LVTrans

Manual
October 2023

dr.ing Bjørnar Lona Svingen

CONTENT

Page

1 UPDATES TO THE MANUAL

2003 : First version for Statkraft (Norwegian)
…
December 2016 : Updated version based on new versions of LabVIEW and LVTrans (Norwegian)
December 2022 : First English version. Largely translation from December 2016, but updated to a newer version of LabVIEW and LVTrans.
January 2023 : Large fix-up of the manual to reflect the current version.
January 2023 : Update to LVTrans version 02.20.05 and included explanation of the "New update or copy powerplant.vi"
October 2023 : Update to LVTrans version 02.22.11 and included a "must read" explanation of the "New update or copy powerplant.vi" to make earlier versions work.

2 SUMMARY

LVTrans is a simulation- and dimensioning software for fluid filled piping systems. It is general in nature and can be used to simulate all kinds of systems consisting of liquid filled pipes. LVTrans also includes open channel elements. It has also been used to calculate acoustics in gas pipes. LVTrans has been used on several projects in the North Sea in connection with the oil and gas industry (gas pipeline acoustics, firewater systems amongst other things). For hydro power plants, several interactive simulators have been made. It is also the core transient simulation software for an online predictive governing system for the largest power plant in Norway, Tonstad powerplant with 960 MW installed power. However, the largest utilization by far is as a general engineering tool for transient and stability analysis of hydro powerplants. It is used by NTNU/SINTEF and Statkraft, and by Rainpower/Hymatek.

The “brain” in LVTrans is the method of characteristics (MOC), used to solve the partial differential equations governing “elastic” piping systems, where elastic means the water hammer equations, also called the Allievi equations. With MOC the solution is approximately analytically exact, and it is very fast. No separate steady state solver exists in LVTrans. Any steady state solution is merely a special case of the transient solution. In contrast to most other similar software where start and stop of the simulation are typically set up front, LVTrans is fully interactive, running real-time or (usually much) faster. Everything is done interactively, and the solution is always fully transient. Stability analysis in the s-plane is done by FFT.

LVTrans uses LabVIEW’s graphical interface and is object oriented. Every element is an icon connected with wires. In this way literally limitless complexity can be made. By using LabVIEW, all the graphs and dials are professionally made, and is continuously updated by National Instruments. LVTrans works in LabVIEW version 2021 or later.

The source code is open and editable. Anyone can look at the code, edit it, change elements or make new ones. For general hydropower simulations, no editing is normally necessary, in fact no knowledge of LabVIEW is even necessary. To edit and change the code does require knowledge of LabVIEW programming and is more of an “expert feature”. For instance, one can make oneself a new type of governor, or a special pipe element[1]. LVTrans can therefore be used as a general simulation software for hydropower or used as a more experimental software for research on the effect of new elements.

LVTrans is “sporadically under constant development” and will never be a fully “finished” software such as MS Word for instance. The whole thing is made in LabVIEW[2], and it is run in LabVIEW (although standalone executable simulators can be made). A requirement is an installation of LabVIEW base or community edition, version 2021 or newer.

A good way to look at LVTrans is extracts from the book “Fluid transients in systems” by Wylie and Streeter but programmed in LabVIEW. For hydropower plants, no knowledge of LabVIEW or the physics of water hammers are needed, but when going beyond that, then LabVIEW as well as fundamental knowledge of the Method of Characteristics (MOC) is needed. A handbook about fluid resistance in different pipes and equipment is also a good companion.

Note! Some figures may not be updated to the newest version, since it is mostly cosmetic.

---

[1] See the license conditions for use and editing of the code.

[2] The plan from 2023 onward is to transfer all the core numerical code to a separate and independent C++ library (a core MOC “engine”), while using LabVIEW for HMI and IO.

3 LICENSE

LVTrans is "Open Source" and the conditions are given in the license. The license is shown below:


```
### BSD License Begin ###

http://www.opensource.org/licenses/bsd-license.php

Copyright (c) 2001-2023, Bjørnar Svingen <bjornar.svingen@gmail.com>
Copyright (c) 2003-2007, SINTEF Energi AS <energy.research@sintef.no>

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

    * Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
       documentation and/or other materials provided with the distribution.
    * Neither the name of Sintef Energi AS, nor the names of its
contributors may be used to endorse or promote products derived
       from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### BSD License End ###
```

3. OFTEN USED FORMULAS AND EQUATIONS

There are some equations and formulas that are used all the time in LVTrans.

3.1 SINGULAR LOSSES

$$C_v = \frac{Q_0^2}{2H_0} = \frac{A^2 g}{\xi} \tag{1}$$

C_v is used to value singular losses in LVTrans. For instance, if Q_0 and H_0 is known for a valve (from tables or measurements), then the C_v value can be calculated. If no values are known, then the loss coefficient, ξ, can be used to calculate C_v. ξ is used in most handbooks for loss coefficients.

3.2 ANGULAR MOMENTUM

LVTrans uses the acceleration constant T_a instead of the angular moment of inertia. The reason is that T_a has a small numerical range compared with the angular inertia. T_a for a governed turbine is usually within the range of 4-8 s. For ungoverned small hydro, T_a is typically 1-2 s. If one must guess, using T_a therefore causes much smaller errors than using J or GD^2. For first time estimates using T_a is therefore very practical. If the angular momentum is known, then T_a is calculated with:

$$T_a = \frac{J\omega_0^2}{P_0} = \frac{GD^2\omega_0^2}{4P_0} = \frac{\left(\frac{GD^2}{4}\right)\left(N\frac{\pi}{30}\right)^2}{P_0} \tag{2}$$

Here $\omega_0 = \frac{\pi}{30}N$, where N is the rotations per minute (RPM), P_0 is the power at the corresponding ω_0, and GD^2 is the inertia on the nameplate. GD^2 is one single number and is given in ton-meter-squared. The SI form, J is used more and more, but GD^2 still dominates.

3.3 TIME CONSTANTS FOR WATER

The time constant for water, T_W for the water in the turbine is:

$$T_W = \frac{QL}{gHA} = \frac{Q\sum\frac{L}{A}}{gH} \tag{3}$$

The same is also true for water in conduits. The middle expression is for one conduit, while the right expression is for a series of connected conduits. For a Francis turbine element, T_W wil be in the range of 0.01 – 0.1 s. T_W for a Francis turbine represents the water in the spiral casing, turbine and draft tube. The draft tube can however also be modelled as a general pipe element.

3.4 EQUIVALENT DIAMETER

LVTrans often use the “equivalent diameter” instead of the area. That is, if a pipe has a cross sectional area, A, then the equivalent diameter becomes:

$$D = \sqrt{\frac{4A}{\pi}} \tag{4}$$

3.5 FRICTION FACTOR

The friction factor or friction coefficient for pipes and tunnels can be very hard to estimate correctly without measurements. Good approximations are:

- Steel pipes: f = 0.01
- Blasted tunnels: f = 0.03 – 0.06
- Tunnels made by TBM: f = 0.015-0.02

Here f is the factor used in the Moody diagram, often expressed as λ.

LVTrans also can calculate f continuously and automatically. This is done using Håland’s formula. The use of this formula is switched on and off in the input file for the pipe/tunnel. This is done by setting the parameter *full_moody?* to the value *TRUE* (*full_moody? = TRUE*). Håland’s formula:

$$\frac{1}{f^{1/2}} = -1.8 \log\left[\frac{6.9}{Re_D} + \left(\frac{\epsilon/D}{3.7}\right)^{1.11}\right] \tag{5}$$

The additional parameter in the input file is ϵ. ϵ is the absolute roughness in meters. This is usually very small. For steel the value is approximately 0.05 mm (0.05e-3 m). For blasted tunnels a value of 0.2 – 0.5 m is usual. This can be useful in certain circumstances.

3.6 MANNINGS NUMBER

Manning’s number is used for the free surface elements. Manning’s number is defined as:

$$Q = \frac{k_n}{n} A R^{2/3} S_f^{1/2} \tag{6}$$

Her er n is the Manning’s number, k_n = 1.0 (SI units). Typical values for n is 0,013-0,015 for concrete and 0,025-0,045 for blasted rock.

NB! This number is the inverse of the Manning’s number often used for blasted rock tunnels in hydro power. No good explanation for this has been found.

To find the correct Manning’s number without measurements is difficult. One simply has to use best guess.

Converting manning (n) to from Darcy is done using:

$$n = R_H^{1/6}\sqrt{\frac{f}{8g}} \tag{7}$$

$$f = 8g\left(\frac{n}{R_H^{1/6}}\right)^2 \tag{8}$$

Where R_H is the hydraulic radius:

$$R_H = \frac{A}{P} \tag{9}$$

3.7 THOMA CROSS SECTIONAL AREA

Thoma cross sectional area of surge shafts is the minimum surface area required for power control with power feedback to be stable. The Thoma cross sectional area is typicalle multiplied with a safety factor of 1.5

$$A_{th} = \frac{Q_0^2 \sum \frac{L}{A}}{h_f 2gH_0} \tag{10}$$

Here h_f is the friction loss in the tunnel. Q_0 and H_0 are the actual flow and head respectively.

3.8 THE PERIOD OF OSCILLATION OF THE SURGE SHAFT

$$T_{SS} = \frac{2\pi}{\sqrt{g\frac{1}{A_S}\frac{A}{L}}} \tag{11}$$

Where A_S is the surface area of the surge shaft and A is the cross sectional area of the tunnel.

3.9 MAX UPDRAFT IN SURGESHAFT

$$Y_{max} = \frac{Q}{A}\sqrt{\frac{LA}{gA_S}} \tag{12}$$

3.10 EQUIVALENT SURFACE AREA FOR AIR CHAMBER

$$A_{eqv} = \frac{1}{\frac{1}{A_S} + \kappa \frac{H_a}{V}} \tag{13}$$

Where A_s is the cross sectional surface area of the air chamber, H_a is the pressure head, V is the air volume and κ is typically 1.4 for air.

3.11 THE ALLIEVI CONSTANT

$$h_w = \frac{Q_0 a}{2gAH_0} \tag{14}$$

Where a is the pressure wave velocity, Q_0 and H_0 are the actual flow and head respectively, and A is the cross sectional area of the pipe/tunnel.

3.12 THE WATER INERTIA CONSTANT

$$T_w = \frac{LQ_0}{AgH_0} \tag{15}$$

The water inertia constant is the acceleration time for a non-elastic conduit.

3.13 THE REFLECTION TIME

$$T_R = 2T_e = \frac{2L}{a} \tag{16}$$

The reflection time is the time it takes for a pressure wave to go from one end and back again. T_e is the time a pressure wave takes from one end to the other end.

4 INSTALLATION

The program is zipped in a zip file. This file must be un-zipped. This is done in Windows with the "unzip" function or with another "zip program", for instance WinZip or 7-zip. The program will be within the folder, for instance LVTrans-2014-01.09.11. In general LVTrans-ZZZZ-xx.xx.xx where ZZZZ is the LabVIEW version used (this or newer version is required) and xx.xx.xx is the running LVTrans version.

The folder where LVTrans is placed can preferably be “mass compiled” first. This is not strictly necessary but will increase speed and responsiveness when using LVTrans later on, especially if a newer version of LabVIEW is used than the current LVTrans version. This can be done with following procedure:

1. Start LabVIEW from the start menu in Windows (Do NOT start directly by double clicking on an LVTrans file).
2. In the Labview application chose Tools -> Advanced -> Mass Compile
3. Point to the folder where the current LVTrans is situated (for instance the folder named LVTrans-2014_01.09.11 where LVTrans is unzipped. Make sure this folder is in the "Look in” window.
4. Push the button named ”Current Folder”
5. Push the button named ”Mass Compile”
6. After a few seconds or minutes depending on the PC it will be finished. When this is done push the button named ”done” (the "log" is irrelevant, for most cases).

The elements themselves are objects that need to be installed in the menu system in LabVIEW. This can be done as follows:

1. Start LabVIEW from the windows Start menu.
2. Open ”Tools -> Advanced -> Edit Palette Set…”
3. Three new windows pops up. This can take a little while. It's the window named ”Functions” that LVTrans shall reside.
4. Right click at a free space in ”Functions” and chose ”Insert Sub Palette…”
5. Hook ”Link to a directory”
6. Point to the folder where LVTrans resides. Point further to the sub-folder named "LVTrans" and then chose ”current folder” when there.
7. Push the button ”save Changes”.
8. LabVIEW may now use some time to rearrange things. When it is finished, it is a good idea to close LabVIEW and then open it again.
9. The block diagram is where the menu will be found. If the menu is not visible, right click anywhere on the block diagram to make it visible.
10. If everything is done correctly, the menu will be as shown in Figure 4.1

If one wants to delete this menu, do according to the points 1-3 above, right click on the LVTrans-menu, and chose ”delete subpalette”. This can be useful when a new version is ready, linking to the new version instead of the old version. Figure 4.2 shows a “typical” and very simple hydropower system.

Several different versions of LVTrans can be on the hard disk, but it can be useful to just use the latest version, and simply copy and update older user-files. The older folders can then be zipped before deleting them. When they are zipped, LabVIEW will not see them, and will not accidentally use older files on newer version.

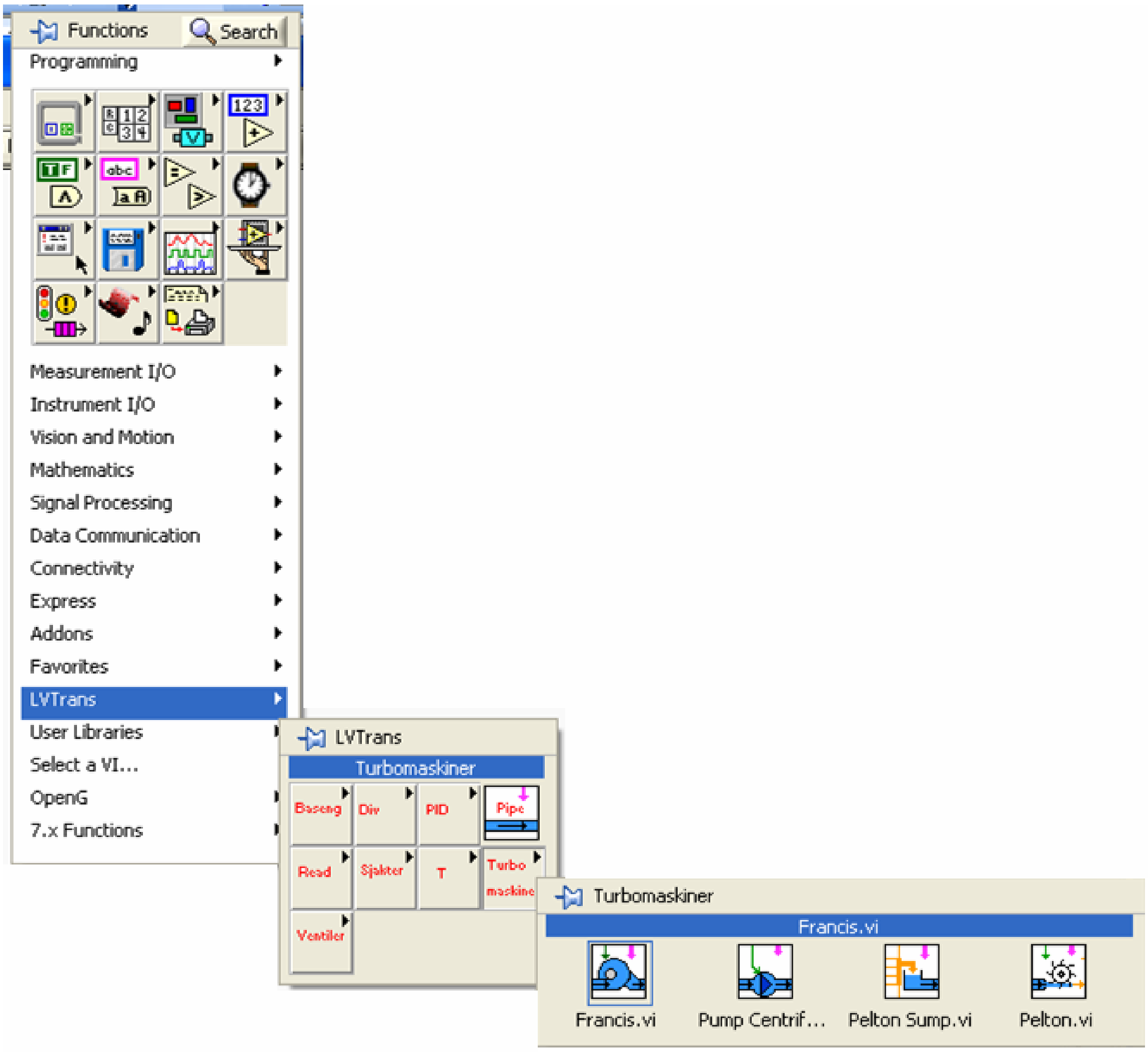


Figure 4.1 Menu system with all (LVTrans) elements

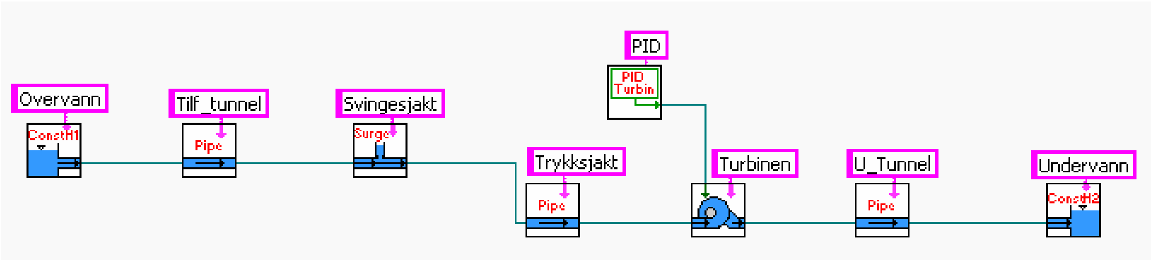


Figure 4.2 A simple hydropower system

5 FILE STRUCTURE

The entire LVTrans resides in a single folder as shown in Figure 5.1 The main folder inside that folder is called LVTrans, while the entire installation is within the folder LVTrans-2014_1.9.11 (for instance). When updating, a new folder is copyed in, LVTrans-2014_1.9.12 for instance. The numbering convention is as follows:
LVTransX_y.z.a where X er LabVIEW versjon and y.z.a are version numbers.

