MOUSE RTC

User Guide

DHI Water & Environment
Agern Allé 11
DK-2970 Hørsholm
Denmark

Tel: +45 4516 9200
Fax: +45 4516 9292
E-mail: dhi@dhi.dk
Web: www.dhi.dk and www.dhisoftware.com

CONTENTS

PART I - INTRODUCTION TO MOUSE RTC

Warranty

The warranty given by DHI is limited as specified in your Software License Agreement. The following should be noted: Because programs are inherently complex and may not be completely free of errors, you are advised to validate your work. When using the programs, you acknowledge that DHI has taken every care in the design of them. DHI shall not be responsible for any damages arising out of the use and application of the programs and you shall satisfy yourself that the programs provide satisfactory solutions by testing out sufficient examples.

[1] DHI Water & Environment is a private, non-profit research and consulting organization providing a broad spectrum of services and technology in offshore, coastal, port, river, water resources, urban drainage and environmental engineering.

PART I

INTRODUCTION TO MOUSE RTC

1. ABOUT MOUSE RTC MODULE

1.1 Key Features and Application Domain

The MOUSE RTC module features advanced Real-Time Control (RTC) simulation capabilities for urban drainage and sewer systems. It permits description of various controllable devices and makes the definition of complex operational logic for interdependent regulators fully transparent and time efficient. The following controllable devices can be specified:

- Rectangular overflow weir;
- Rectangular underflow gate;
- Pump.

The devices may be specified as settings or PID-controlled, with control function selection based on a global system analysis. Each regulator or pump operates under the control logic encapsulated into a set of simple logical rules and control functions. The system allows a schematised definition of any form and size of decision tree, featuring logical operators AND, OR, NOT and NOR, in any associate combination.

The process of selecting an appropriate control function under the current operational situations relies on the evaluation of system state conditions including measurable and derived hydraulic and WQ variables (e.g. water level, flow, pollutant concentration, level difference), devices statuses (e.g. gate blade position, pump ON/OFF) and the current control function.

The control functions range from the simplest constants for the operational variables (e.g. constant weir crest setting or constant flow set-point) to dynamic controlled variables, set in a continuous functional relation with any of the measurable variables in the system (e.g. CSO discharge set-point as a function of flow concentration or a pump START/STOP levels as functions of water level at strategic location in the system).

1.2 Software Implementation

MOUSE RTC is an add-on module to MOUSE HD Pipe Flow Model. The MOUSE RTC capabilities can be accessed only after the MOUSE license has been extended to include MOUSE RTC. For details about the DHI copy protection system and the license update procedure, please refer to the 'MOUSE Installation and Update Guide'.

MOUSE RTC utilizes the standard MOUSE Windows interface with on-line HELP facility, which has been extended to accommodate actions related to MOUSE RTC. This implies that the MOUSE on-line HELP system and documentation related to the standard versions of MOUSE are essential as support for working with this model (ref./1/ and /2/).

2. ABOUT MOUSE RTC USER GUIDE

This manual provides information related to the principles and techniques for modelling Real-Time Control (RTC) schemes in urban drainage and sewer systems. The document contains a comprehensive reference on the MOUSE's RTC capabilities, allowing users to design, model and simulate various RTC schemes within their MOUSE system.

Usage of MOUSE and its' other add-on modules is described in respective user manuals & tutorials.

This manual is divided into three units:

- **Part I: Introduction** - General information about MOUSE RTC and about this document.
- **Part II: MOUSE RTC User Manual** - Basic information about MOUSE RTC principles and techniques and extensive reference on using MOUSE RTC for modelling RTC schemes.

APPENDIX A: The PID Algorithm - A theoretical background and numerical implementation of the continuous control equation.

3. MOUSE RTC USER SUPPORT

3.1 Product Support

If you have questions or problems concerning MOUSE RTC, please consult the documentation (Installation and Update Guide and MOUSE RTC User Guide) first. Secondly, look in the README files that came with your installation. If you have access to the Internet, you may also have a look under 'Frequently Asked Questions' or 'Problems & Work-arounds' on the MOUSE Home Page. The MOUSE Home Page is located at http://www.dhisoftware.com/mouse.

If you cannot find the answers to your queries, please contact your local agent.

In countries where no local agent is present, you may contact DHI directly by mail, phone, fax or e-mail:

DHI Water & Environment

Agern Allé 11, DK-2970 Hørsholm, Denmark

Phone: +45 45 169 200

Telefax: +45 45 169 292

e-mail: software@dhi.dk

When you contact your local agent or DHI, you should prepare the following information:

- The version number of MOUSE that you are using.
- The type of hardware you are using including available memory.
- The exact wording of any messages that appeared on the screen.
- A description of what happened and what you were doing when the problem occurred.
- A description of how you tried to solve the problem.

3.2 DHI Training Courses

DHI software is often used to solve complex and complicated problems, which requires a good perception of modelling techniques and the capabilities of the software.

Therefore DHI provides training courses in the use of our products. A list of standard courses is offered to our clients, ranging from introduction courses to courses for more advanced users. The courses are advertised via DHI Software News and via DHI home page on the Internet (http://www.dhi.dk).

DHI can adapt training courses to very specific subjects and personal wishes. DHI can also assist you in your effort to build models applying the MOUSE software.

If you have any questions regarding DHI training courses do not hesitate to contact us.

3.3 Comments and Suggestions

Success in perception of the information presented in this document, together with the user's general knowledge of urban sewer systems, RTC technology and experience in numerical modelling is essential for getting a maximum benefit from MOUSE RTC. This implies that the quality of the documentation, in terms of presentation style, completeness and scientific competence, constitutes an important aspect of the software product quality. DHI will, therefore, appreciate any suggestion in that respect, hoping that future edition will contribute to the improved overall quality of MOUSE RTC.

Please submit your contribution via e-mail, fax or letter.

PART II
MOUSE RTC USER GUIDE

1. BACKGROUND

1.1 RTC in Urban Drainage and Sewer Systems

Real time control (RTC) is an active control and operation of flow regulators based on real-time information about the system state.

