LEVEL_KR (Pelton)
  TURB_COIL_DIRT_LEVEL_KR (Francis)

- The input data must also be showed to the operator on request.

The % dirt level value will be fuzzificated and used as input data in other functions.



4. DYNAMIC BEHAVIOUR

As mentioned in section 1., the function will make some simple calculations to transform the initial data selected from the D.A.S. So, the function will be executed when retrieving data from SCADA by the monitoring system, and the results stored into the real-time DB.

5. DATA PROCESSING (ALGORITHMS)

In this section we will define the model used to determine the level of obstruction of a coil. It is based on two functions that define respectively, the theoretical and the real pressure-flow curve of a coil.

Theoretical Pressure-Flow curve:

$$\mathbf{P = K_T * Q^2}$$

$P = P_1 - P_2$

P: Pressure Drop.
P_1: Inlet water pressure.
P_2: Outlet water pressure.
Q: Water flow.
K_T: Theoretical constant.

The value of K_T is calculated as P/Q^2 when the coil is clean, or it could be given by the supplier.

Real Pressure-Flow curve:

$$\mathbf{P = K_R * Q^2}$$

$P = P_1 - P_2$

P: Pressure Drop.
P_1: Inlet water pressure.
P_2: Outlet water pressure.
Q: Water flow.
K_R: Real constant.

The value of K_R is calculated as P/Q^2 in the current situation.

- **Dirt Level (%)= $(K_R - K_T) / K_T * 100$**

6. INTERFACES

6.1 OPERATOR INTERFACES

The operator interfaces will be the same of those relative to monitoring tasks, since the result of this function will be used in monitoring tasks.

The implementation of this function can be done in C or C++.

The simulation of the system for testing purposes can be done easily by including in the D.B. some historical incident data.

6.2 SYSTEM INTERFACES

The system interfaces are the input and output data specified above.

7. ERROR MANAGEMENT

- Input data into normal limits.
- To control software errors (overflow, division by zero).
- To discard abnormal input values (deviation from the mean).
- To control null values or not existent (for a given period) in B.D.
- To control result values [0,100].
- Errors must be included in separate files/tables, identified by a key and containing error type.
- All kind of error signals from the computer must be captured in the function.
- To evaluate the relation between k_r and k_t and adjust k_t if relation is not correct.

8. CONSTRAINTS

The only real-time constraint is the availability of data in the D.B. for the chosen period of time. This means that the process for the data gathering from the sensors 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.

Among other features the following will be tested:

- Control of incorrect input data.
- To prove that under optimal conditions $k_t = k_r$.
- To prove that for different (P,Q) pairs of real values the resulting k_r keeps constant.
- To prove that the resulting values keep into normal limits [0,100].
- Control of limit values (Q=0, P=0, etc.).