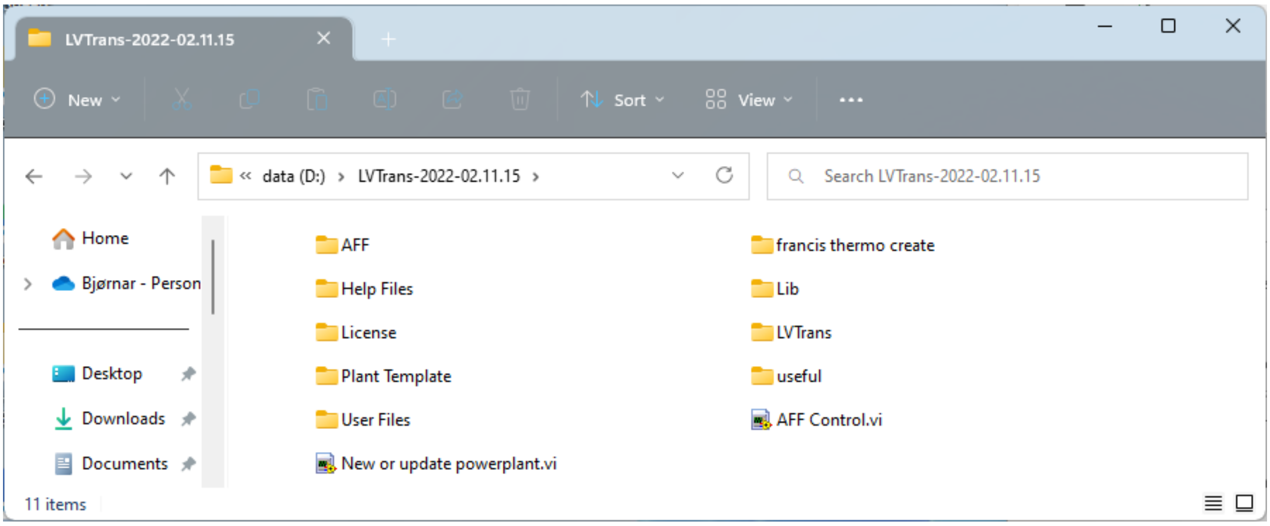


Figure 5.1 Main LVTrans folder

Several sub folders exist. Most of these sub folders contains no user changeable stuff (unless one know what one is doing), but the folders called "Help Files", "useful" and "User Files" are useful when running the program. The folder named "Lib" consists of vi's necessary for the program to run. In the folder named LVTrans, are all the elements (the one the menu system is linked to in the previous chapter). The folder "User Files" is where all the user power plants is put. Making a new power plant from scratch, and it will end up there.

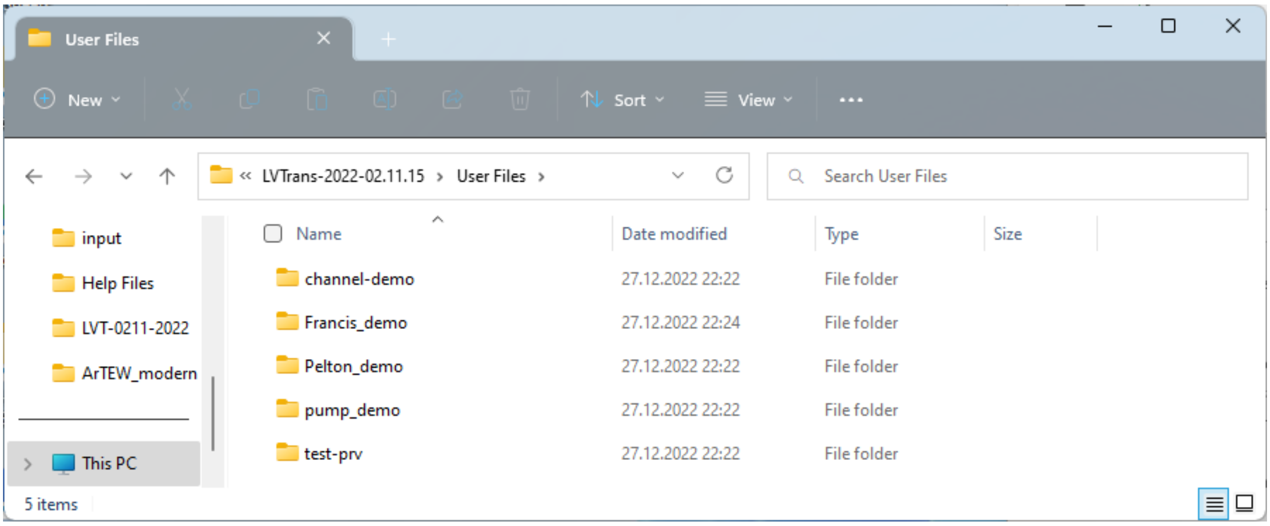


Figure 5.2 Sub folders in the User Files folder

When the file ”New or update powerplant” is opened and run, and the button "create new power plant" is pushed, se Figure 5.1, a new and fresh sub folder is created inside the User Files folder (for instance Francis_demo). In this folder, a folder named Data is automatically created, along with some program file (vi) called Franics_demo.vi and one that is called Francis_demo_system.vi.

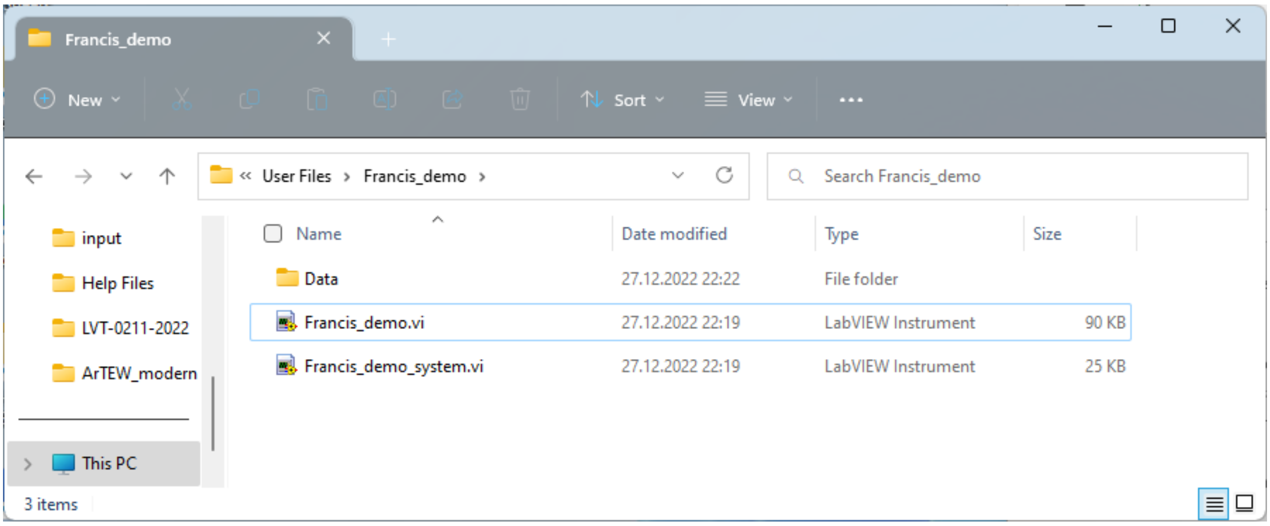


Figure 5.3 A power plant inside the User Files folder

It’s the xxx_system.vi (Francis_demo_system.vi in this example), and the data files in the folder Data that one edits and create a power plant with. How this is done is explained in the next chapter.

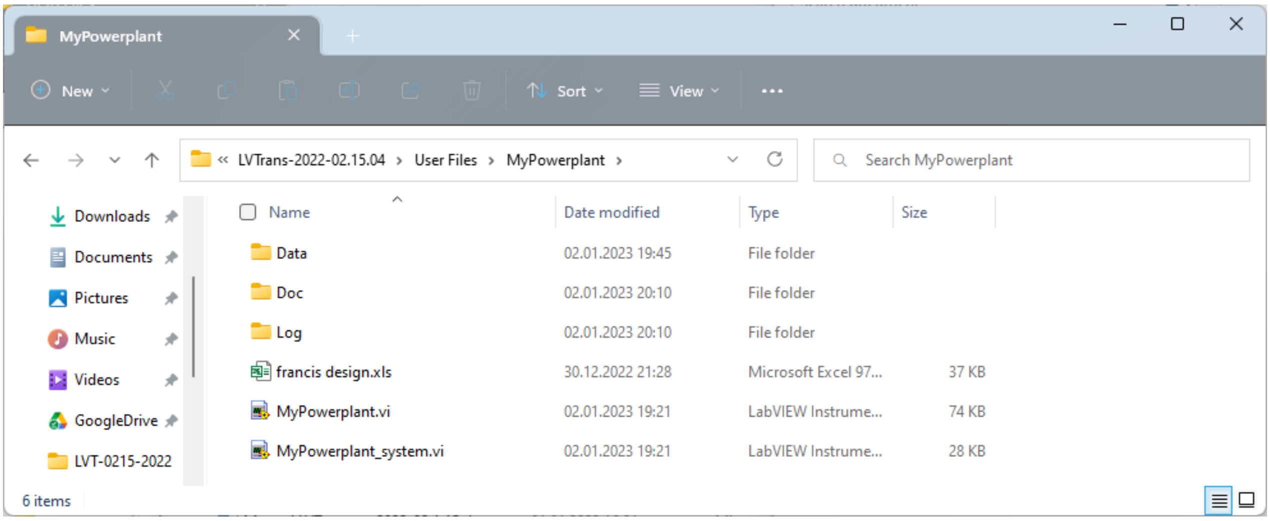


Figure 5.4 Log and Doc folder

Whenever you press the Log button, a Log folder is created (if it does not already exist), and the log file is created there. The Doc folder is a folder for a "snapshot" of the system, a picture of the system, and a text file with all the data files for the elements collected.

6 AN EXAMPLE

The fastest and most efficient way to learn LVTrans is simply to start using it. One cannot do anything wrong. Just remember to take backups and use the save function regularly. The «save all» function is also useful, since this will save and compile all open and referenced vi's. This chapter is about how to create a power plant model for simulation and analysis.

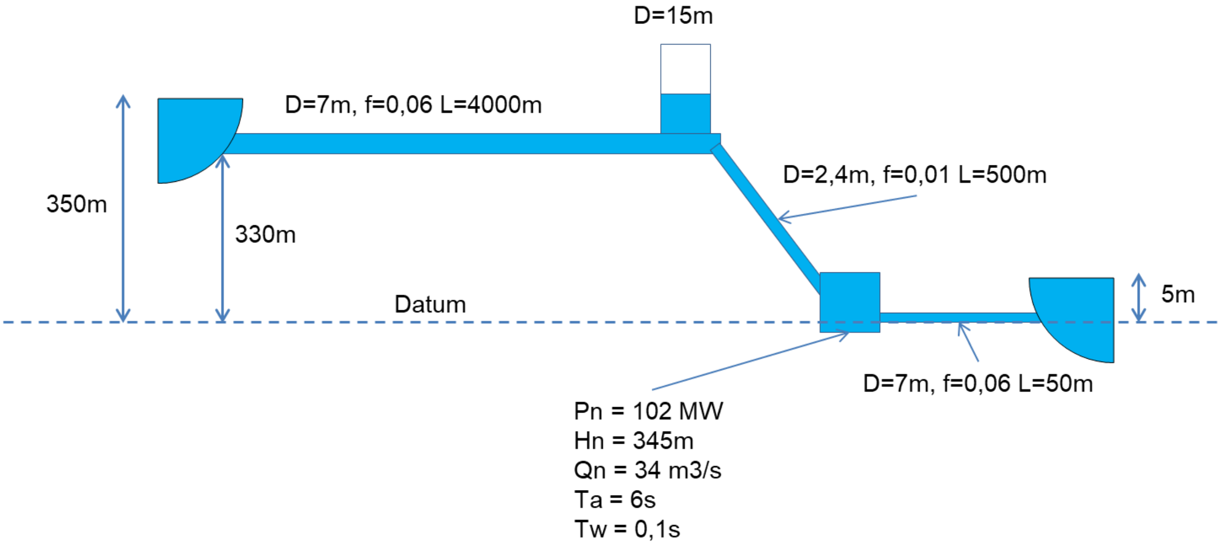


Figure 6.1 Example power plant to be built

6.1 OBTAIN THE OVERALL VIEW AND DATA

When working with LVTrans (or any other similar software to be honest), it soon becomes very apparent that the most time-consuming task, and often the more difficult part, is to get correct data and an overview of the water ways in the plant. A suggested process is therefore to simplify as much as possible, and then detail the model as necessary later. A simplified model containing the 5-15 most important components will typically become at least 90% correct, and often this is enough, especially for control analysis (analysis of turbine governing)[3]. A rule of thumb is to try to pick the 10% of elements that constitutes 90% of the plant. This is typically reservoirs, turbines, surge shafts, and tunnels/pipes/penstocks that connects these elements. Further, a principal sketch made by "pencil and paper" will be of great help, like for instance Figure 6.1.

Still, there is literally no limits for how large and complicated one can make the model in LVTrans. One can include every little detail by fine graining. For each little detail that is added, additional data for that detail must also be found. Make sure the data is correct though, or the addition makes little sense. The rule garbage in = garbage out is always valid. A simple but roughly correct model is usually more useful in practical analysis than a complicated and overly detailed model with questionable input data. To find a good balance takes practice and experience and will be dependent on what the model is used for.

[3] In addition, all elements such as surge shafts, pipes and Ts have damping constants that can be tuned. The damping in these elements together is at least 95% of all the damping in the system, making this simplified model much more accurate in fact than 90%. It is therefore no real reason to detail more than necessary unless there are some details that are of importance.

6.2 CREATING THE FOLDER/FILE STRUCTURE AND A TEMPLATE FOR A NEW SYSTEM

The file structure and a template for a new system is made by a vi called ”New or update powerplant.vi”. This vi is found in the main folder.

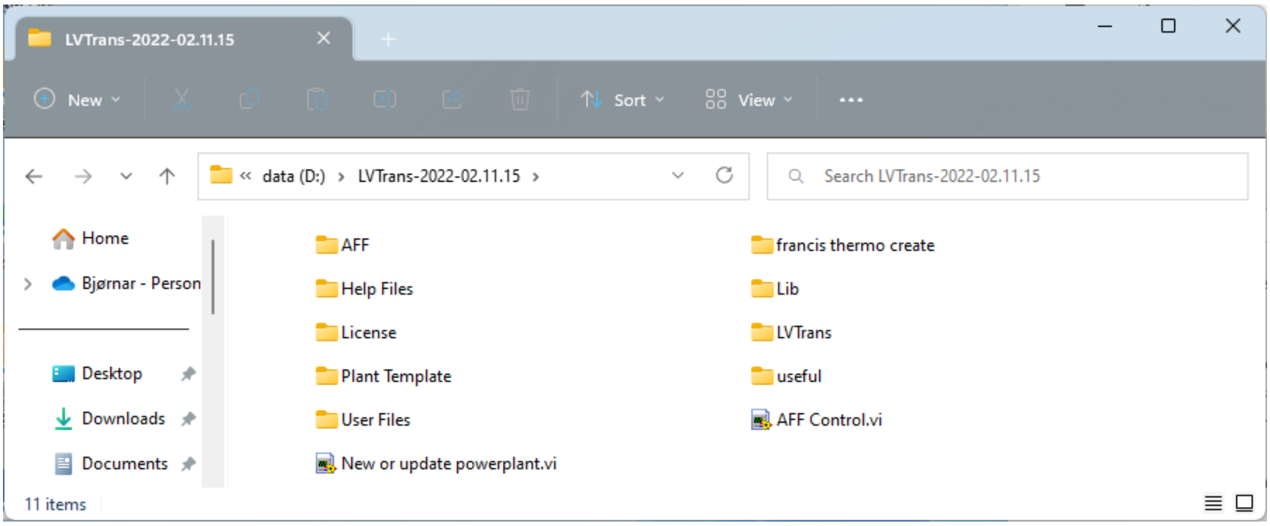


Figure 6.2 The main folder with the vi named “New or update powerplant”

Double click on the ” New or update powerplant.vi” and a new window is opened as shown in Figure 6.3[4][5]. Push the button names «Create new powerplant» and a new window opens where one can write the name of this new system. Here it is called ”MyPowerplant”.

[4] LabVIEW must be installed for this to work. A file with the extension vi is a LabVIEW file. vi stands for virtual instrument.
[5] This will also open LabVIEW. This vi is configured as “run when open” and LabVIEW will ask if this is really what you want. Just answer yes/run if that happens.

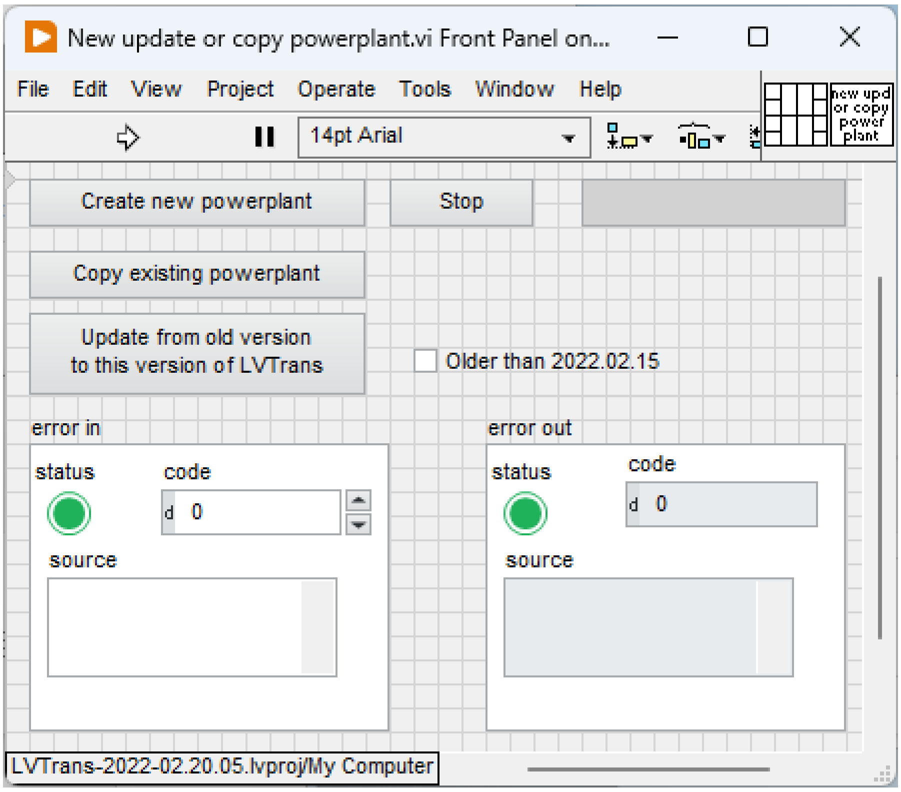


Figure 6.3 ”New update or copy powerplant.vi”

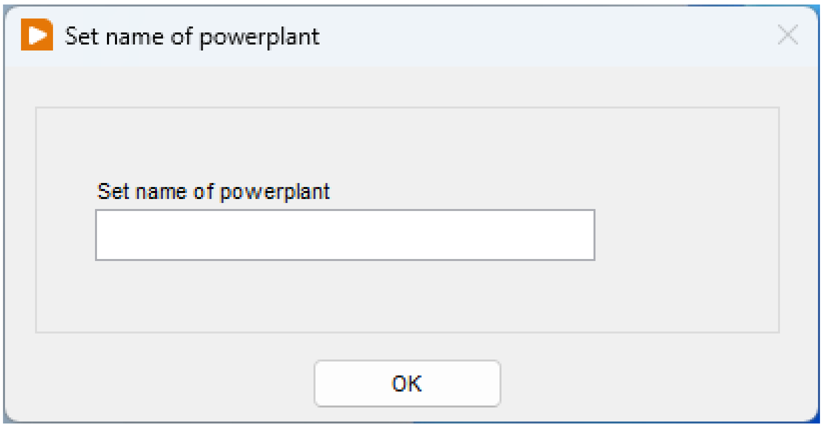


Figure 6.4 Window for writing in the name of the new powerplant.