RTC is feasible where it proves that flexible redistribution of water in space and time contributes to the fulfilment of the specified operational objectives, based on economically and technically sound solutions. Accordingly, application of RTC to urban drainage and sewer systems may be relevant:

- where the system has substantial transport, storage or treatment capacity which is not effectively used under passive system operation;
- where typical rainfall patterns over the catchment area exhibit high degrees of spatial variability resulting in some parts of the system becoming overloaded whilst others are under utilised;
- where the urban wastewater system includes treatment processes whose performance is amenable to active, short term control;
- where the assimilative capacity of the receiving waters is variable over time.

Usually, RTC is implemented as an integral part of a rehabilitation/upgrade scheme also involving significant civil upgrading works to increase the transport, treatment or assimilative capacity of the urban wastewater system. In such circumstances, the role of RTC is to optimise the operation of both the new and the existing facilities, thereby maximising the benefit in performance terms. Where the overall objective is to achieve compliance with specified performance targets, RTC serves to minimise the scale and extent of the necessary works.

1.2 Architecture of RTC Systems

An RTC system includes **monitors/sensors**, which generate measurement values characterising states of the system. To be useful for RTC, the measurements must be available with the relatively insignificant time lag (delay). The sensors must be accurate and reliable.

The active control is performed by **regulators** - controllable movable devices (weirs, gates and valves) and pumps. Regulators may take various forms and sizes, and the regulation may be continuous within the functional range, step-wise, or discontinuous (e.g. ON/OFF, OPEN/CLOSED). The regulators may be powered mechanically, hydraulically or pneumatically.

Controllers on the basis of a pre-programmed operational strategy determine the regulator movements (the control actions). The operational strategy may consist of two parts: the

control function(s) and, if more control functions are specified, the control logic (**rules**), responsible for the selection of an appropriate control function. A control function establishes a relation between a **control variable** and a **controlled variable**. A controlled variable can be a regulator **setting** (e.g. gate position, pump START/STOP level) or some of the flow variables (e.g. water level, flow).

In the latter case, the control decisions are derived by evaluating (comparing) the current value of the controlled flow variable and the pre-defined **set-point** value. The control algorithm is based on the numerical solution of the "continuous control problem" equation and is usually termed as **PID** (**P**roportional-**I**ntegral-**D**ifferential) **control**. The actuation signal for the regulator is generated by a PID controller, which usually appears as part of the operational strategy programmed in a **P**rogrammable **L**ogical **C**ontroller (PLC).

Selection of a controlled variable is, however, subject to limits set by the variable's "controllability". Therefore, a controlled variable is usually selected among the flow variables (flow, water level), preferably in the vicinity of the regulator. As a controlled variable becomes more distant from the regulator, it becomes more difficult to control, due to time lags, diffusion and uncontrollable interference. Control of relatively distant controlled variables is difficult and often cannot give satisfactory results.

When a regulator setting is used as a controlled variable, the control algorithm is reduced to an explicit functional relation between the control variable and the regulator setting, which controls the system response indirectly. This is much simpler than PID control, but in turn, the control results are in many cases inherently inexact and only a rough flow control can be achieved. This type of control is most suitable for regulators of the ON/OFF (or OPEN/CLOSE) type, while the application to continuously controllable regulators should be carefully considered.

If the operational strategy is based on conditions local to the regulated device (for example the ON/OFF-control of a pump based on the water level in a wet well) it is called **local control**. A PLC receives signals (measurements) from local sensors and sends the control decisions (actuation signals) to the regulators. The usual situation for a sewer system is to have a number of local controllers associated with pumps. Although in strict terms this is an RTC system, it is usually not understood as such.

If the operational logic is based on global conditions, it is then called **global control.** In such a situation, a **global controller** is required. A global controller is a computer program that makes the overall system state analysis in real time and provides additional input to the local controllers, which overrides or supplements the local logic with e.g. actuator signals, or by modified set-point values.

Additional component needed is then a **data transmission system** (UHF radio, leased or dialled telephone lines, GSM, etc) to transfer data between sensors, controllers and the global controller. In connection with the global controller function, an RTC system is usually equipped with the **data management and storage** facilities (databases) and the **user interface**. This is usually termed as **SCADA** (**S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition) system.

Both local and global RTC based on sensor measurements is termed as **reactive RTC**.

The global control can be extended also to include forecast data in addition to real-time data, which is then called **predictive RTC**. The most comprehensive way to obtain forecast data is to include a model in the control system. Predictive control brings additional benefits in relatively inert systems, i.e. where the response time of an operational variable is long compared to the change of relevant disturbance (external input or control action).

1.3 MOUSE RTC vs. Real Life

MOUSE RTC simulates reactive local and global RTC systems in urban drainage and sewer networks. The software implementation is inherently a conceptualisation of real life, of which the user must be fully aware. Some conceptualisations applied in MOUSE RTC are listed below.

- The program does not distinguish explicitly between local and global RTC. Per default, all elements of a modelled RTC system are assumed available for global control.
- Sensors are specified as operational devices with definition of sensor type and position in the MOUSE network. Sensors with multiple functionality must be specified individually.
- When devices (weirs, gates and pumps) are specified as controllable in the MOUSE interface, a number of additional physical parameters about the behaviour of the structure is required to describe e.g. the allowed change rates for the state of the structure.
- The actual controllers are not specified explicitly as physical devices, but their function (i.e. operational logic as a combination of operational conditions and control functions) is associated with the respective devices.
- MOUSE RTC uses sampling and actuation (control loop) frequency identical to the simulation time step.
- Sensor readings are simulated as perfectly accurate and with 100% availability.
- Low-level logic of the pump START/STOP operation is built into the program and is controlled by the START and STOP levels.
- The PID control algorithm is built into the program and is controlled by the PID constants and by factors for weighting the terms of the numerical solution of the control equation.

2. DATA INPUT

2.1 Devices

Basic data for controllable devices (Weirs, Gates and Pumps) are specified in the standard data dialogs for the actual devices, with the "RTC" selected as the operational mode.