Write the name (MyPowerplant), then push the button «OK». When that is done, one can just close the New or update powerplant.vi by pushing the X in the upper right corner (ordinary Windows feature).

What has happened is a new folder in the folder User Files is created with the name «MyPowerplant» in this case. This is shown in Figure 6.5

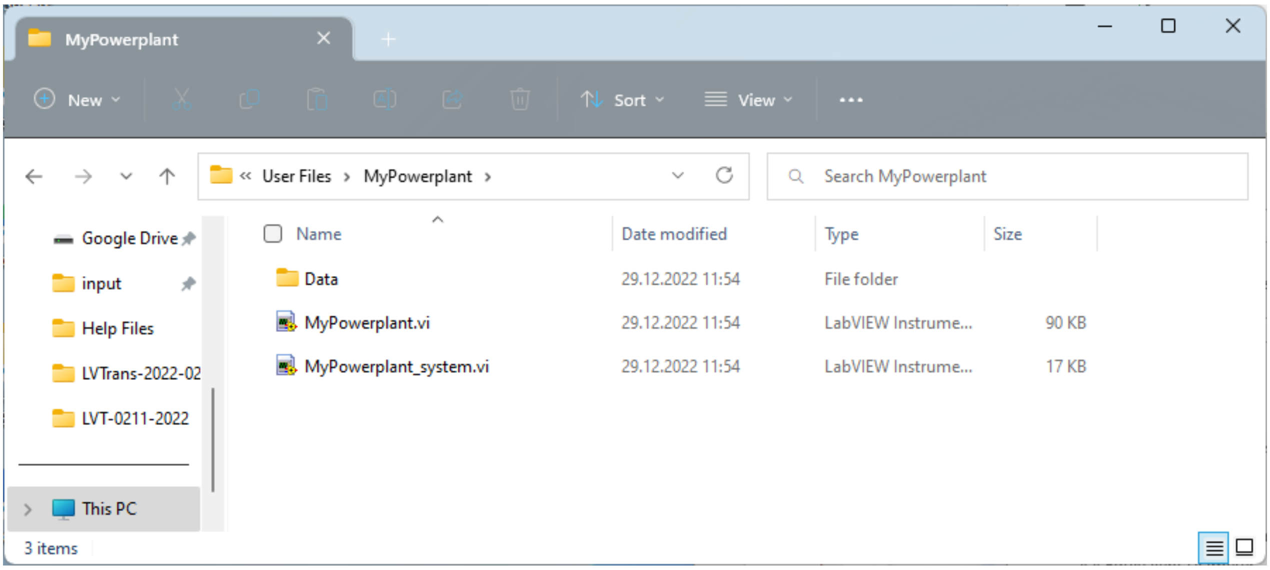


Figure 6.5 Folder and a template for ”MyPowerplant” automatically created in the folder ”User Files”.

The new folder will be equal for all new powerplants, except the name of the folder and the name of the files within.

One can also copy an entire powerplant including the datafiles. This is done by instead pushing the button "Copy existing powerplant". A question pops up asking for "Select main file". This is a reference to for instance "MyPowerplant" in Figure 6.5 or any other existing powerplant one has already created. When this is done, a new question pops up "Select system file". The "MyPowerplant_system.vi" is chosen, that is the "system"-file. To copy an existing plant can be very useful when one already have a good model, but wants to make it more detailed, or change a few parts on it.

6.3 INCLUDE ELEMENTS

The next thing to do is to put in elements as shown in for instance Figure 6.1. At this stage it will help enormously with a good sketch up front. When putting in the elements, the plant is built piece by piece, very much like Lego. Also, like Lego, there is no limits, it’s all up to the person in front of the screen.

Double click on ”MyPowerplant_system.vi” (the xxx_system.vi file). Then the front panel of that vi is opened. The front panel does not need any modifications, so we go straight to the block diagram of that vi. This is done with the menu Window->Show Block Diagram, or the keyboard shortcut Ctrl_E[6].

[6] Changing from the front panel to the block diagram is done with Ctrl_E, a LabVIEW shortcut.

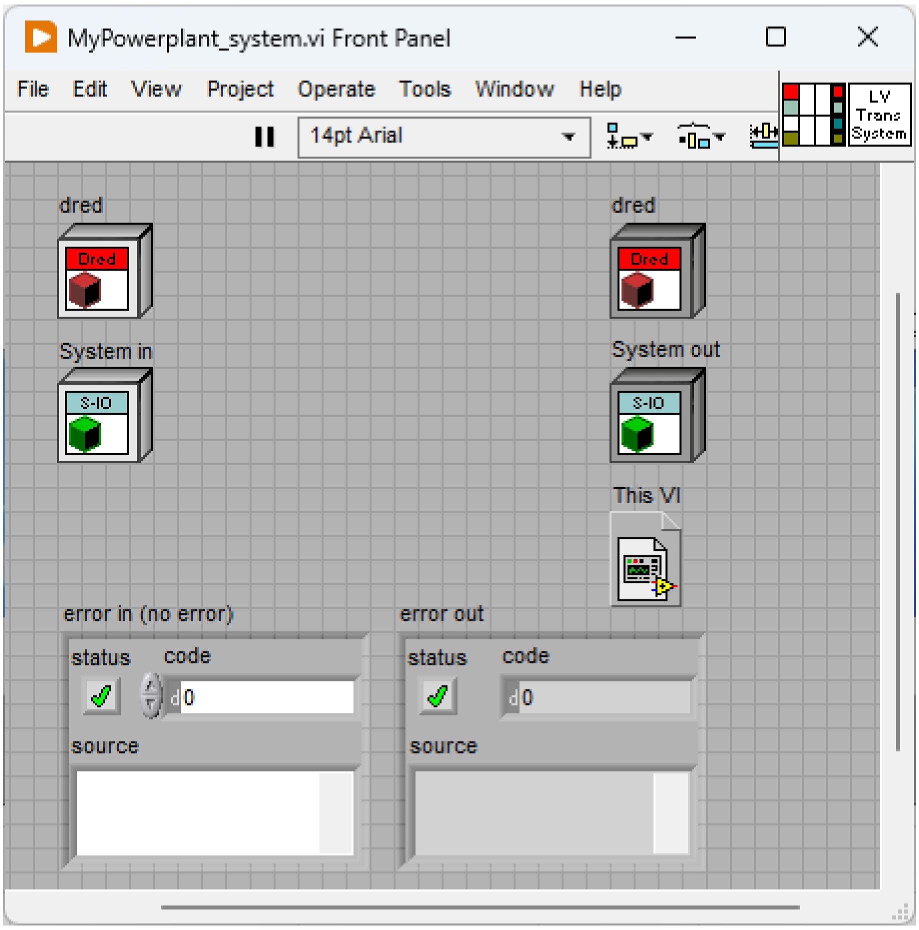


Figure 6.6 The Front Panel of XXX_system.vi

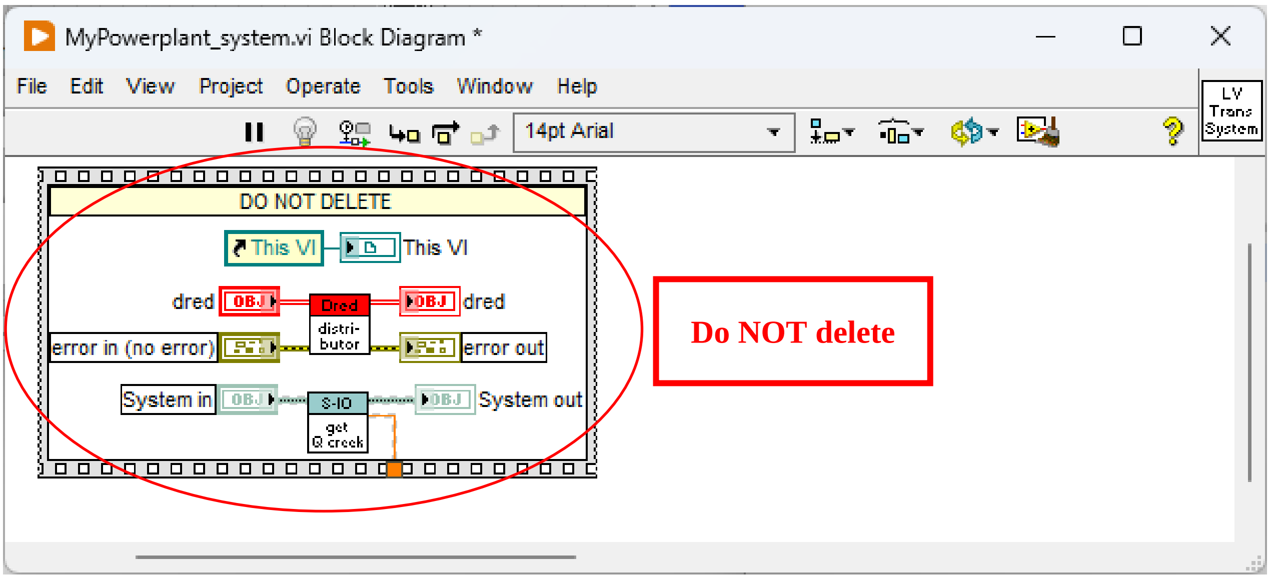


Figure 6.7 The Block Diagram (empty) of XXX_system.vi

A fresh block diagram of xxx_system.vi has some objects already there. These objects **must be there at all times** to assure the program runs. They must not be deleted! They can however be moved around to get them out of the way.

When the block diagram is open, one can start inserting elements from the menu. This is done by picking the elements directly from the ”LVTrans” menu in LabVIEW (see page 11 on how to set up the menu). If the functional menu isn’t already opened, it can be opened by right clicking anywhere on the block diagram. Elements are chosen with LVTrans->[the actual chosen element].

From the sketch, Figure 6.1, one can see that 3 pipe elements are needed. They can be picked one by one from the menu, or just pick one and copy that using ordinary Windows mouse clicks; (Ctrl + left click or Ctrl+C and Ctrl+V). Every element has full «drag and drop» functionality. When this is done, the block diagram looks something like Figure 6.8.

In addition, one surge shaft must be placed, 1 turbine, 1 constant level left and one constant level right and 1 PID governor. The block diagram becomes as shown in Figure 6.9.

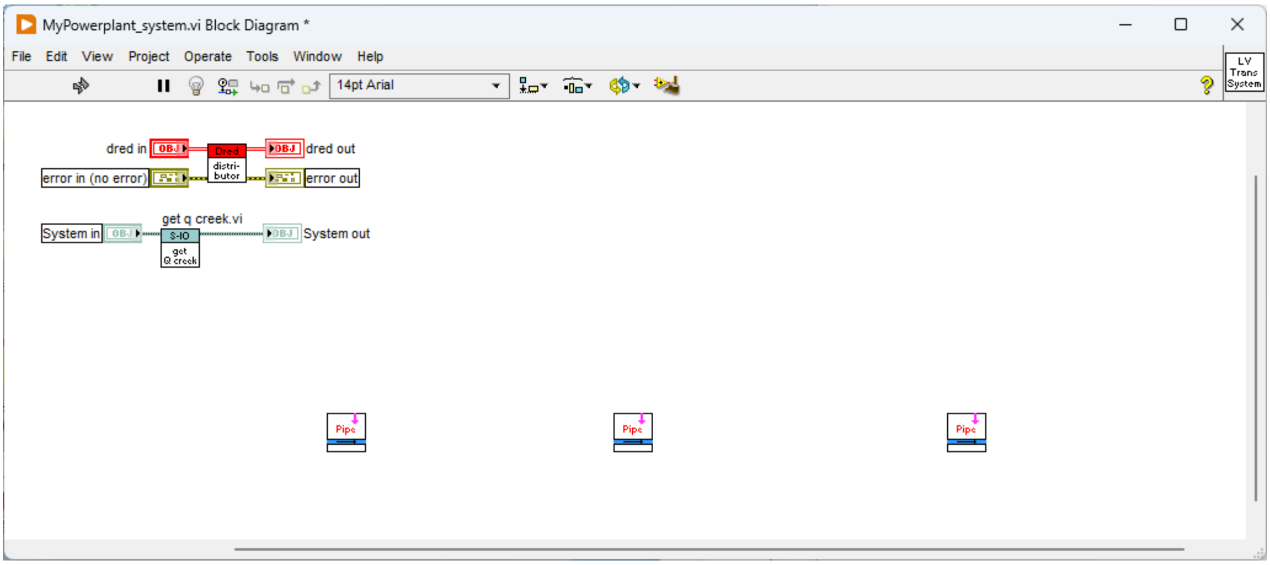


Figure 6.8 All 3 pipe elements into the block diagram

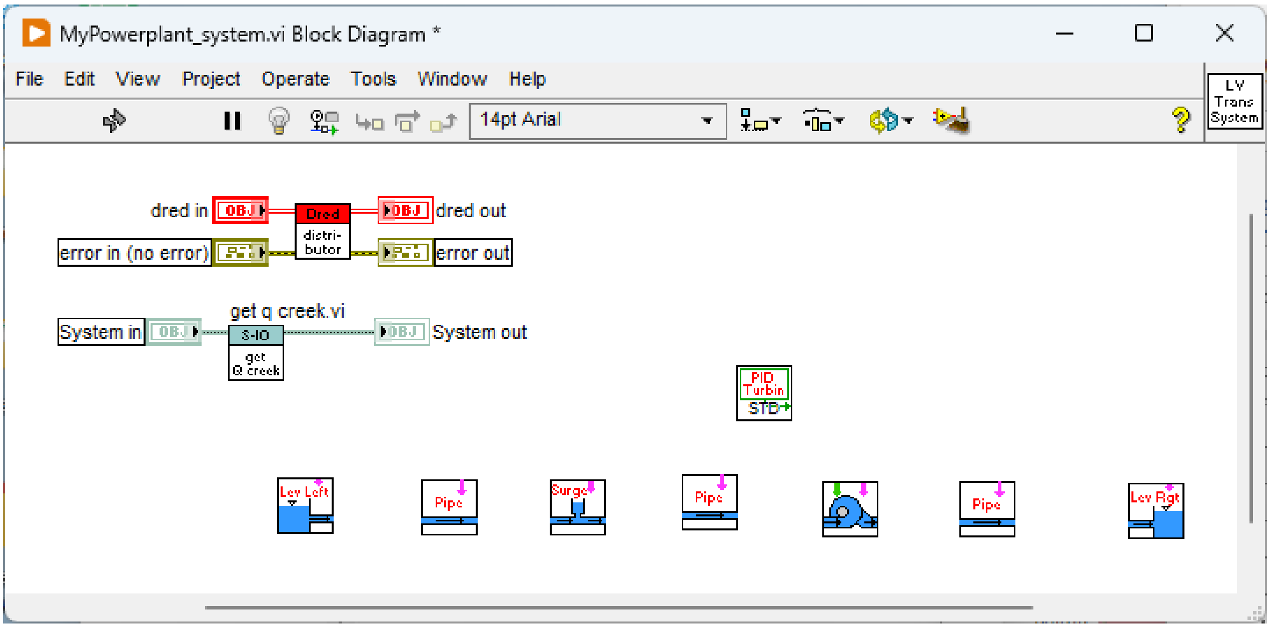


Figure 6.9 All elements in place

6.4 PUT DOWN SOME WIRE

When all the elements are roughly positioned, one has to draw wires between them. The rules are as follows:

- A Pipe element must have a non-pipe element on each side[7].
- Non-pipe elements connect only to pipe elements (with a few exceptions)
- PID elements can only be connected to turbine/pump elements.
- Follow the direction of the arrows

It's not possible to connect pipe to pipe or non-pipe to non-pipe. This will result in so called broken wires. When there are no broken wires, the connections should be OK.

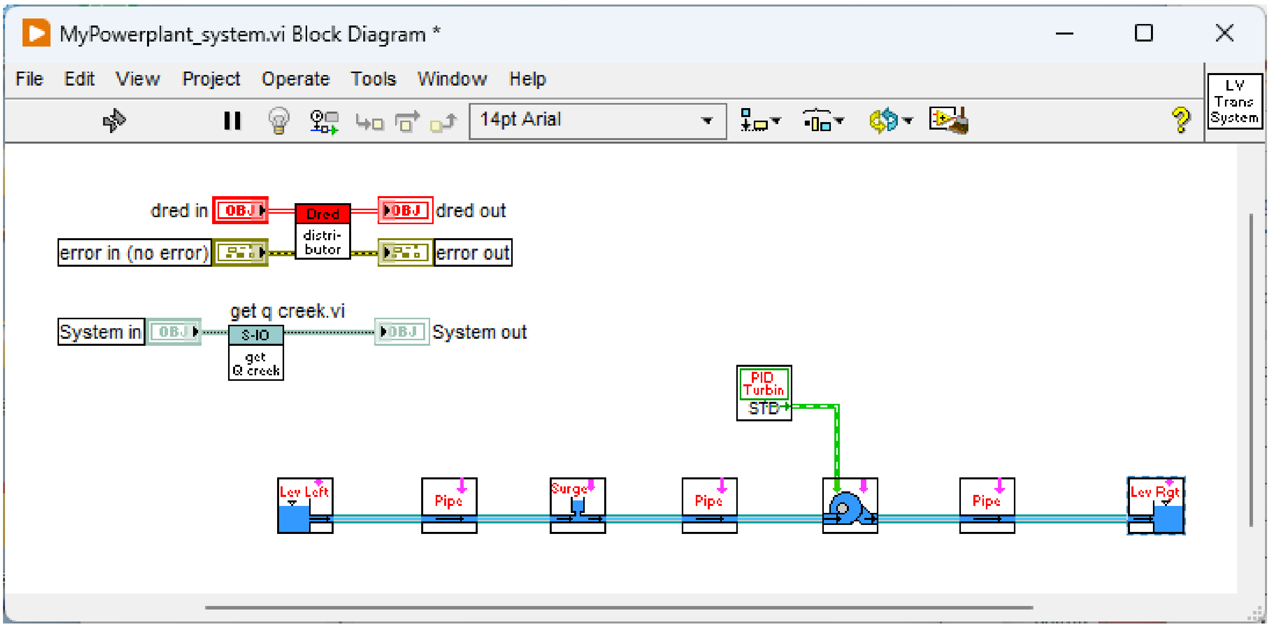


Figure 6.10 All the elements connected

Figure 6.10 shows all elements connected. One can leave this "as is", or move the elements around a bit to get a better visual impression of the topology. The elements can be moved at will, as long as the topology of calculations is the same. That is, the connection between elements must represent the actual plant one is modelling.

[7] This is done on purpose and is the way the Method of Characteristics will work in an object oriented setting.

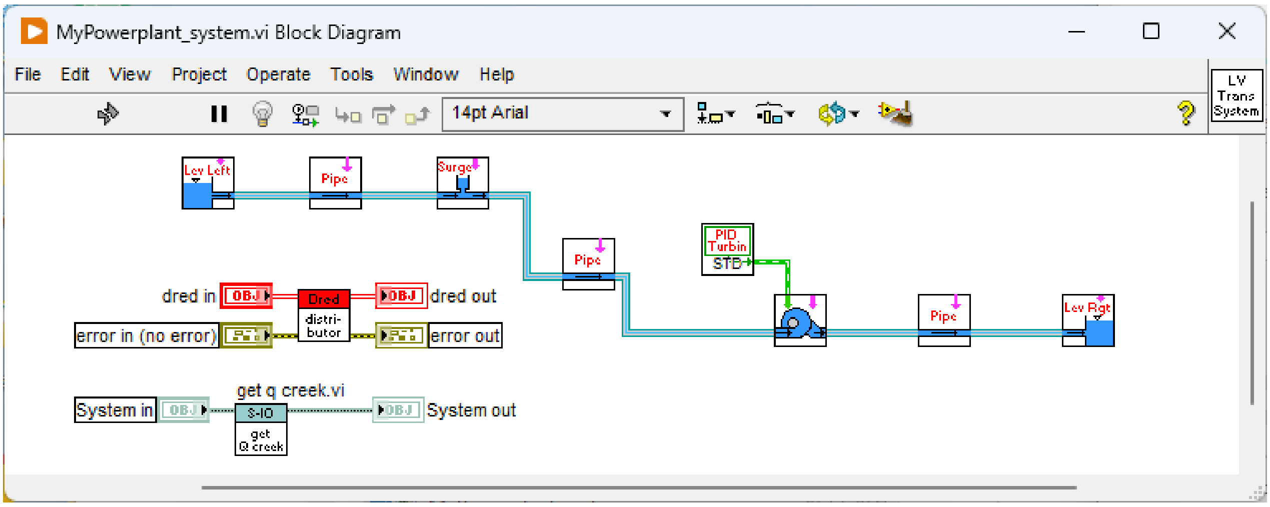


Figure 6.11 All the elements connected and moved slightly

The next thing is to give each element it's unique name. Every element must have a unique name, or LVTrans refuses to run. Finding appropriate names is not as easy as it seems. It's a good thing to make them descriptive, but not to go overboard with that, since the majority of the pipes and tunnels in a powerplant isn't named in the usual sense of the word, and particularly not when dividing a penstock into several pieces for instance. Making them short and descriptive is good (often just naming them pipe01, pipe02 etc works just fine). Any international character that Windows recognize can in theory be used. However, this does not always work in a Linux environment, and experience has shown that Windows installations are not always set up correct. The problem with this is when copying a plant from one machine to another. Using only ASCII characters is therefore a fail safe «trick». Line shift in names cannot be used because the names are also used as filenames for the data.

NB! In LabVIEW the main enter key on the keyboard will automatically create a line shift instead of acting as a "enter key". Ctrl+Enter will "enter" the text, or one can use the enter key on the keypad.

From the point of view of LabVIEW, the name is just a string constant, and any method to create/copy a string constant can be used. The most intuitive way is perhaps to right click on the pink terminal and chose "create -> constant". Then simply write the name in the little window.

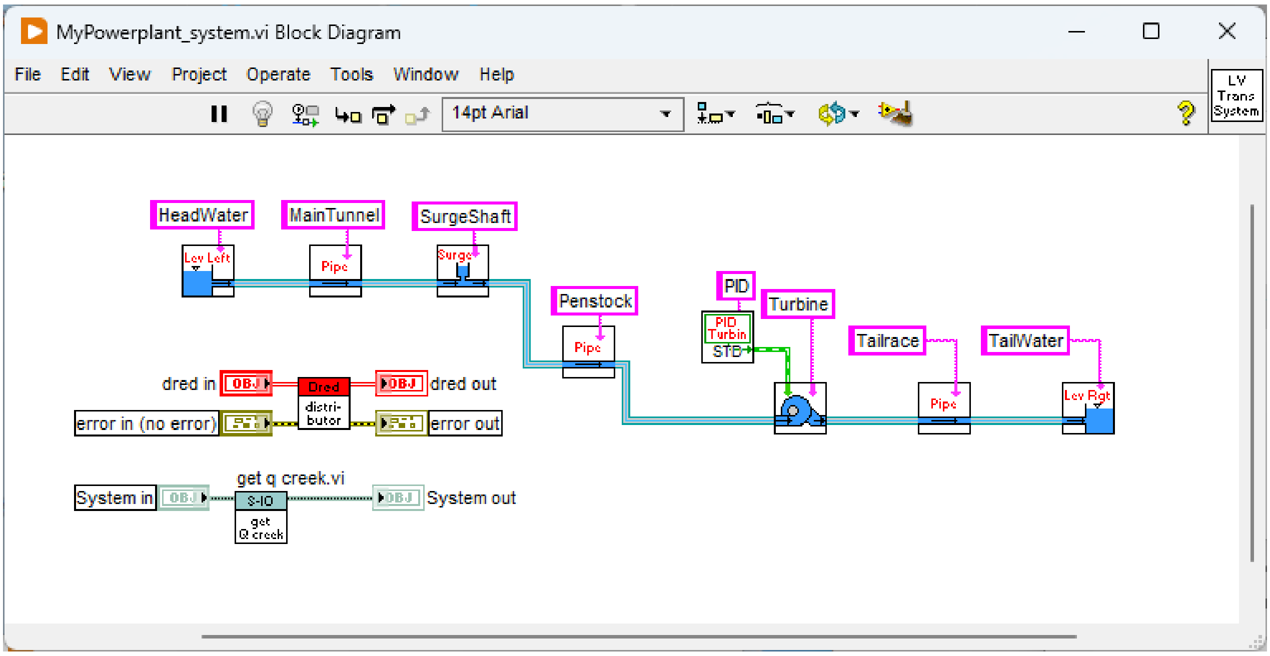


Figure 6.12 The entire system is set up. Notice the broken arrow in the upper left corner from the other figures is gone.

When every element has a name, and they all are connected, the broken arrow in the upper left corner disappears. This means the system is ready to be run, there are no errors in the topological setup of the plant (programmatically speaking).

The block diagram has two purposes.
1. The system is set up so that LVTrans can use it for calculations.
2. Set up the system in such a manner that it will be easy to come back to it in a couple of years' time or more, and making it easy for other people to use.

NB! Remember to save the system regularly. A star close to the file name indicates changes has been done that are not saved. Save by «File_Save» or Ctrl+S or "save all".

When the proper preparations are done up front, setting up even the most complicated hydro power plants takes no longer than 30-60 minutes. Usually, it takes 5-15 minutes. Lots of plants are more or less equal topologically. Making templates and copy them for new plants is an effective approach. Then it's only the data files that need to be changed.

What we have done so far is to:
- Set up the topology of the system, which elements are placed where and how they are connected.
- Given each element a name.

The next thing to do is to define the details in each element. For instance, a pipe/tunnel must have data describing the length, diameter, roughness and so on.

6.5 INSERTING DATA FOR EACH ELEMENT

Let the xxx_system.vi stay open with the block diagram visible. If it's closed, then open it and show the block diagram as in Figure 6.12. Then the other file in Figure 6.5 must be opened by double clicking it. In this example it is”MyPowerplant.vi”. There is no need to open the block diagram for that file.

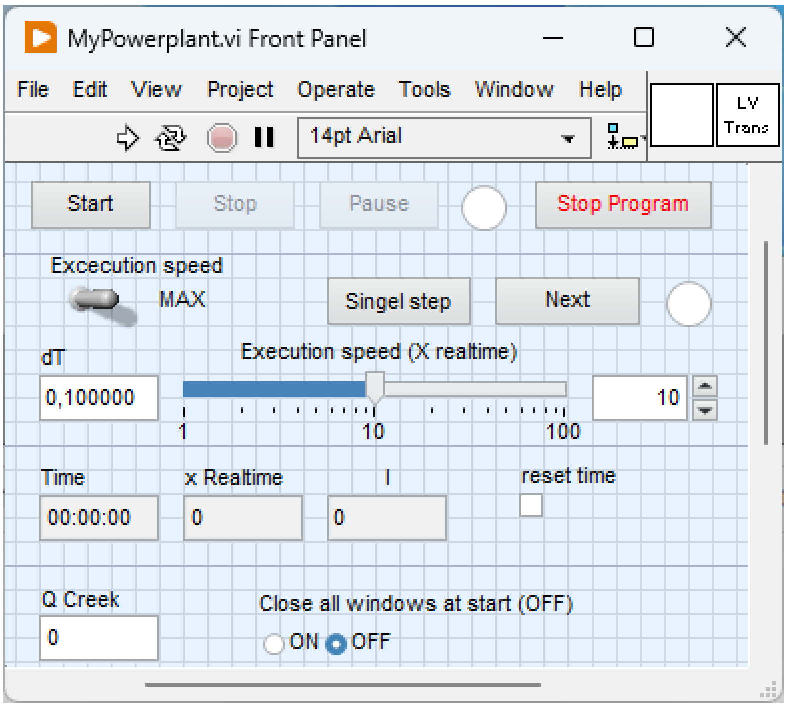


Figure 6.13 The main program, ”MyPowerplant”

Figure 6.13 shows the main program, or main vi. This is mainly used for starting, stopping, pausing and a few other things. To start LVTrans, push the white button. This starts the program, but not the calculations. The other fields in the main program are:

- Start-button: starts the calculations
- Stop-button: stops the calculations
- Pause-button: pauses the calculations. This is very useful in many circumstances. For instance, one can push pause, then configure several other elements to do certain task, then push pause again and it will simulate based on the configuration. To simulate emergency shut down by several turbines for instance. One is then assured that all the changes happen at the exact same time.
- dT: This is where the time step is set, the time step is global for all elements. 0,1 s is the default and is OK for a first approach. Typically, this is set to 0.01-0.03 for final calculations in a report. Smaller dT causes the simulations to go slower. dT cannot be changed while the program runs. dT must be set before pushing start.
- Execution speed: Changes between max execution speed and using the slider. LVTrans calculates much faster than real time. If one wants to follow the transients, it is therefore a good idea set execution speed to slider and run at run for instance at 10x real-time. Otherwise the execution is normally so fast there is no chance to follow the calculations.
- Stop Program: This stops the program so that the white arrow is visible again.
- Time: This shows the «true» time in the simulations, that is the physical time. Since the calculations go much faster than real time, so will this time.
- x Real time: Shows how much faster the calculations compared with real time.
- I: shows the number of iterations done, or number of time steps.
- Q Creek: A place (among other) to set the flow into creek shafts.

- Reset time: Resets the time so its starts at zero. Often used together with “Pause” to start at a defines point zero in time.
- Next: time select can be set in single step. This button will then move one single time step for each push.

Before the actual simulations starts, the data files must be set up and filled with data for each element. The data is stored in files, but since the program has never run before (considering a new powerplant has been built), these files do not yet exist. The files are ordinary text files. They can be edited in each element themselves or edited «by hand» in an ordinary text editor. They are made as Windows ini files.

When the main program (main vi) is open, push the white arrow. Leave the block diagram of the xxx_system.vi open. The main program will get the “runtime look”.

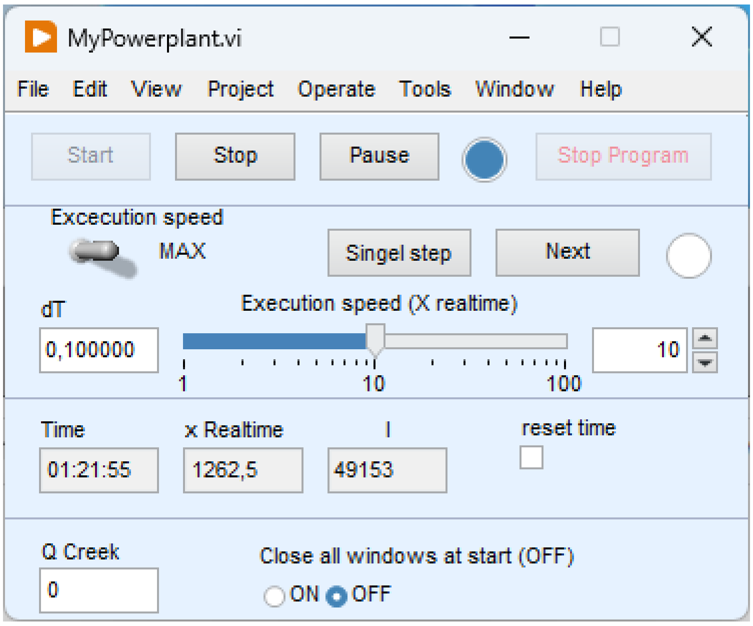


Figure 6.14 LVTrans has started

When analyzing with LVTrans it’s practical to have the main vi in some corner, and the block diagram, Figure 6.12, of the system vi opened another free space. LVTrans, as LabVIEW itself, benefits with lots of screen estate, due to the graphical nature.

A small trick for this first run, is to set Execution speed to real time so it don’t need to calculate like mad on the default data, which usually are way off. In older days, this would also increase the response of the PC, but that is no longer the case with newer PCs and newer operating systems.

Set Execution speed to real time and push the Start button. Lots of windows will now pop up saying that new and default data files have been made for each element. Just push OK on these until no more pops up. LVTrans has now made all the necessary data files, one for each element. The names of these files are the same as the names we have chosen in the system block diagram, see Figure 6.12.

The next thins is to enter actual data for each element. While the program is running, look at the system vi, XXX_system.vi, see Figure 6.12. Double click on the HeadWater element. A window representing the element will then pop up as shown in Figure 6.15.

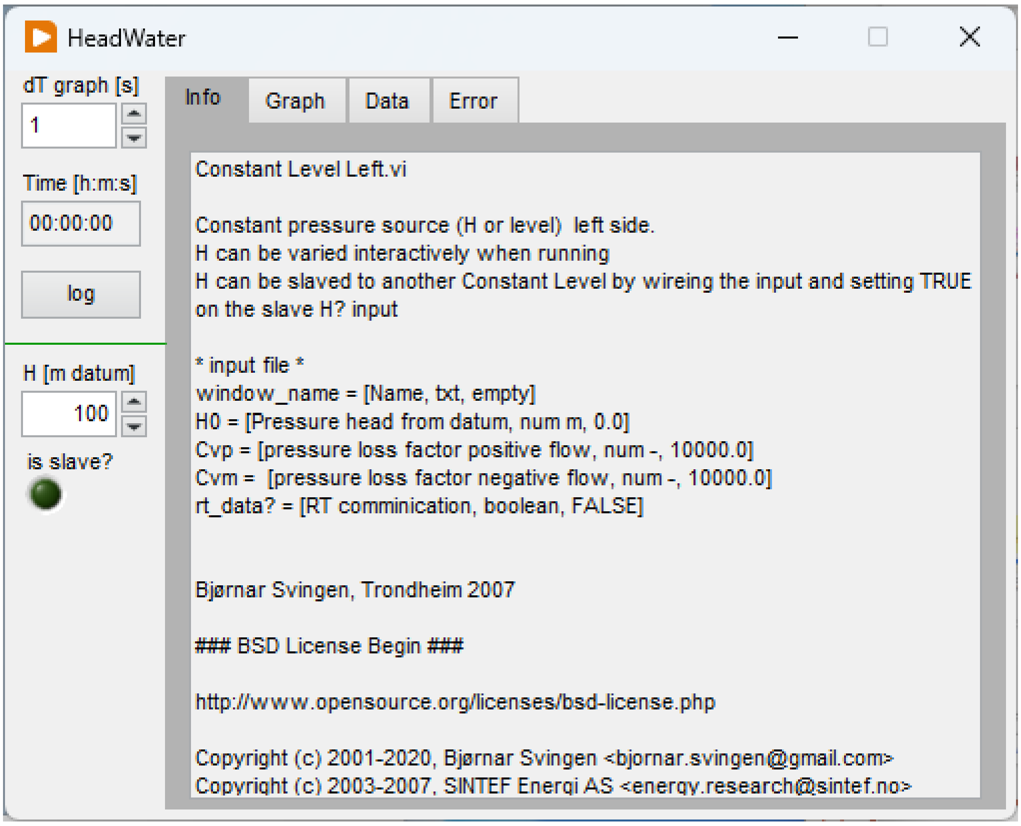


Figure 6.15 HeadWater element opened for the first time.

There are some buttons etc that controls the graph, logging of data and input while running. This time we will only insert data. Push the Data tab.

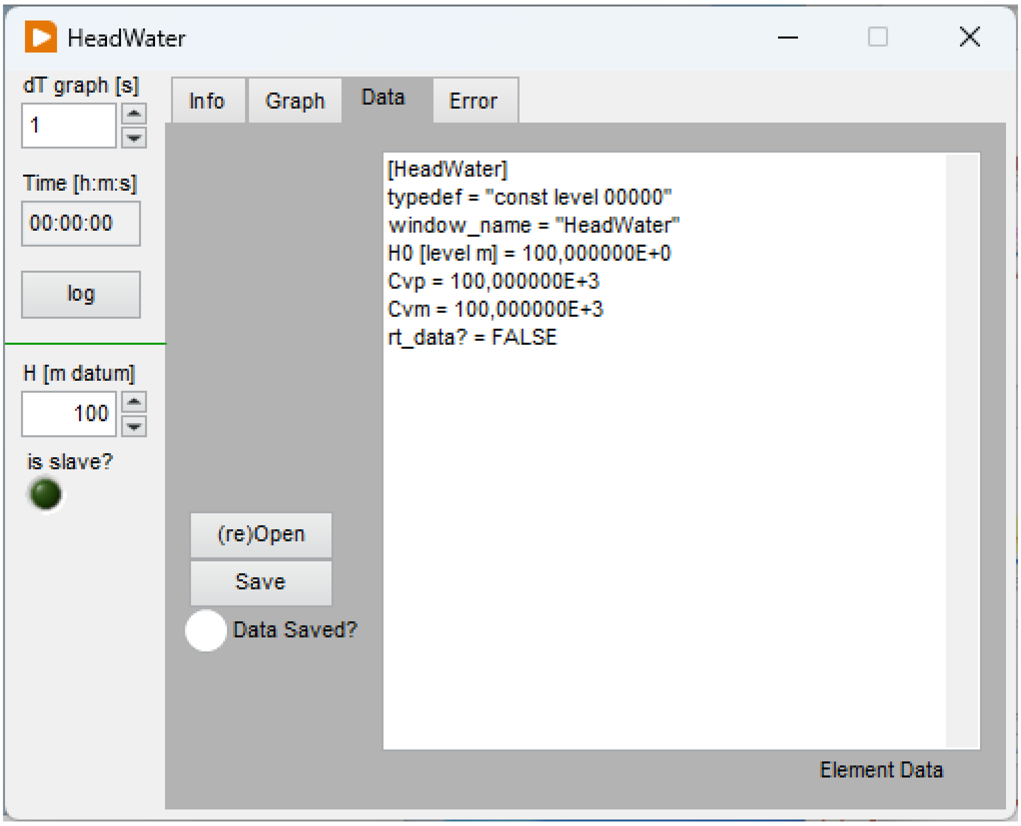


Figure 6.16 HeadWater element with default data

Again, the sketch of the plant will come in handy. Look at the sketch, Figure 6.1, and insert the data that fits with reality for the element.

NB! Do not change the two upper most lines in the files. The first line is needed for the Windows ini functionality, and the next line is a typedef string for the version of the file. Also, only change what is to the right of the equal sign. The Boolean T/F are written as TRUE, FALSE

Edit the data in the window to make it fit with Figure 6.17. The only important data for now is H0 (head water level). Cvp and Cvm are loss coefficients for the water when going in positive and negative direction respectively. We don't need to bother with those now, but to get as correct results as possible, these are also important to edit.