2.1.1 Weirs

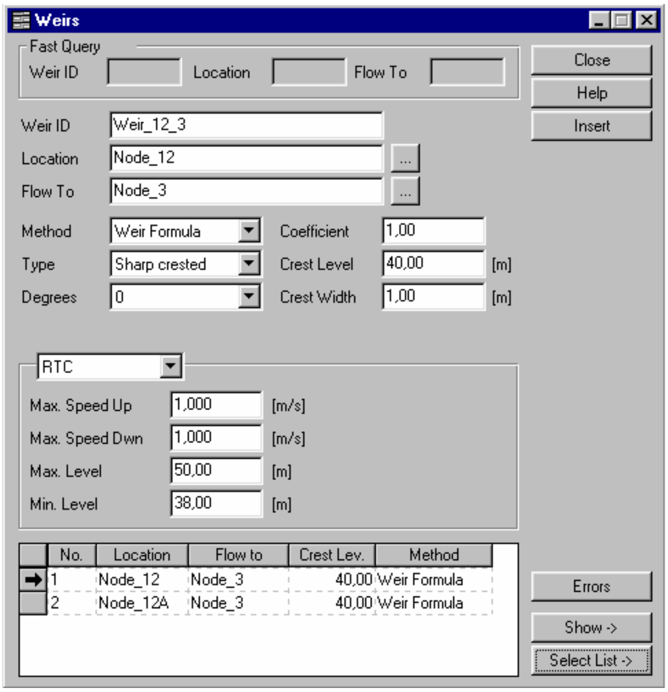


Figure 2-1 Dialog for input of Weir data

No Control / RTC
This field allows for the selection of the weir type, i.e. if the weir is controllable or not. If the weir is specified as an "RTC" weir, a number of additional parameters must be specified.

Max Speed Up
The maximum velocity for movement of the weir in upward direction.

Max Speed Down
The maximum velocity for movement of the weir in downward direction.

Max Level
The maximum elevation of the movable weir crest.

Min Level
The minimum elevation of the movable weir crest.

☞ Note: The fixed weir crest level is NOT used for a RTC weir!

2.1.2 Orifices

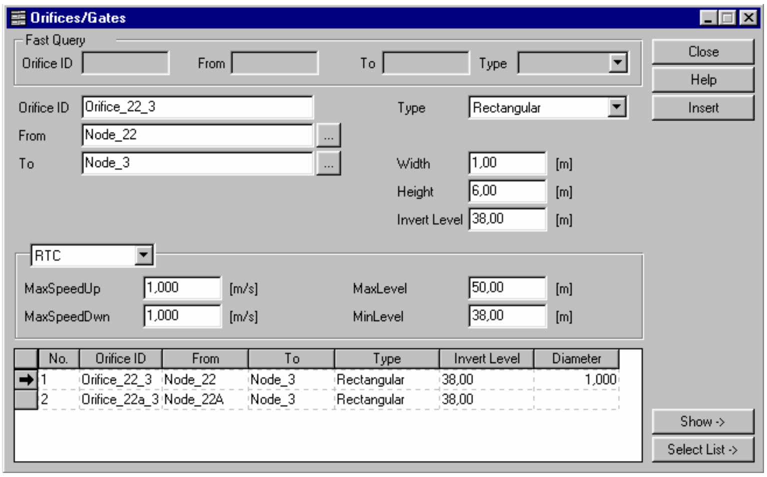


Fig 2.2 Dialog for input of Orifice/gate data.

No Control / RTC
This field allows for the definition of a controllable underflow gate. If a rectangular orifice is specified as an "RTC" device, several additional parameters must be specified for the description of the gate. The gate covers the upper part of the orifice opening, with a horizontal lower edge (blade).

Max Speed Up
The maximum velocity for movement of the gate in upward direction.

Max Speed Down
The maximum velocity for movement of the gate in downward direction.

Max Level
The maximum elevation of the movable gate.

Min Level
The minimum elevation of the movable gate.

2.1.3 Pumps

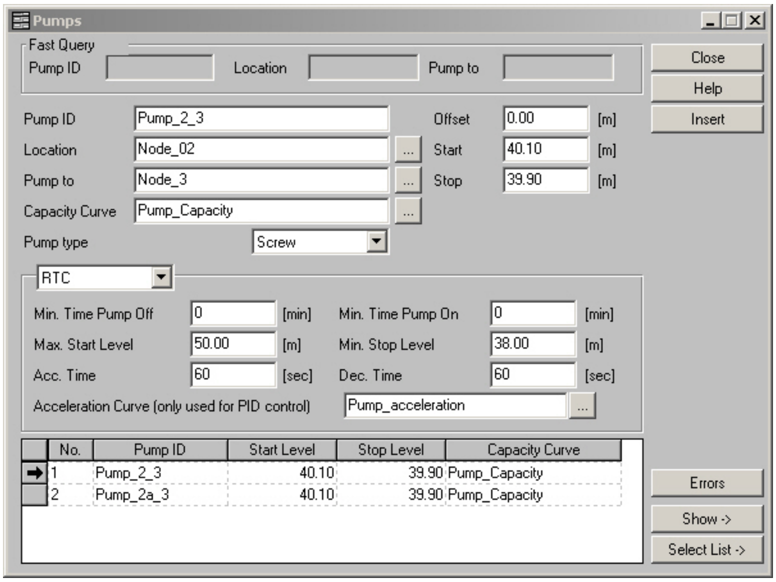


Fig 2.3 Dialog for input of pump data.

No Control / RTC
This field allows for the definition of an RTC-controllable pump. If a pump is specified as an "RTC" device, several additional parameters must be specified for the description of its operational properties.

Min time Pump OFF
The minimum time the pump has to be OFF before it can start again.

Min time Pump ON
The minimum time the pump has to be ON before it can stop.

Max Start Level

The maximum START level for the pump. If exceeded, the pump is unconditionally switched ON.