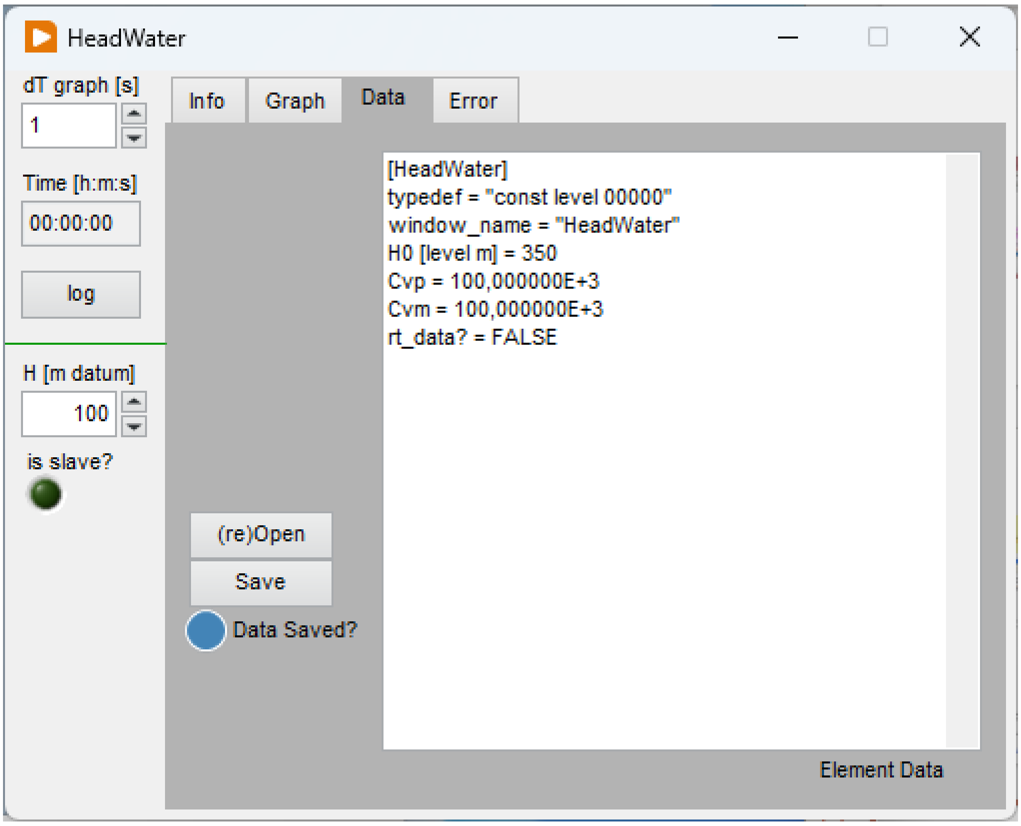


Figure 6.17 Edited values of HeadWater

When the window looks roughly as in Figure 6.17, push the Save button. Close the element with the red X (ordinary Windows close).

The same procedure is used for all elements. Edit those so they are as shown in the following figures. On some of the elements one might possibly scroll down to see all the data.

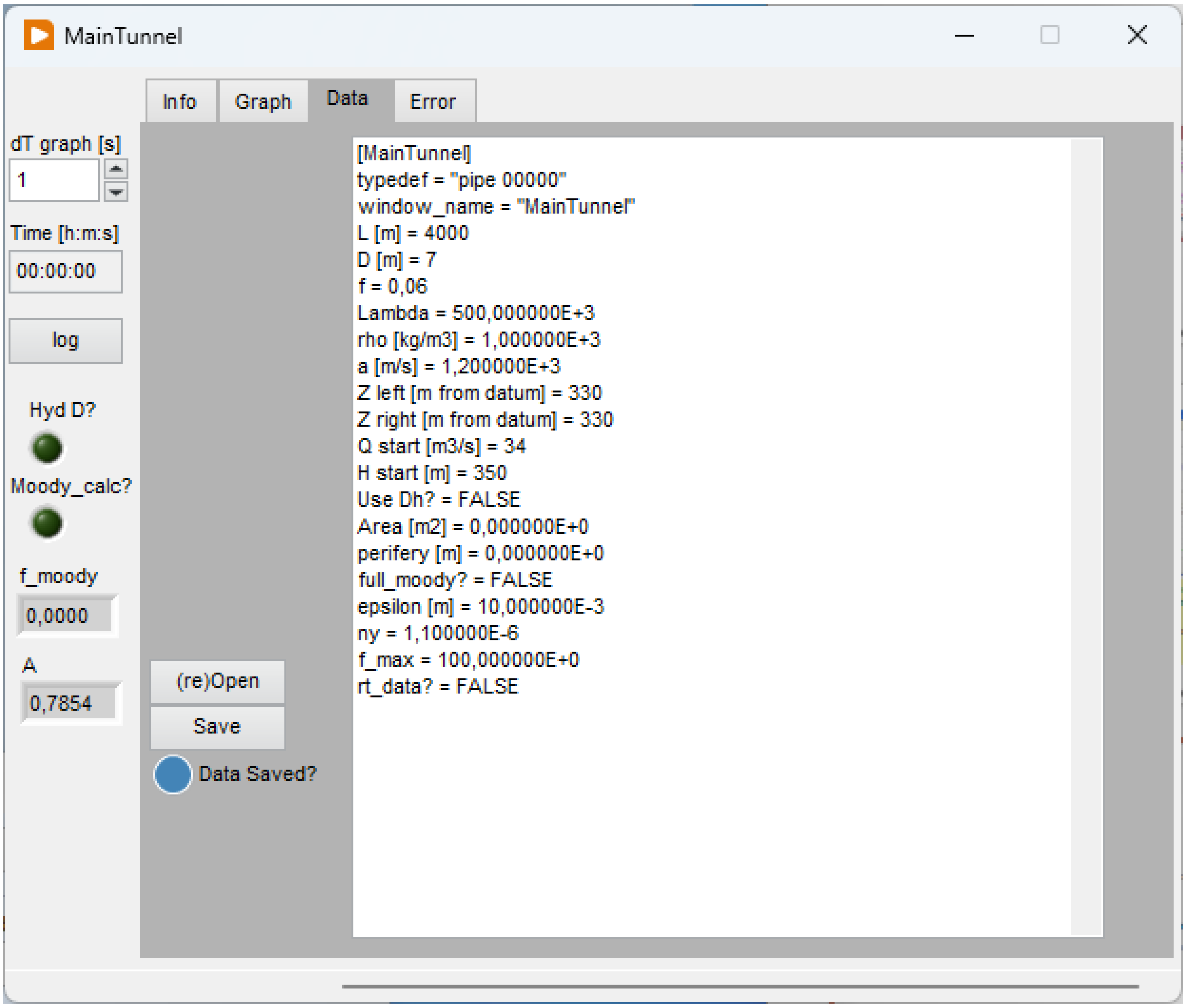


Figure 6.18 MainTunnel

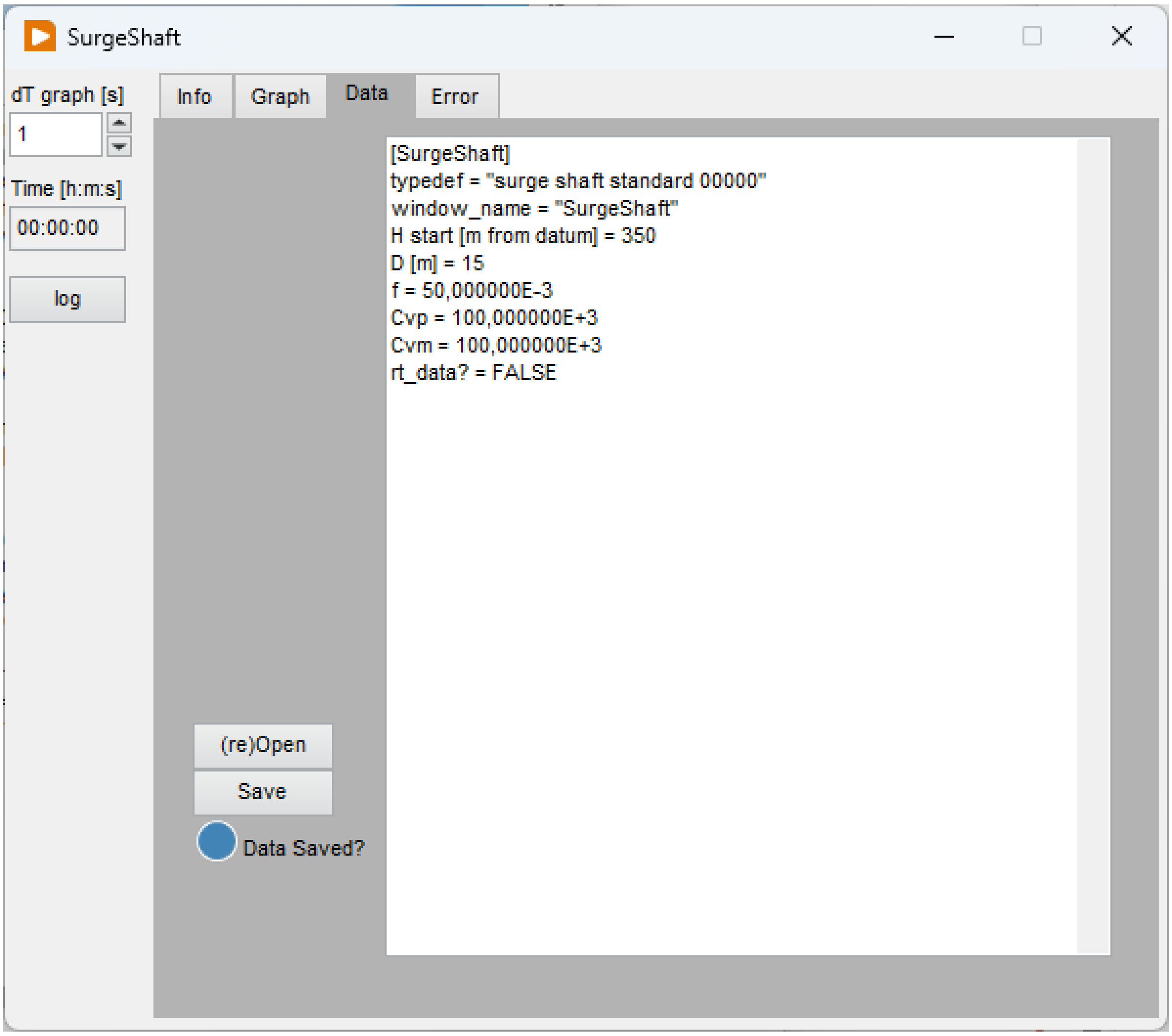


Figure 6.19 SurgeShaft

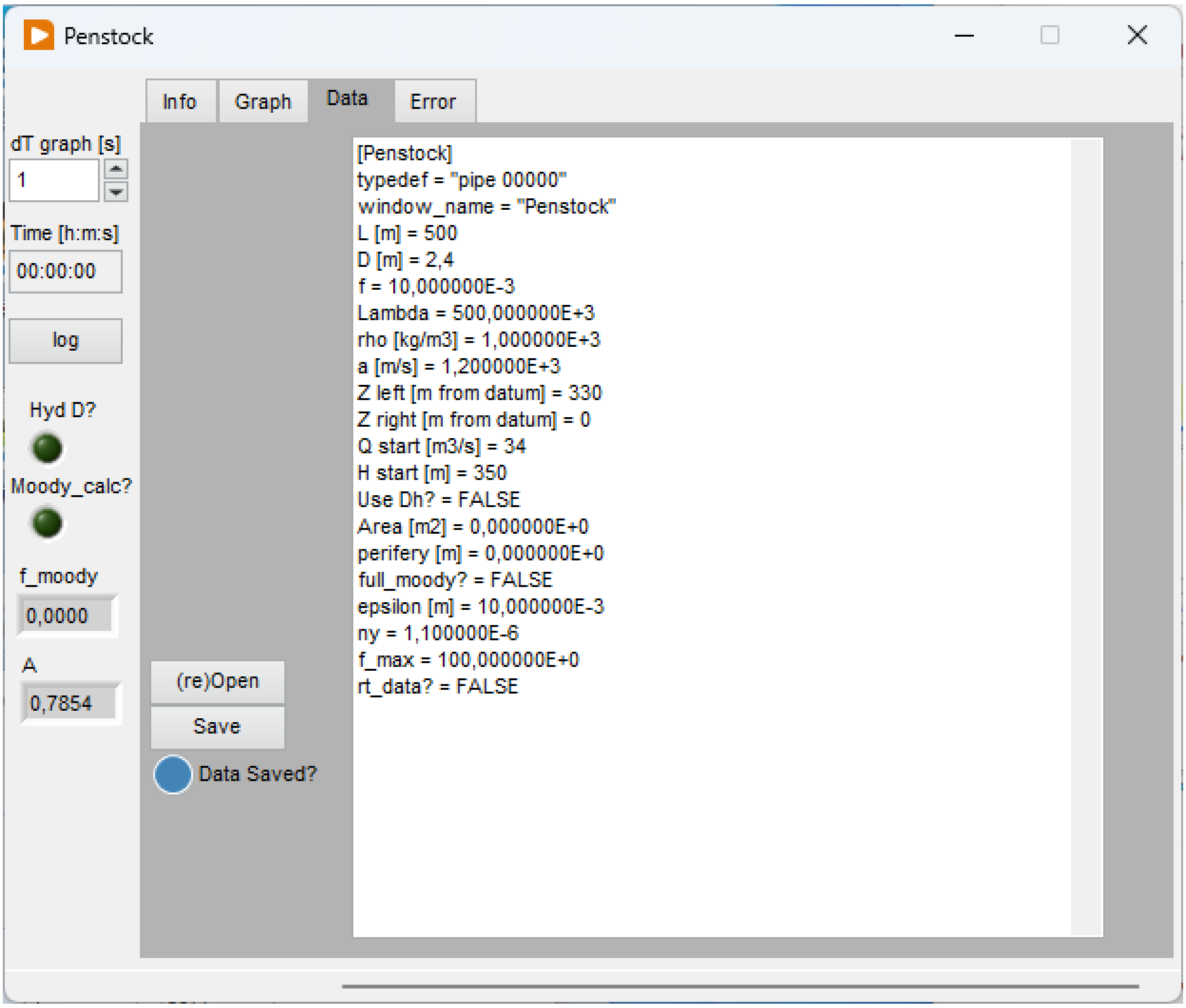


Figure 6.20 Penstock

With regards to turbine and governor, one does not usually know the exact data up front, if the plant is not yet built, or one is doing a study of how to lay out a new plant.

A small Excel snippet is made that can be used to find the main dimensions of a Francis turbine. This file can be found in the folder ”useful” in the main folder of LVTrans. The name of the file is ”francis design.xls”. This will find the main dimensions after a method by Prof Hermod Brekke. Open the file, and follow the instructions there. The result will be the main dimensions of a turbine that can be inserted into the turbine data file, see Figure 6.21. This sheet can also be used for Pelton turbines as shown later, but not on Kaplan turbines[8][9].

Both Francis and Pelton turbines are calculated «dynamically» in LVTrans. What this means is that the transient turbine equations by Prof. Torbjørn Nielsen are used. These equations are based on Euler turbine equations. This is rather different from most other programs that use hill charts from laboratory tests. Using hill charts is described in Wylie and Streeter, which also is the method used for pumps in LVTrans. These hill charts are in principle only valid for steady state where they are more or less spot on, since they are measured at steady state. It is assumed they also are valid during the transient phase, but no one really knows to which degree this is true.

[8] One can also use a special commercial software called Alab. This software is however no longer being developed.
[9] At the moment LVTrans can only be used for Francis and Pelton.

What can be said is that accuracy is likely to increase when the transients get slower. The accuracy is also likely to increase when the step (in the hill chart) from one steady state to another steady state gets smaller.

By using Nielsen/Euler equations together with the simple design method, steady state will not be 100% accurate because these equations can never be a 1 to 1 relation with the full 3D geometry of a turbine. At least not without explicitly tuning the model to a specific turbin/hill chart. What can be said however is that the model is 100% physically correct also in the transient phase because it is a first principles approach. In practical terms using hill charts is easier to document in a report, and easier to use *as long as one can get hold of a correct hill chart*. Both methods have their merits, and a hill-chart turbine model is also coming to LVTrans at some point.

Note! The values in the turbine element, and to some degree also the PID element is based on BEP (best efficiency point). BEP power is typically around 80-85% of nominal power. One can also use nominal values, just remember to be consistent. The error done will be insignificant, at least for initial calculations. Consistency is also true for the servo. An opening of 1.0 (pu) using BEP correspond roughly to an opening of 0.85(pu) using nominal values. This is important also for closing/opening times for the servo.

	A	B	C	D	E	F	G
1	1 Set nominal or BEP head, H0						
2	2 Set nominal or BEP flow, Q0						
3	3 Set the frequency (50 or 60 Hz)						
4	4 Adjust the number of pole pairs until beta2 [calculated] is between 13 and 21 degrees						
5	5 Set beta2 [shall] equal to beta2 [calculated]						
6							
7							
8	INPUT			Values to enter into LVTrans			
9	Grid frequency [Hz]	50		H BEP [m]	345		
10	eta	0,94		Q BEP [m3/s]	34		
11	H BEP or nom [m]	345		n [o/min]	428,57		
12	Q BEP or nom [m3/s]	34		Speed number	0,35		
13	Pole pairs	7		r1 [m]	1,339		r1/r2
14	beta_2 [shal]	18,8		r2 [m]	0,891		
15				a1 BEP	12,9		
16	beta_2 [calculated]	18,8		b1 BEP	64,3		
17				beta2	NA		
18				Pole pairs	7		
19				P BEP [MW]	108,28		
20				T BEP [kgm]	2412601		
21							
22	If nominal values are used as input, then the output will be nominal values						
23	If BEP values are used as input, then output will be BEP values						
24							
25							

Design | Beregning | Design ...

Figure 6.21 Excel sheet for main dimensions of a turbine

The values in red in Figure 6.21 are used in the data file for the turbine.

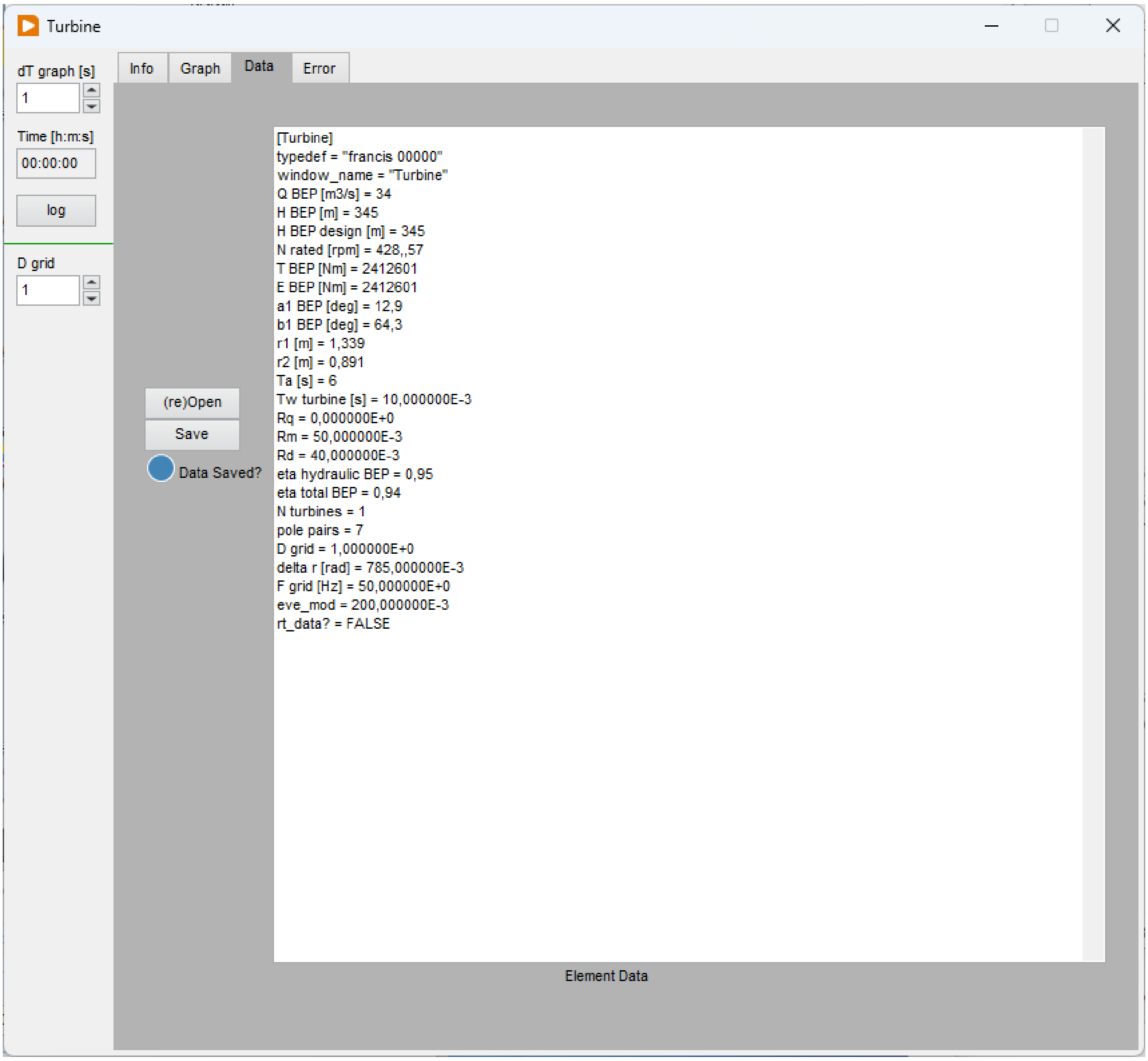


Figure 6.22 Data for the turbine element

Then the data fpr the PID is entered. That is, the ones we know for now. More data is specified later.

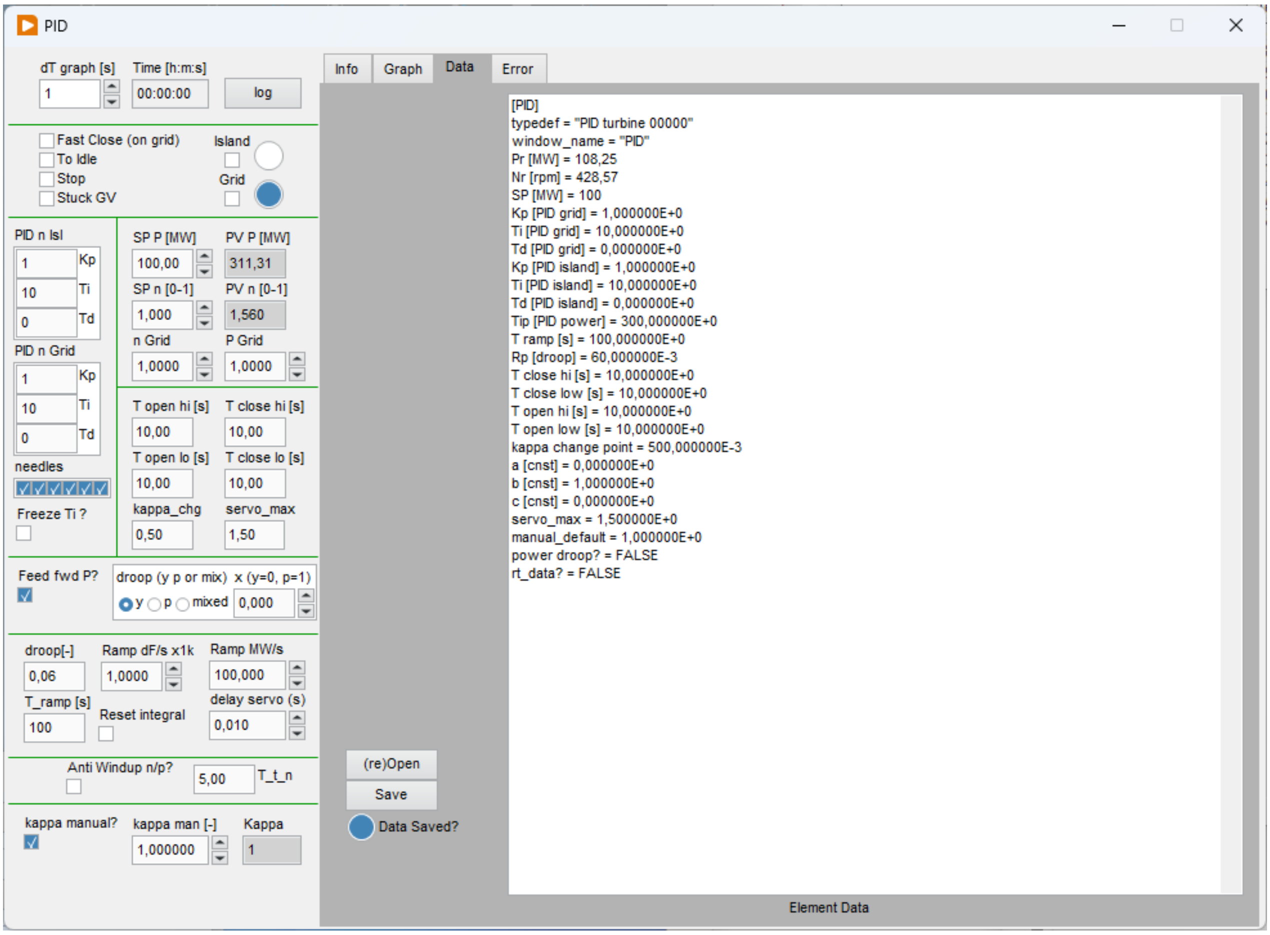


Figure 6.23 Data for the PID (those known at this stage)

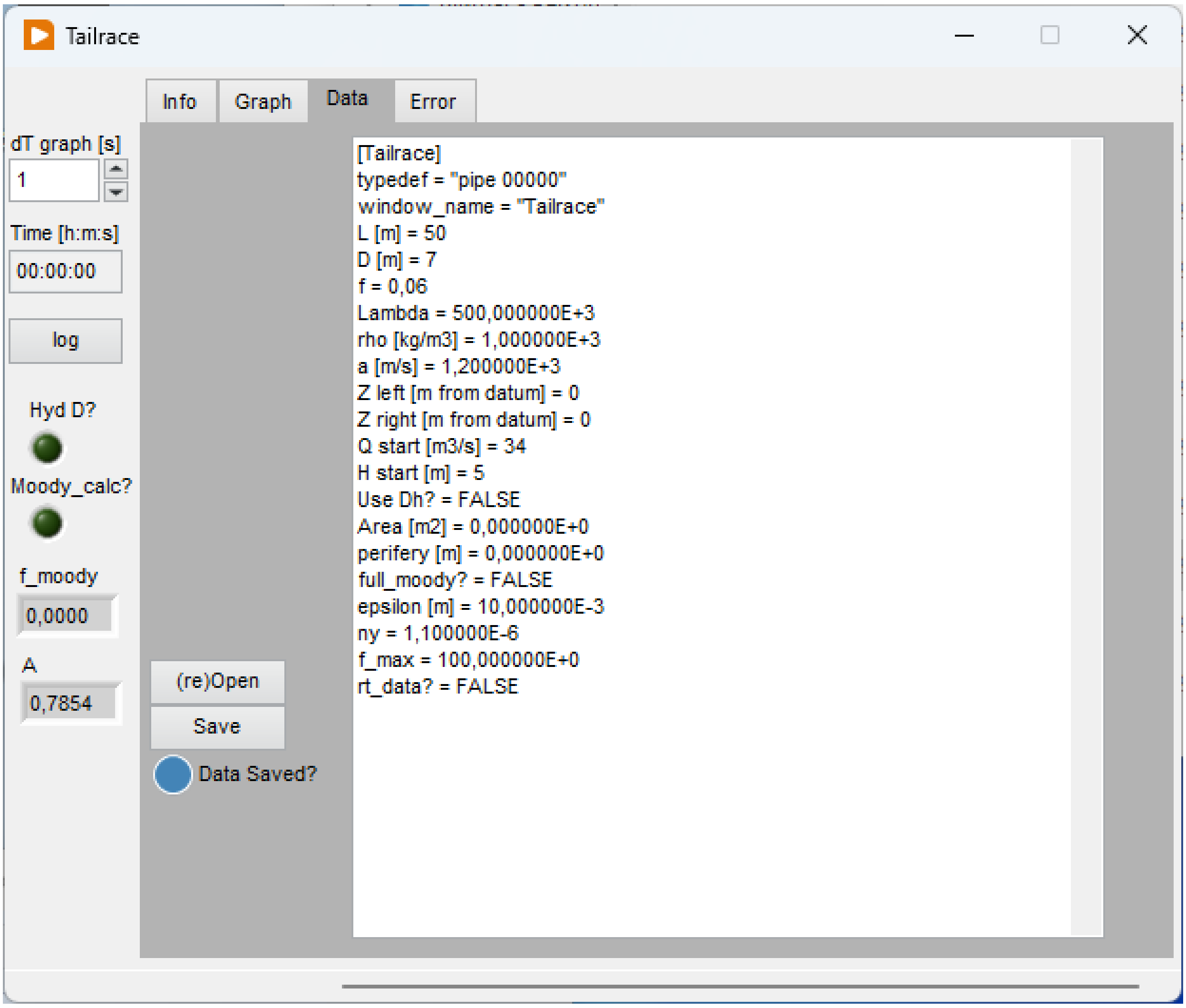


Figure 6.24 Data for the Tailrace tunnel

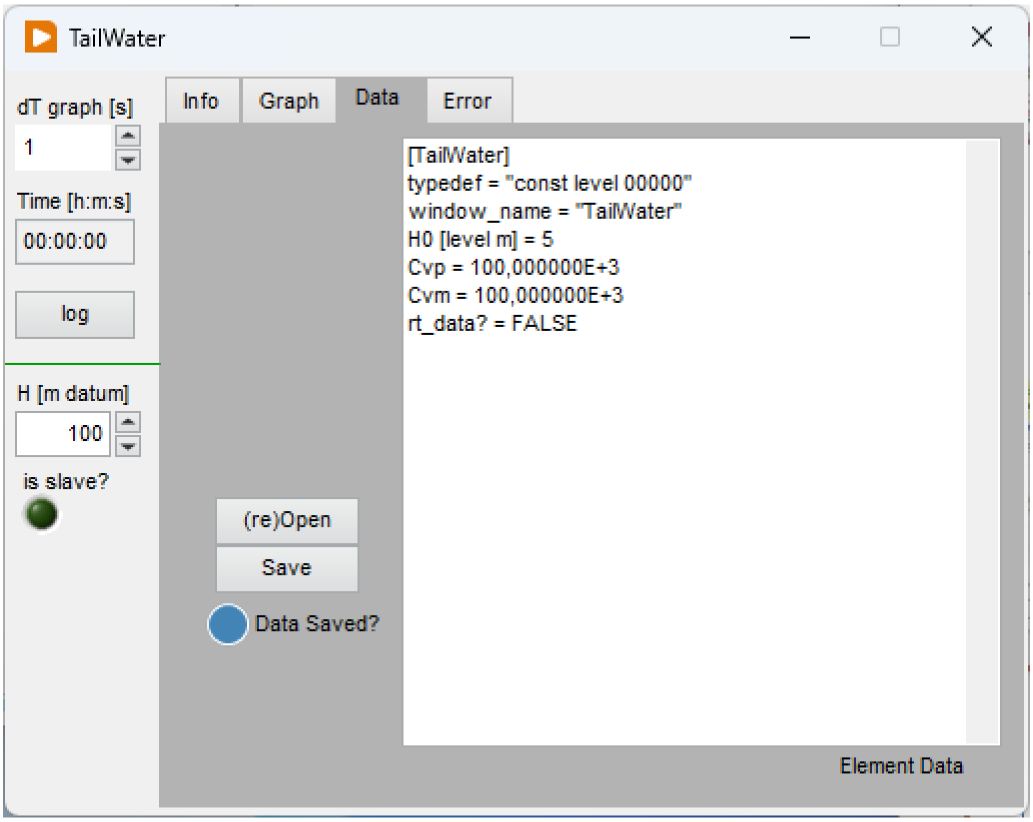


Figure 6.25 Data for the tail water

When all this is done, and checked, one can start doing some analysis.

7 NEW UPDATE OR COPY POWERPLANT.VI

As touched upon earlier the file in the main folder called "New update or copy powerplant.vi" will do the following, see Figure 6.3:

- Create a new powerplant from scratch
  - It will create a new folder with a user defined name in the "User Files"
  - It will create the named main file and the named system file from templates.
- Copy existing powerplant
  - Creates a new folder with a user defined name in the "User Files"
  - Copy an existing powerplant to this new folder and change its name accordingly.
- Update from an older version
  - Insert the system file in a new main file template
  - Possibly update the system file (user defined) for an old version made in LVTrans 20XX.02.15 or older. Mark the "Older than 20XX.02.15" check mark.
  - No new folders or new files are created.

The create new powerplant is described earlier. "Copy" and "Update" function much in the same manner. First you have to choose the main file, then the system file when asked by the program, as shown in Figure 7.1

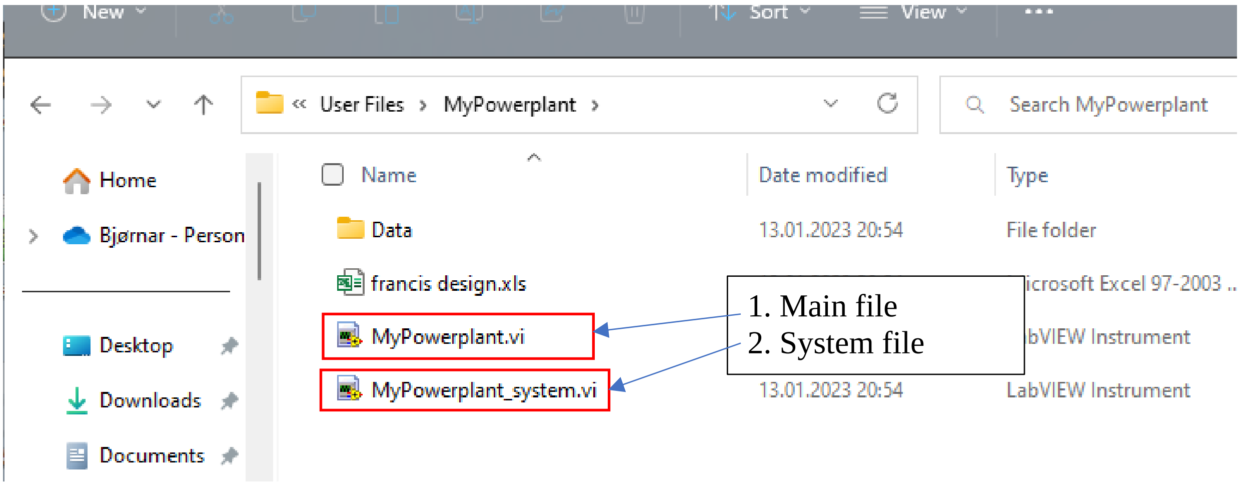


Figure 7.1 The MyPowerplant folder in the User Files folder

Must read: Version 20XX.02.22.11and upward use a slightly different internal wiring. For this to work properly on older versions, the plant must be **updated** with the "New update or copy powerplant.vi"[10]. Also note that the newest version works in LabVIEW 2021 and later (this is older than the 2022 version required for the previous LVTrans version).

[10] It's fully possible to do it manually as well, but not without basic LabVIEW programming skills.

8 RUNNING THE SYSTEM

If the tutorial has been followed, LVTrans is now running. To «reset» everything, and to be sure all the new values are read, press Stop, and then Stop Program in the MyPowerplant window. Push the white arrow again, set Execution speed to "max", and press Start. LVTrans starts and runs. The time step can be set to 0.01s for more accuracy and to prevent the program from running way too fast to follow.

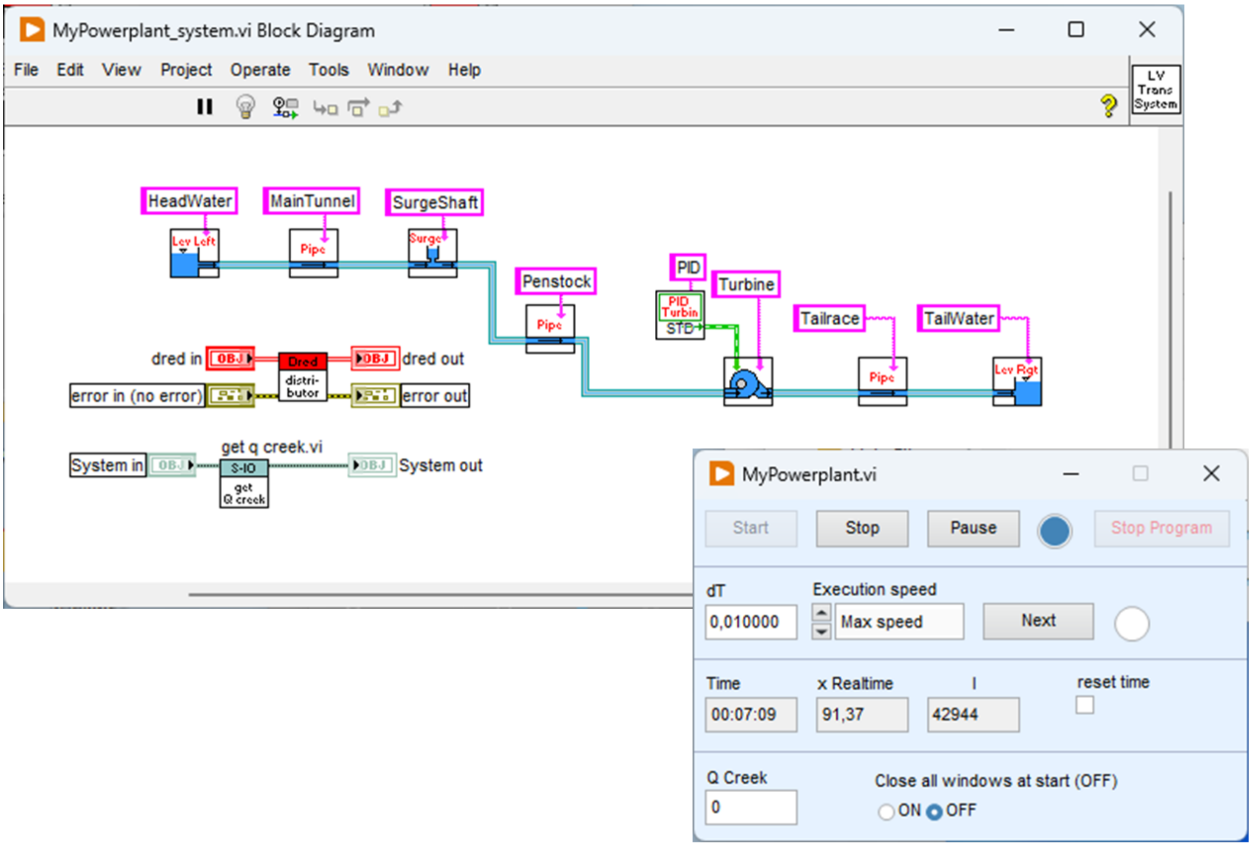


Figure 8.1 The system block diagram and main program opened and running

When running/calculating a good idea is to have the block diagram of the system open and at the same time the front panel of the main program open as shown in Figure 8.1. The main program is used to start, stop pause etc, while the block diagram of the system is used to open the elements for input and output and see what is happening.