Min Stop Level

The minimum STOP level for the pump. If a lower level occurs, the pump is unconditionally switched OFF.

Acceleration time

The time used to accelerate the pump from a standstill to the maximum capacity derived from the capacity curve. This acceleration can never be exceeded in any part of the simulation.

Deceleration time

The time used to stop the pump from running with the maximum capacity. This deceleration can never be exceeded in any part of the simulation.

Acceleration Curve (for PID control only)

For PID-controls, the acceleration of the pump can be specified as dependent on the actual flow. The acceleration curve is specified in tabular data and referred to by the ID of the data set.

2.2 Sensors

A sensor is a physical device positioned somewhere in the system, which provides information of the actual value of a monitored variable.. A sensor can only monitor one variable – if more variables are measured at the same locality a corresponding number of sensors has to be described.

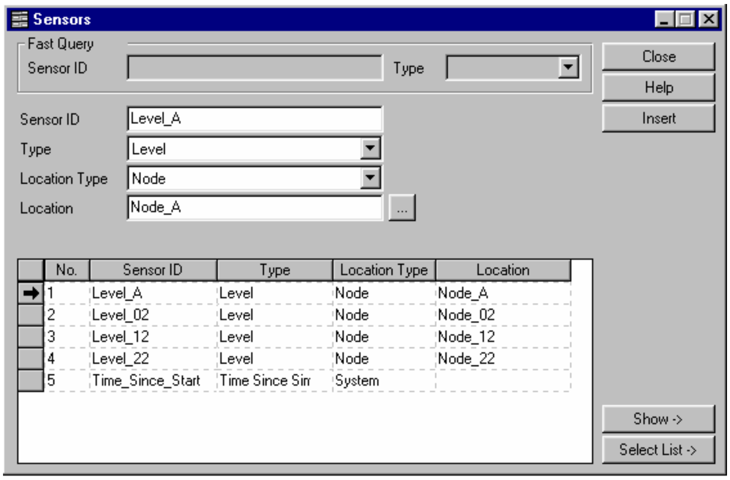


Fig 2.4 Dialog for input of sensor data.

Sensor ID
Each sensor needs a unique ID, which can be used to access the sensor information from other dialogs.

Type
The type of the sensor defines which variable the sensor measures.

Location Type and Location
Depending on the sensor type, there could be one or more location types. Accordingly, the actual location can be specified be selecting from the comprehensive list.

Component
For the "Concentration" sensor type, a measured WQ component must be specified.

2.3 Logical Conditions

A logical condition stands as a frame which demarcates the boundaries of a certain operational situation in the controlled system. This frame consists of up to four independent logical tests on the various operational variables, where the relation of the actual value (or state) of the variable (provided by a sensor) is tested against the specified threshold (limit) value (or state). The individual tests are evaluated as “TRUE” or “FALSE”, with the outcome depending on the actual variable value (or state), the threshold and the specified operator.

A logical condition is evaluated as "TRUE", only if all of its constitutive tests are evaluated as "TRUE". If only one of the tests is "FALSE", the logical condition is rejected.

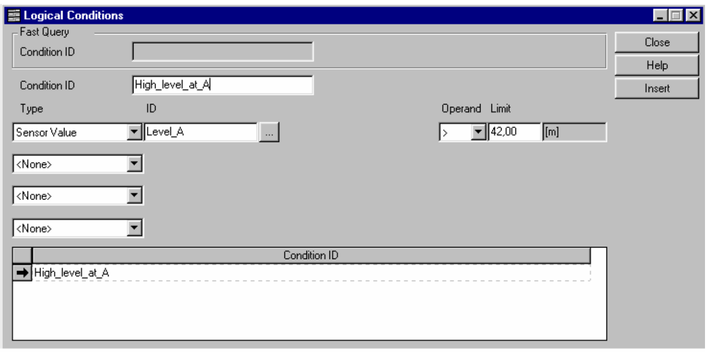


Fig 2.5 Dialog for input of logical conditions.

Condition ID
Each Logical Condition needs a unique ID, which can be used to access the logical condition information from from other dialogs.

For each of the tests included in a logical condition the following should be specified.

Type and ID
The type and identification of the variable which should be evaluated. Depending on the type, it might be necessary with two IDs –e.g. for the evaluation of difference between two sensor values.

Operator and Limit
This part of a test defines the condition which must be fulfilled for the test to be evaluated as "TRUE". In cases with the tested continuous variables, it consists of an algebraic operator (< or >) and a limit value. In cases of logical evaluation (e.g. pump ON/OFF), the test is defined simply by setting the desired device state as "TRUE" or "FALSE"

2.4 Control Functions

A control function is a functional relation between input from one or two sensors and the set-point or the setting for the controlled device.

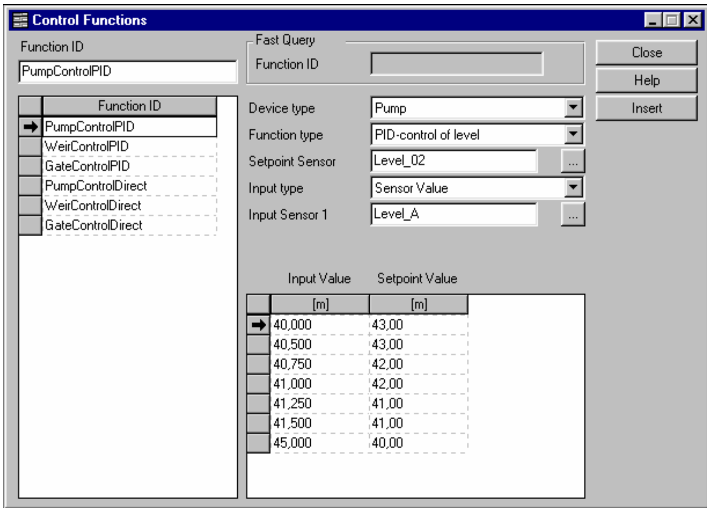


Fig 2.5 Dialog for input of control functions.

Function ID
Each control function needs a unique ID, which can be used to access the functions from other dialogs.