To actually see what is happening, apply changes and so on, just double click on the element you want to see. The block diagram of that element will appear in a new window. For instance, we can change the load in the PID and look at how this affects the surge shaft and the turbine. This is shown in Figure 8.2.

Note! As default all governors starts in manual with an opening of 1.0, or whatever is set in the manual_default variable in the data file. This is to prevent instabilities when starting with way out parameters. Be sure to unset the "kappa manual" switch when running, see Figure 6.23

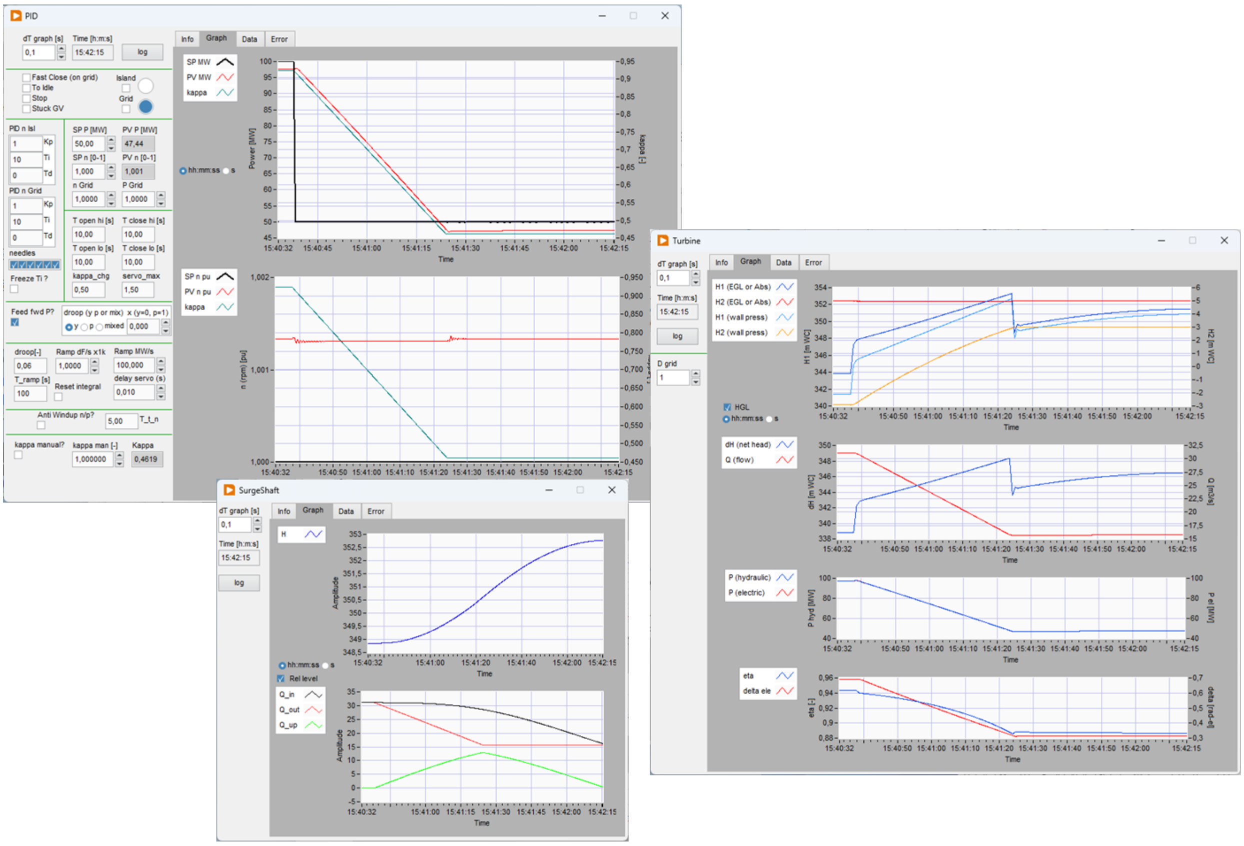


Figure 8.2 Changing power setting of the PID/turbine

The plant can be run as a real plant. Everything is interactive and every change is seen immediately. Everything on the front panel of each element with an option to adjust, can be adjusted during running.

One can also log data. This is done with the log button in every element. Time series will then be logged to a text file which can be opened later in for instance Excel. One can also just use the graphs and copy/paste into reports.

The PID can be run in two modes; Island and Grid. Switching between these modes is done by pushing either button.

9 ABOUT TIME STEP

One very common question is “What time step shall I use?”. There are no simple answers to this question, but a whole lot can be said.

All the elements in LVTrans except the free surface channels, are numerically stable, and no artificial numerical artifacts exists. This is one of the main advantages of MOC. It is unconditionally stable, and analytically correct regardless of time step. Shock waves (water hammer) are calculated analytically correct if the wave speed velocity, a, is within ±20 %, **and this is also regardless of time step**.

However, this doesn’t mean that we can extract the solution in a meaningful way regardless of time step. High frequency details will become completely masked and disappear if the time step is too large. Pole oscillations of the generator is a typical example. Steady state will in general always be correct. For frequency response analysis using FFT, as is done in LVTrans to create AFF diagrams, the time step becomes very important as explained below.

9.1 TIME STEP IN LVTRANS

- Shorter time step gives more detail in both time and space (pipe length) due to the use of MOC. $\Delta x = a\Delta t$, thus, smaller Δt gives smaller Δx.
- Shorter time steps will result in longer computational time
    - Exponentially when using ordinary MOC pipe elements.
    - Linearly when using algebraic MOC pipe elements.
    - Linearly for all non-pipe elements.
- MOC is unconditionally stable, any time step will in theory work, but:
    - There exists a largest Δt that will divide a pipe in two. All pipes must be divided in at least 2 due to the use of staggered grid[11]. If L=100 m and a=1000 m/s, then the largest time step is: $\Delta t \leq {}^{\frac{L}{2}}\!/_{a} = {}^{\frac{100}{2}}\!/_{1000} = 0.05\ s$
    - In LVTrans adjustments of pressure wave velocity, a, is done to make the relation $\Delta x = a\Delta t$ true for all arbitrary lengths[12]. This adjustment should however not in general exceed ±20% of the value of “a” for any pipe in the system, at least not when shock waves (watherhammer waves) are to be calculated. The value “a used” can be seen on the front panel of the pipes to check this.
    - Full freedom is given in LVTrans to define lengths and pressure wave velocity. If very short time steps shall be used due to short pipes, then consider using the algebraic pipe[13] for longer pipes to improve calculation speed. All tunnels placed somewhere between a surge shaft and a reservoir are preferably modelled with algebraic elements when speed of calculation is a factor.
    - If water-hammer considerations are unimportant. For instance, one wish to calculate steady state or just slow transients, then adjustment of “a” much further than ±20% can be used with no measurable error. The reason for this is that for low frequencies, it is the mass (momentum) that is dominating. The mass of water in each pipe will be correct regardless of adjustments in “a”.

[11] A pipe could also theoretically be divided in 1 sub-element, but this would not be a staggered grid anymore. In LVTrans the minimum division is 2. For the algebraic method, all pipes are divided in 2 regardless of length.
[12] This is the common method according to literature.
[13] The algebraic method is a “full-fledged” MOC with the same analytical accuracy, the main difference being a discretization backward in time rather than along the pipe in space.

- Non-pipe elements may have instability issues if Δt becomes too large. It is impossible to say at which Δt this will happen except in the extreme case when the oversampling ratio becomes less than 2.0 (see below). Larger Δt then 0.5s should be considered carefully.
- Pole oscillations in the generator are generally not seen unless Δt approaches 0.001 s.
- The default Δt in LVTrans is 0.1 s. This was set ages ago when a PC only had a fraction of the computational power of today. Consider using smaller time steps as a "personal default". A time step of 0.01 s will give accurate waterhammer calculations for pipes of length 20m or larger typically. Do however remember that a pipe can only be divided in n elements, where n is 2, 3, 4 and so on. Only whole numbers. Always take a look a "a used".
- Time steps must be set regarding closing/opening time of servos. If the closing time of a servo is 3 s, then those 3 s will "only" be dicretized in 30 discrete units if the time step is 0.1 s. This will increase to 300 for a time step of 0.01 s.
- Open surface elements are the only elements in LVtrans that has real stability and accuracy considerations for the numerical code. One aspect of this regarding transients (within the element) can be seen by observing the numbers "dx/dt" and "C" on the front panel of each element when the "Graph" tab is chosen. "dx/dt" should be less than "C". C is given as: $C = |V| + \sqrt{g\frac{A}{T}}$ where V is the velocity, A is the cross-sectional area of the water and T is the width of the water surface. Simply decreasing dt without creating more divisions in length (n) will not make it more unstable but will instead smear the solution out in space. Steep transient fronts will become less steep, not due to physics, but due to artificial numerical convection. This is improved by including more divisions, larger "n" in the input file until dx/dt becomes smaller than C. It's up to the user to decide if this is important or not, in most cases for hydropower it is not. If the analysis includes tracking these waves, it is of outmost importance, and then one should also look at the Fr number (Froude number).

9.2 TIME STEP FOR FREQUENCY RESPONSE ANALYSIS.

LVTrans can be viewed as a digital "machine". Aspects of the solution can then be analyzed in terms of:

- **Oversampling ratio**: The ratio of the sample rate to the maximum frequency we want to calculate/simulate.
- **Time latency** between input and output.

For LVTrans the sample rate is always 1/Δt, and the time latency is at most Δt, for instance between certain inputs and outputs.

A rule of thumb for digital sampling is:

- **An oversampling ratio of 10 creates a maximum error of 5%.**
- **An oversampling ratio of 32 creates a maximum error of less than 0.5%**.

5% may sound like much, but it is less than the error from using the Moody diagram. This is the maximum possible error, not the average.

For instance, with a time step of 0.1 s the sampling rate becomes 10 Hz. Oscillations at 1 Hz (oversampling ratio of 10) will give a max error of 5%. One should keep in mind that the cross-over frequency for most powerplants lies between 0.01 Hz and 0.1 Hz. What happens above 1.0 Hz isn't all that important.

Table 9.1 Time steps vs accuracy for AFF diagrams.

Time step [s]	**Sampling rate [Hz]**	**Max frequency, 5% error. Oversampling ratio of 10**	**Max frequency, 0.5% error. Oversampling ratio of 32**
0.1	10	1	0.3125
0.05	20	2	0.625
0.01	100	10	3.125
0.005	200	20	6.25
0.001	1000	100	31.25

From Table 9.1 we can get the time step and at which frequency this error enters. For higher frequencies the error will increase. For a fast analysis to get an overview, a time step of 0.1 will do just fine. For a nicer plot and higher accuracy, a time step of 0.01 is better. This will give a max error of 0.5% at 3.125 Hz and an error of 5% at 10 Hz.

9.3 TIME STEP WITH PID

All the PID elements in LVTrans are separate elements and calculated separately. One thing to remember about this is that the PV (Process Value) for the PID comes from a time step earlier than the time step in which the PID is calculating.

This is exactly what happens in a real-life system. All governors today are digital governors with their own sampling rate. The sampling rate of a modern governor is typically between 10 to 100 Hz, which is a time step of 0.01 to 0.1 s. A real governor could also have additional latencies for calculations. To complicate the issue further, there are always some filtering of the input signal, or a PID would simply not work, and there are always some signal latencies. Bounds for these variables could be defined in standards, but the exact values are not. The error this represent is a phase shift of the input signal due to the time latency, and an unknown error due to filtering. In addition, there is backlash which complicates further and can sometimes have large effects. Only in recent time have these factors started to get the recognition they deserve.

It's literally impossible to include all this accurately without “inside information” from manufacturers in each case and/or with direct measurements. What can be done is to quantify the effect this has in LVTrans in relation to sampling rate and oversampling ratio for a sine wave.

In LVTrans the time latency is always Δt. This time dilation is a pure phase shift. For hydropower plants the cross over frequency is typically somewhere between 0.01 Hz and 0.1 Hz, sometimes above 0.1 Hz. This is also where the phase margin and gain margin is found. We are interested in finding what error the time step has on these frequencies to be able to choose an adequate time step. The error is the average calculated as the absolute difference in % of the amplitude of the full sine wave.

Table 9.2 Time steps vs error due to time dilation with no predictor.

Time step [s]	Sampling rate [Hz]	Error at 0.01Hz [%]	Error at 0.1 Hz [%]	Error at 1 Hz [%]
1	1	2.0	19.0	NaN[14]
0.5	2	1.0	10.0	NaN[15]
0.1	10	0.2	2.0	19.0
0.05	20	0.1	1.0	10.0
0.01	100	0.03	0.2	2.0
0.005	200	0.015	0.1	1.0
0.001	1000	0.003	0.03	0.2

The max and average error due to time dilation of Δt in LVTrans is shown in Table 9.2. For a time step of 0.1 s one can a relative accuracy of 2% at 0.1 Hz. A more reasonable time step of 0.01 s gives max 2% error all the way up to 1 Hz.

Although the max error for time dilation is small, this can be improved by using forward Euler extrapolation of the two previous time steps instead of the previous time step alone as given by:

$$Y_{predicted} = 2Y_0 - Y_{-1}$$

This is for the time being hardcoded into LVTrans at places where it does not cause numerical instabilities. For a sine function, the error now becomes a function of the oversampling ratio only. The oversampling ratio must be larger than 2.0 for stability.

Table 9.3 Oversampling ratio vs max and average error due to time dilation with predictor.

Oversampling ratio	Max error [%]	Average error [%]
2.1	198	126
5	66	42
10	18	12
20	4.9	3.1
32	1.9	1.2
50	0.8	0.5
100	0.2	0.1

An oversampling ratio of 32 gives an error less than 2%. For instance, a time step of 0.01 s gives a sampling frequency of 100 Hz. With an oversampling ratio of 32, this will give less than 2% error all the way up to 3.125 Hz which is an improvement over the direct method from Table 9.2. An oversampling ratio of 100, a time step of 0.01 gives an error of less than 0.2% at 1 Hz. An improvement by a factor 10.

9.4 TIME STEP CONCLUSION AND RECOMMENDATIONS

Any time steps will work in most circumstances, but not all time steps will give the result you want. Smaller time steps are always better, but the improvement may only be academic. Rules of thumb are:

- A time step between 0.01 and 0.1 will in most circumstances be OK

[14] Impossible to have frequency of 1 Hz with a sampling frequency of 1 Hz.
[15] It must be more than 2 samples per period.

- A time step of 0.01 will give a total numerical error of less than 0.28 % at 1 Hz for AFF analysis and approaching zero for lower frequencies.
- A time step of 0.1 will give a total numerical error of less than approximately 3% at 1 Hz for AFF analysis and approaching zero for lower frequencies.
- For AFF analysis an oversampling ratio of 32 or higher should be used.
- Consider using the algebraic pipe for long tunnels having a reservoir in one end and a surge shaft in the other end.
- Observe “a used” for the shortest pipes. This is of outmost importance when calculating waterhammers.
- Oversampling ratio and time latencies are of little concern when calculating waterhammers, due to the analytical correctness of MOC.

10 VARIATIONS OF ELEMENTS

Most of the elements have different variations. The «default» version works OK, but other variations may be useful in certain circumstances. This is in particular true for the governopr element. Note! Some of these variations may not work, or may do things in unexpected ways.

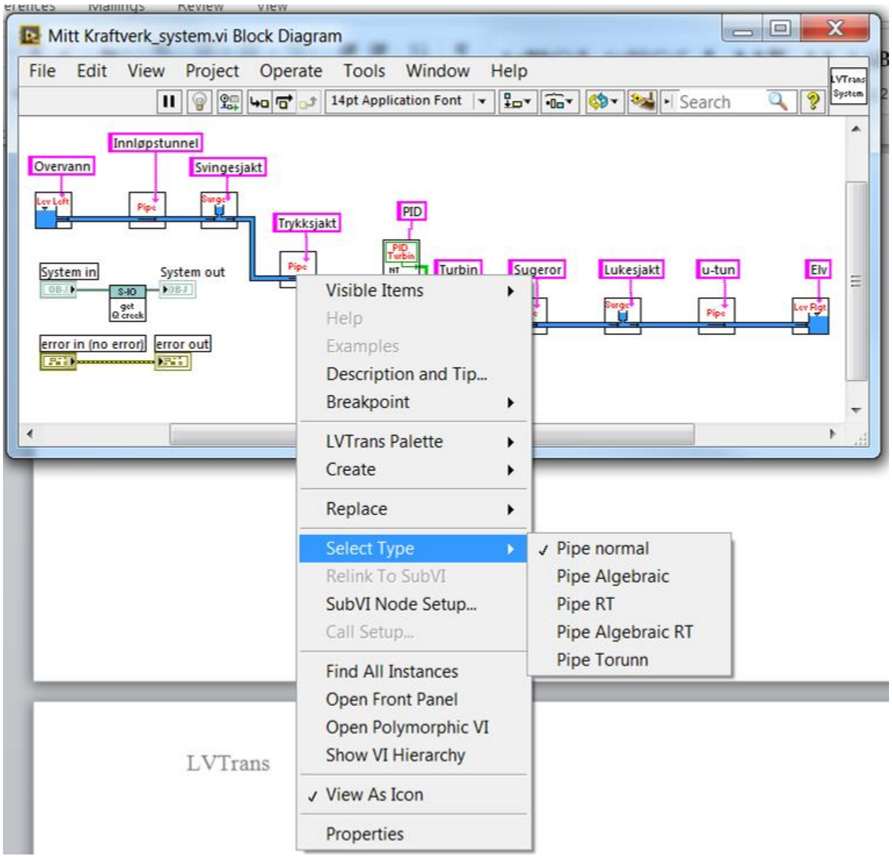


Figure 10.1 Variations of the pipe element (old figure)

To see and pick the version, right click on the element, then select type and pick the one wanted. With the governor element this is necessary for certain tasks, like frequency response analysis. The way this is programmed is simply to use the built in polymorphic vi methods in LabVIEW. This is a point worth remembering. Everything is ordinary LabVIEW and things can be changed and modified at will. If you don't like the look of the graphs, change them to your liking.