For each of the functions the following should be specified.

Device Type and Function Type

These two fields hold the information about the applicability of the function (device type) and which type of control it describes. The devices which can be controlled are pumps weirs and gates, and each of these types can be controlled directly or by a PID control.

If a direct control is used, the function specifies information of START- and STOP levels for pumps, or of the wanted position for gates or weirs. If a PID-control is applied, the function specifies a set-point (flow or level) anywhere in the system.

Setpoint Sensor

For functions of PID-type, it must be specified where in the system the set-point is located. For this purpose a Sensor ID must be specified.

Input Type and Input Sensor

The type and identification of the parameter which should be evaluated. Depending on the type it might be necessary with two sensor IDs –e.g. if the flow is regulated as a function of the difference between two level sensor values. If the "Constant" type is selected, a constant value is specified, instead of the sensor(s) and the functional relation.

Input and Output Values

The functional relation between the actual input value (sensor reading or a combination of two sensors) and the set-point value (or setting). The tabulated values are linearly interpolated.

2.5 PID parameter sets

MOUSE RTC includes the possibilities for PID-control of weirs, gates and pumps. Independently on the choice of the controlled variable, the PID algorithm adjusts the settings of the regulator (or outputs in case of pumps), according to the current error between the specified set-point and the actual value of the controlled variable. The following settings/outputs can be used as means of flow control:

- **Weirs**: weir crest level (*WCL*) setting;
- **Gates**: level of the bottom lip of the gate (called *WCL*) setting;
- **Pumps**: pump discharge (*Q*).

☞ **NOTE**: when using a weir as a PID regulator, the weir crest level (*WCL*) will not be used. On the other hand, the START and STOP levels for pumps will be used to start a PID-controlled pump.

The following types of controlled variables/set-points are available:

- *H*, water level in a node.

- *Q*, flow in a pipe.

☞ **NOTE**: Hot start does not work with set-points!

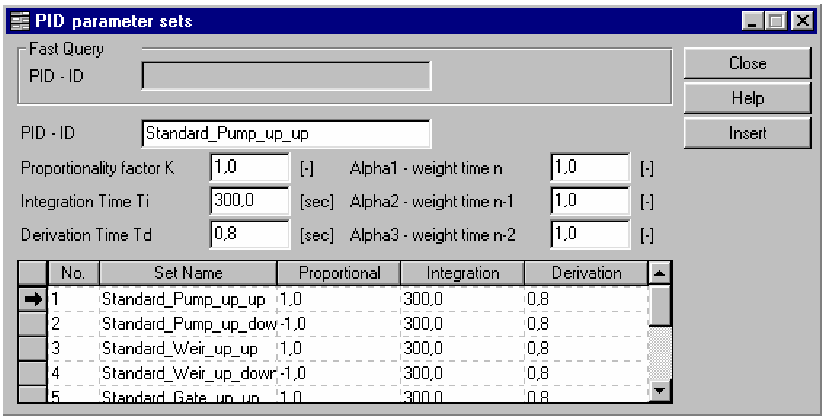


Fig 2.6 Dialog for input of PID parameters.

PID ID
Each set of PID parameter sets is identified with a unique ID which can be used to access the information from other dialogs.

Proportionality Factor, Integration Time and Derivation Time
These are the 3 main parameters for the PID control. (refer to Appendix A: "PID Algorithm" for a detailed description).

Alpha-1, Alpha-2 and Alpha-3
Weighting factors for time level n, n-1 and n-2.

2.5.1 Calibration of the PID constants

Tuning of the PID constants (Ti, Td and Kd) is not a straightforward task. Understanding of the theoretical background and the numerical solution of the control equation would be beneficial in this process (refer to Appendix A: PID Algorithm). The following may be used as an elementary guideline.

Typical values of the PID constants and weighting factors

T_i:		300 sec.
T_d:		0.8 sec.
K:	Pumps and gates:	
	- setpoint downstream:	**1.0**
	- setpoint upstream:	**-1.0**
	Weirs	
	- setpoint downstream:	**-1.0**
	- setpoint upstream:	**1.0**
α_1		1.0
α_2	Gates and weirs:	**0.7**
	Pumps:	**1.0**
α_3		1.0

☞ **NOTE**: The sign on the K-factor is very important – if it is wrong it will cause the control function not to work at all since the device will typically move to one of the extreme positions and stay there till the end of the simulation.

Figures 2.7 through 2.9 show examples of how the actual variable (flow or water level) can fluctuate around the set-point as consequence of various choices of the PID constants. Each figure has three different graphs depending on whether the constant is too high, too low or adequate.

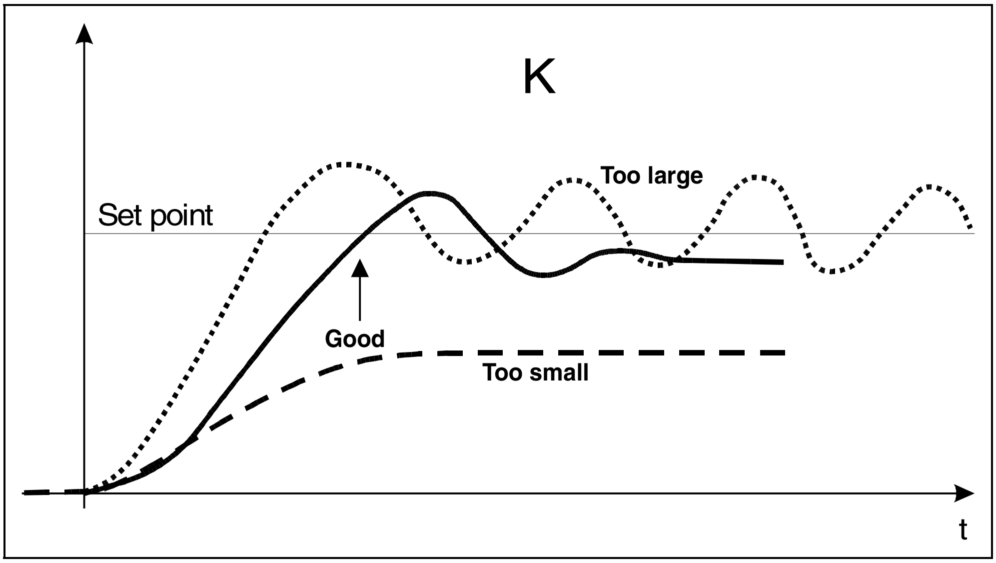


Fig. 2.7 Fluctuations around the set point depending on the size of the proportionality factor, K.

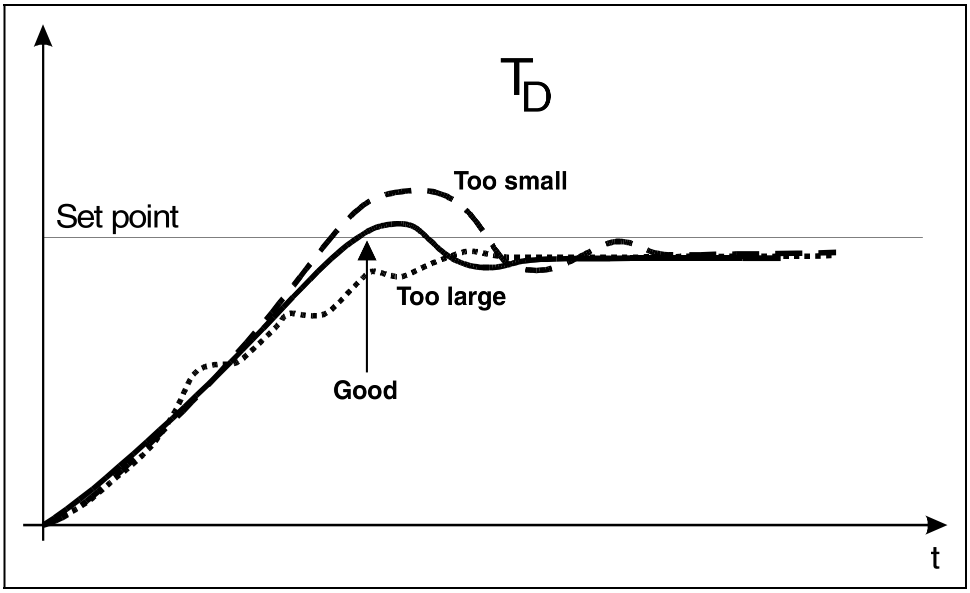


Fig. 2.8 Fluctuations around the set point depending on the size of the derivation time, T_D.

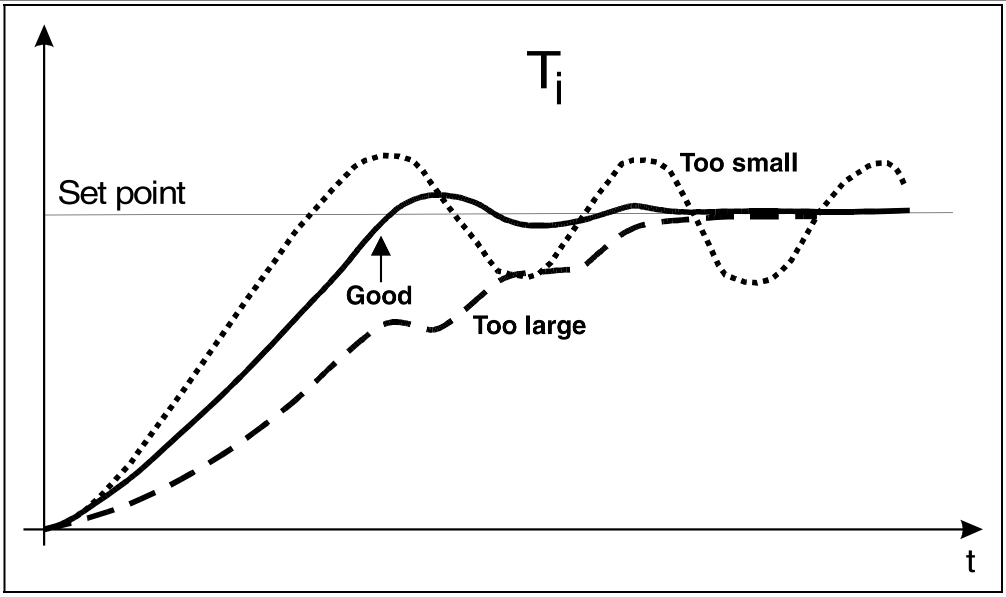


Fig. 2.9 Fluctuations around the set point depending on the size of the integration time, T_i

2.6 Controlled Devices

In this dialog the operational control logic for controllable devices is specified. The control is specified as a set of rules, linking the logical indirectly conditions and the control functions. The rules are evaluated sequentially following the rules list sequence.

Evaluation of a logical condition belonging to a rule as "TRUE", leads to the selection of the specified control function. On the contrary, if a logical conditions is "FALSE", the evaluation proceeds to the next rule on the list.

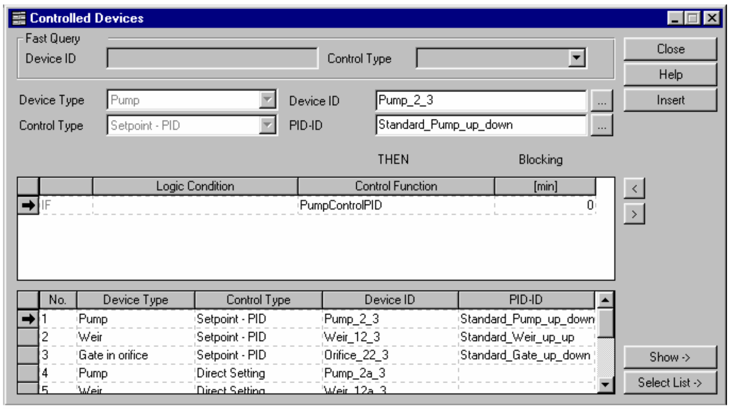


Fig 2.10 Dialog for input of Controlled devices.

Device Type and Device ID

Device Type and ID identifies the device which should be controlled. The actual description of the device should be made in the relevant menu for pumps, weirs or orifices.

Control Type and PID-ID

The control type can be either direct setting,PID-control of set-point or no control. If PID-control is used a set of PID-parameters should be selected by the PID-ID.

Control rules

Any number of rules can be specified to control the device. The statements are evaluated sequentially starting from the top. This means that appropriate sequence of rules is essential for the achievement of the desired control logic.

If no logical condition is specified, the rule is unconditionally evaluated as "TRUE". This implies that the last rule in the sequence must not include any logical conditions in order to ensure a selection of a "default" control function if all specified conditions are found "FALSE".

When specifying the rules a few constraints should be notified:

All control functions used to control a specific device must be of the same function type – corresponding to the specified Device Type and Control Type.

For PID-control, all control functions must refer to the same set-point sensor – change of the set-point sensor during simulation is not allowed

For time dependent control (Input sensor of type "Time since start of simulation") only ONE rule can be specified. The menu allows more functions to be inserted, but only the first one will be used during the simulation.

When starting a simulation, the system checks if these conditions are fulfilled and in the case of any violation, the simulation will not start.

3. MOUSE RTC COMPUTATIONS

3.1 General

A MOUSE RTC computation is started from the usual 'Pipe Flow Computation' dialog and will be performed if the checkbox "Real Time Controls" is marked.

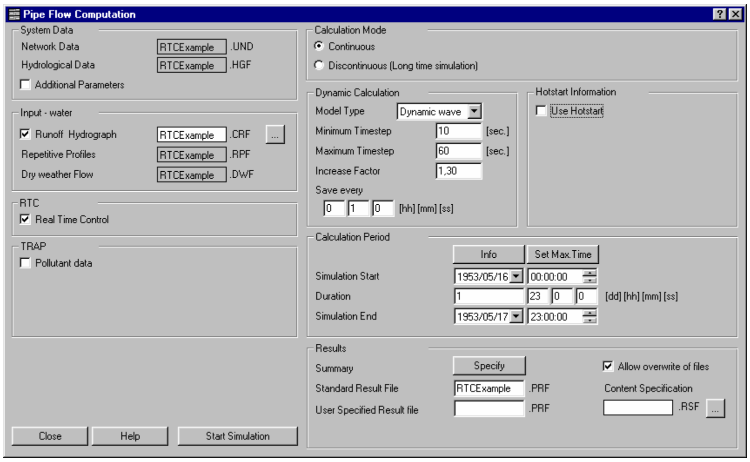


Figure 3.1 Pipe Flow computation dialog with Real Time Control

4. USER WRITTEN CONTROL

4.1 General

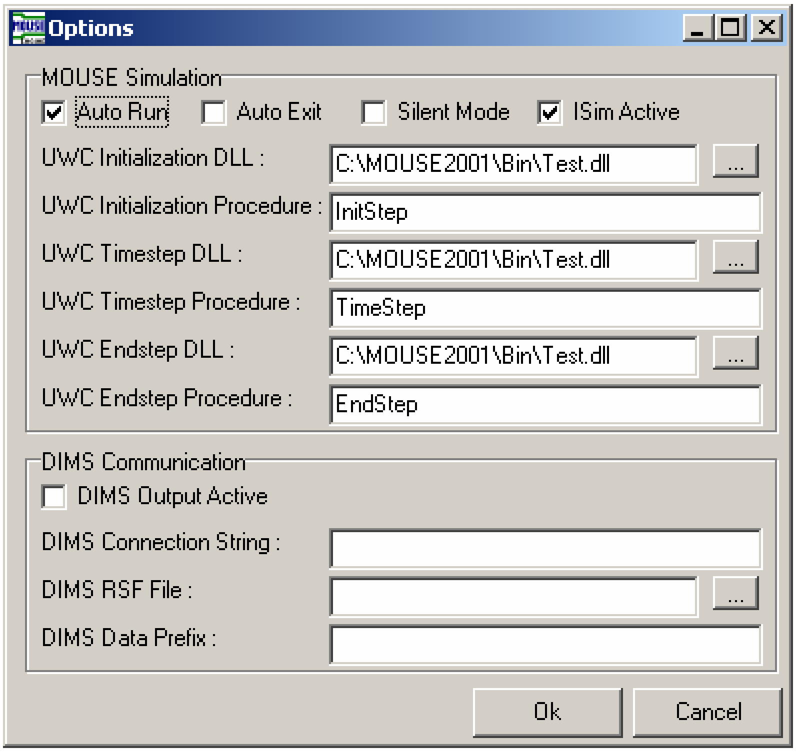
MOUSE 2002 and newer versions support User Written Control (UWC) for advanced RTC control of the system. This gives the user the possibility of controlling almost all aspects of MOUSE each time step and thereby implementing advanced control strategies in the system. UWC is only recommended for the advanced computer user since it involves programming experience. The currently supported language is Delphi the procedures and functions can however be called from any programming language including Visual Basic, C++ and others that support generating DLLs.

The user can execute code at 3 different steps in the code. The Initialization step is for executing code opening, preparing or reading files needed in the simulation. The Time step is the step in which the control strategies are run. The End step is the step in which files opened in the Initialization step can be closed. The 3 steps are defined under the mouse604.exe Simulation | Options dialog, where a filename and a procedure name are specified for each step under UWC. The compiled DLL with the UWC must be located in the same folder as the mouse_hd.dll file.

There are currently over 140 different procedures and functions exported from MOUSE ranging from setting PID constants runtime to retrieving the flow in a specific grid point in a link. Using the methods require the user to include the MOUSEDLL.pas file into the DLL Delphi project. An example of the use of this is listed below.

```
library Test;

uses
  SysUtils,
  Classes,
  MOUSEDLL; // <- Important

{$R *.res}

var
  OutFile : TextFile;

procedure InitStep;
begin
  AssignFile(OutFile, 'C:\MOUSEOutput.txt');
  Rewrite(OutFile);
end;

procedure TimeStep;
var
  NodeLevelString : String;
  MOUSETimeString : String;
begin
  NodeLevelString := Floattostr(GetNodeLevel('Node_11'));
  MOUSETimeString := FormatDateTime('YYYY-MM-DD hh:mm:ss', MOUSETime);
  Writeln(OutFile, NodeLevelString+' '+MOUSETimeString);
end;

procedure EndStep;
```

```
begin
  CloseFile(OutFile);
end;

exports
  InitStep,
  TimeStep,
  EndStep;

end.
```



The example above simply writes the water level in a node and the corresponding time to a file, but the possibilities are endless.

The next example is the RTCExample provided in the installation where the actual control is handled from UWC. The RTC control in the UND file is not changed for this, which means the UWC RTC overrides the RTC in the UND. This provides a way of overriding standard RTC in e.g. emergency situations in the network. Alternatively the UWC RTC can be used alone. In this case the Control Functions and Logical Conditions are not needed for the Controlled Devices, since the actual control is handled externally. In addition to this, the control type number for the 3 PID controlled devices is changed to 5. This change is done using a text editor.

```
library RTCExample;
```

```
uses
  SysUtils,
  Classes,
  MOUSEDLL;

{$R *.res}

procedure TimeStep;
var LevelA, Level02, Level12, Level22,
    SetPoint1, Measured1, SetPoint2, Measured2, SetPoint3, Measured3,
    StartLevel, StopLevel, GatePosition, WeirPosition : Double;
begin
  LevelA    := GetSensorValue('Level_A');
  Level02   := GetSensorValue('Level_02');
  Level12   := GetSensorValue('Level_12');
  Level22   := GetSensorValue('Level_22');

  //Pump_2_3
  SetPoint1 := -0.6*LevelA+67;
  SetPumpPIDSetpoint('Pump_2_3', SetPoint1);
  Measured1 := Level02;
  SetPumpPIDMeasurement('Pump_2_3', Measured1);

  //Weir_12_3
  SetPoint2 := -0.6*LevelA+67;
  SetWeirPIDSetpoint('Weir_12_3', SetPoint2);
  Measured2 := Level12;
  SetWeirPIDMeasurement('Weir_12_3', Measured2);

  //Orifice_22_3
  SetPoint3 := -0.6*LevelA+67;
  SetGatePIDSetpoint('Orifice_22_3', SetPoint3);
  Measured3 := Level22;
  SetGatePIDMeasurement('Orifice_22_3', Measured3);

  //Pump_2a_3
  StartLevel := LevelA+0.2;
  StopLevel  := LevelA;
  SetPumpStartLevel('Pump_2a_3', StartLevel);
  SetPumpStopLevel('Pump_2a_3', StopLevel);

  //Weir_12a_3
  WeirPosition := LevelA;
  SetWeirPosition('Weir_12a_3', WeirPosition);

  //Orifice_22a_3
  GatePosition := LevelA;
  SetGatePosition('Orifice_22a_3', GatePosition);
end;

exports
  TimeStep;

begin
end.
```

A documentation of the different procedures and functions can be found in the MOUSEDLL.pas file. DHI does not provide support in the Delphi Programming language. We are however open to suggestions on improving and adding functionality to this

5. REFERENCES

/1/ DHI (2000): MOUSE - Short Introduction and Tutorial (Version 2000), DHI, Hørsholm, Denmark.

/2/ DHI (2000): MOUSE - User Guide (Version 2000), DHI, Hørsholm, Denmark.

APPENDIX A : THE PID ALGORITHM

At each simulation time step, the set-point is evaluated against the actual value of the control variable (flow or water level, depending on the set-point type).

The actual *WCL* or Q_{pump} is determined from the following equation:

$$u = K \cdot \left(e + \frac{1}{T_i} \int_0^{T_i} e\, dt + T_d \frac{de}{dt} \right) \qquad \textbf{(1)}$$

where:

u	=	the output signal (*WCL* or Q_{pump}),
e	=	the error to be minimised,
K	=	the proportionality factor,
T_d	=	the derivation time,
T_i	=	the integration time.

For numerical solution, a discrete form of this equation is required. By use of the Laplace transformation one obtains:

$$U(s) = K \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \cdot E(s) \qquad \textbf{(2)}$$

where:

$U(s)$	=	the Laplace transform of u,
$E(s)$	=	the Laplace transform of e.

By use of the backward Euler transformation:

$$s = \frac{z - 1}{T_s \cdot z} \qquad \textbf{(3)}$$

where T_s is the sampling period, one obtains the following discrete representation of the PID equation:

$$\begin{aligned} u(n) = {} & \alpha_1 \cdot K \cdot \left(1 + \frac{T_s}{T_i} + \frac{T_d}{T_s} \right) \cdot \left(y_{ref}(n) - y(n) \right) \\ & - \alpha_2 \cdot K \cdot \left(1 + \frac{2T_d}{T_s} \right) \cdot \left(y_{ref}(n-1) - y(n-1) \right) \\ & + \alpha_3 \cdot K \cdot \left(y_{ref}(n-2) - y(n-2) \right) \cdot \frac{T_d}{T_s} \\ & + u(n-1) \end{aligned} \qquad \textbf{(4)}$$

where:

u	=	value of the control variable, i.e. *WCL* or Q_{pump}
K	=	the factor of proportionality
T_i	=	the integration time
T_d	=	the derivation time
T_s	=	sampling period, i.e. simulation time step
y_{ref}	=	the set point for the regulated variable
$\alpha_1..\alpha_3$	=	factors for weighing each time step (index n, n-1 and n-2)
y=	actual value of the regulated variable	

Indexes *n*, *n-1* and *n-2* denote the current, the previous and the second previous time step, respectively.

The weight factors α_1 .. α_3 have been added to the PID algorithm in order to give the user more ways of stabilising the algorithm in case of instability problems.