10.1 PELTON TURBINE

Pretend the turbine in Figure 6.1 was a Pelton turbine. The system may then look approximately like Figure 10.2

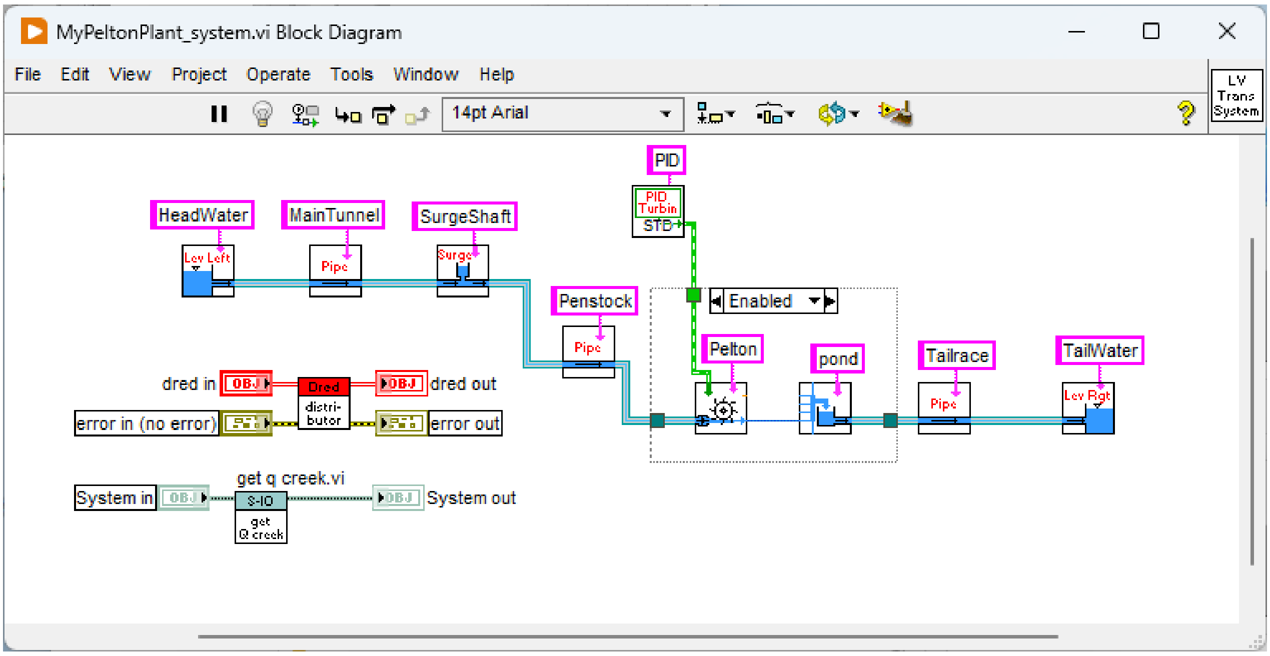


Figure 10.2 Pelton turbine plant

Figure 10.2 shows a simple Pelton plant. It also shows another very useful feature in LabVIEW, namely the “Diagram Disable Structure”. The Diagram Disable Structure makes it possible to have lots of different smaller sections in the system, like changing between Francis and Pelton without ruining an already working plant, and without making a brand new one.

The «Francis_design» excel file can be used also to «design” a Pelton turbine. Use the same parameters as the Francis turbine, but set «eta” to for instance 0.92 and lower the pole pairs to for instance 5, or whatever gives the correct RPM for a real plant. LVTrans handles the rest. It uses essentially the same equations as before, the Nielsen modified Eurler turbine equation. This is shown in Figure 10.3. Be sure to also update the PID with corrected values.

Note! This is just an example. It would not be natural to use a Pelton turbine on that head and flow instead of a Francis turbine.

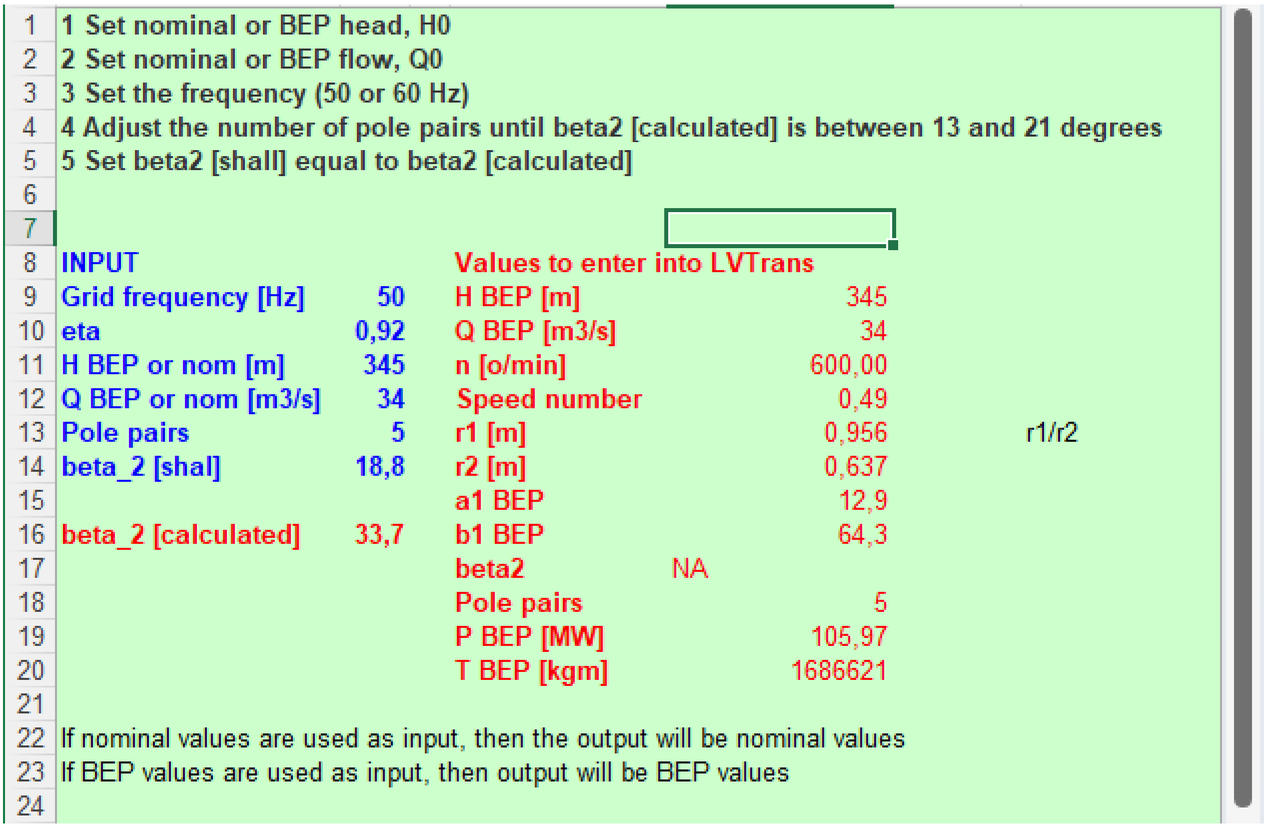


Figure 10.3 Parameters for a Pelton turbine

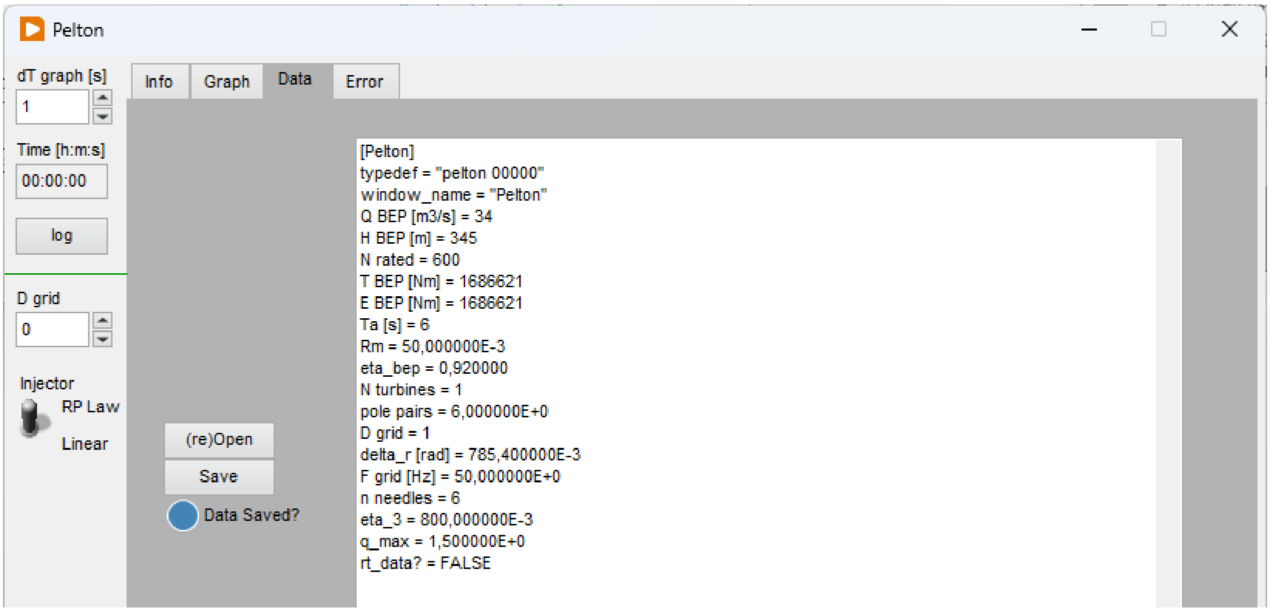


Figure 10.4 Data for a Pelton turbine

Pelton turbines have tail water directly out, or via a «sump» that leads to a tail race tunnel. This sump is in principle just a modified surge shaft or creek shaft. Several turbines can exit the flow to one Pelton Sump.

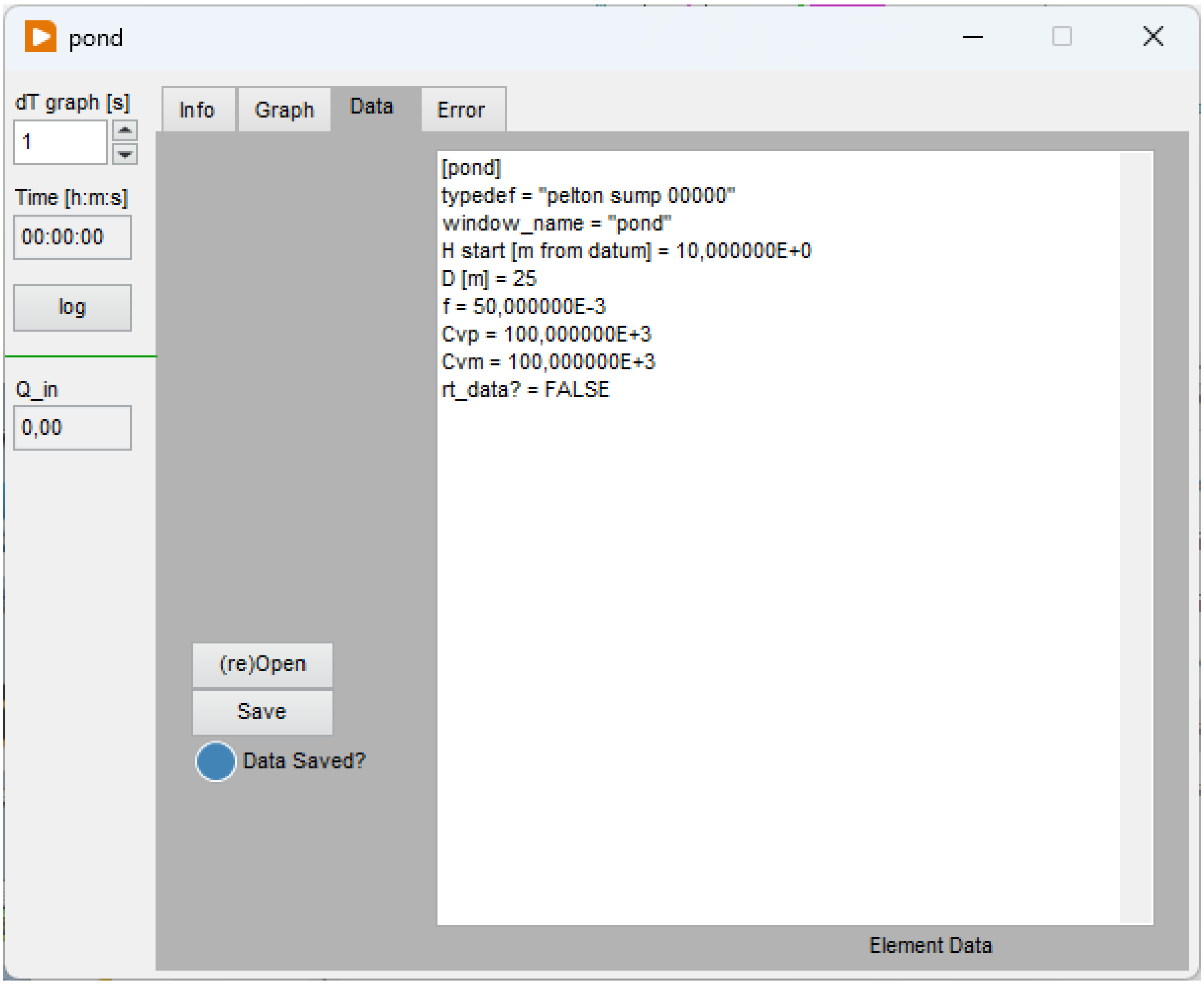


Figure 10.5 Pelton Sump

11 PID TUNING AND ANALYSIS

LVTrans includes a PID -tuning and –analysis ”toolkit”. This is based of frequency response of time series. This is done with FFT (Fast Fourier Transform).

Note! The pictures in this chapter are from a slightly older version of LVTrans, but close enough to not make confusion.

Several years ago, many different autotune functions were made. They are still there in the vi named "AFF Control.vi". This vi is in the top main folder. They were made by request from Statkraft. They work more or less but are **very** far from bullet proof. Today, AFF Control is used almost 100% exclusively to plot AFF diagram and show phase margin and gain margin. The tuning of the PID is done manually by trial and error. After several years of experience and investigations, a fool proof method is found. Compared with everything else, this method is faster and gives more optimal PID parameters. How good they are can be seen in hindsight by plotting the AFF.

The PID element gives full freedom to try and test different parameters. One can also modify the whole PID element to ones choosing.

A note about PID tuning. There exist lots and lots of different trial and error methods and more academic methods. Some are more known than others. The academic method are usually all s-plane methods, while the trial-and-error methods are typically time domain methods. The academic methods works well when the system is well behaved after some metric. Usually, it means the plant can with good accuracy be approximated with some polynomial of $s=j\omega$. The problem with hydropower is that a system only exceptionally can be approximated in such a manner. Time delays and saturation usually governs what happens, but exactly how *can* vary from plant to plant, and usually does exactly that. This is also reflected in grid codes. It is the end performance that matters (stability, accuracy, speed) according to grid code. Thus, tuning a PID is more a matter of meeting and documenting the performance in whatever manner this is specified in the grid code. In later years this means not only on isolated grid, but also on the common grid. This manual covers only basic PID on isolated grid, and with no specific grid code in mind. Tuning to common grid will be too specific for each grid code and is outside of the scope of the manual.

There are two main PIDs for turbines in LVTrans. One standard and one linear, each with their own AFF version. They can be found by right clicking and “select type”.

- Standard (with or without AFF)
  - Power oriented droop or opening related droop, or a mix
- Linear (with or without AFF)
  - Power oriented droop or opening related droop, or a mix

Several other governors are also there, but they are made for very specific use and not explained in this manual. Standard in this case means all saturations in for instance servos are included. Linear means it is a pure mathematical function with no non-linearities or saturations. Both are often required for grid codes to show adherence to specific aspects of the code. They all use the same data file, so the PID can be changed at will.

The author has found no analytical method that is fool proof in any sense of the word. However, one trial-and-error methods works exceptionally well. It can be used for both PI and PID. An additional method also works, but only for PI governor. The author never uses the second method anymore but is included for completeness. The main procedure is therefore:

- Tune the PID on isolated grid using trial-and-error
- Check the transient response
- Check the AFF diagram (gain margin etc)

Note! The following method(s) is for tuning the PID on isolated grid. It does not necessarily reflect any requirement a grid code may have regarding performance, nor the parameter setup described in the code to verify the calculations, for instance droop. However, to this day there have not been a single case where method number 1 has not created a set of parameters that will not satisfy a grid code on isolated grid. As far as the author is concerned, this method creates optimal PID parameters regardless of powerplant or grid code. If good parameters are not found, then this is due to a poorly designed plant (too small Ta, unstable surge shaft etc) is the author's experience. *This is an opinionated opinion considered to be "true" until a better method is found, or the method is proven wrong. It's also worth mentioning, and certainly important to keep in mind, that tuning a PID is sometimes referred to as black magic.*

11.1 PID TUNING – METHOD 1

This is the main method used by the author. This method has never failed. A prerequisite for all tuning is that the speed of the servo is already calculated or found and set into the data file, which is a main part of any transient analysis[16].

1. Set the PID in island mode and P Grid to 85-100%[17]. Set **SP P [MW] =0 and droop = 0**.
2. Sett Kp = 1 and Ti = 10 (eventually adjust Ti up from 10, and Kp down from 1 until the system becomes stable). Set Td = 0.
3. Wait until everything is stable and settled (n = 1 og stable, non-oscillating or just slightly oscillating), see Figure 11.1
4. Set Ti = 10^{10} (a really big number. This makes it a P governor for the time being)
5. Do steps in **SP n** of 10% (0.1) and study the response of PV n. Up and down around pu (per unit) of 1.0 and steps of 0.1
6. Increase Kp a little bit and do point 5 until the overshoot reaches 15% of the step. See Figure 11.2. The size of the overshoot should be 15% of the size of the step in SP n.
7. Decrease Ti until the overshoot in PV n increases to approximately 30% of the step in SP n. Take a note of Kp and Ti at this point. See Figure 11.3
8. For a PI governor: Keep Ti from step7, then **decrease** Kp until the overshoot is OK, or around 5-20% of the size of the step in SP n. 15% is a good goal here. That's it, the PI governor is now tuned. See Figure 11.4
9. For a PID governor: Keep Kp and Ti from step 7, then **increase** Td until the overshoot is OK, or around 5-20% of the size of the step in SP n. 15% is a good goal here. That's it, the PID governor is now tuned. See Figure 11.5

[16] Not covered here, because it is assumed that most people wanting to use LVTrans know about servo speed, turbine speed and pressure rise.

[17] If BEP is used throughout, then 100% is the "correct" value because this will be about 85% of full power. If nominal values are used throughout, then 85% is "correct". The point is to tune the PID at **roughly** 85% of max power.

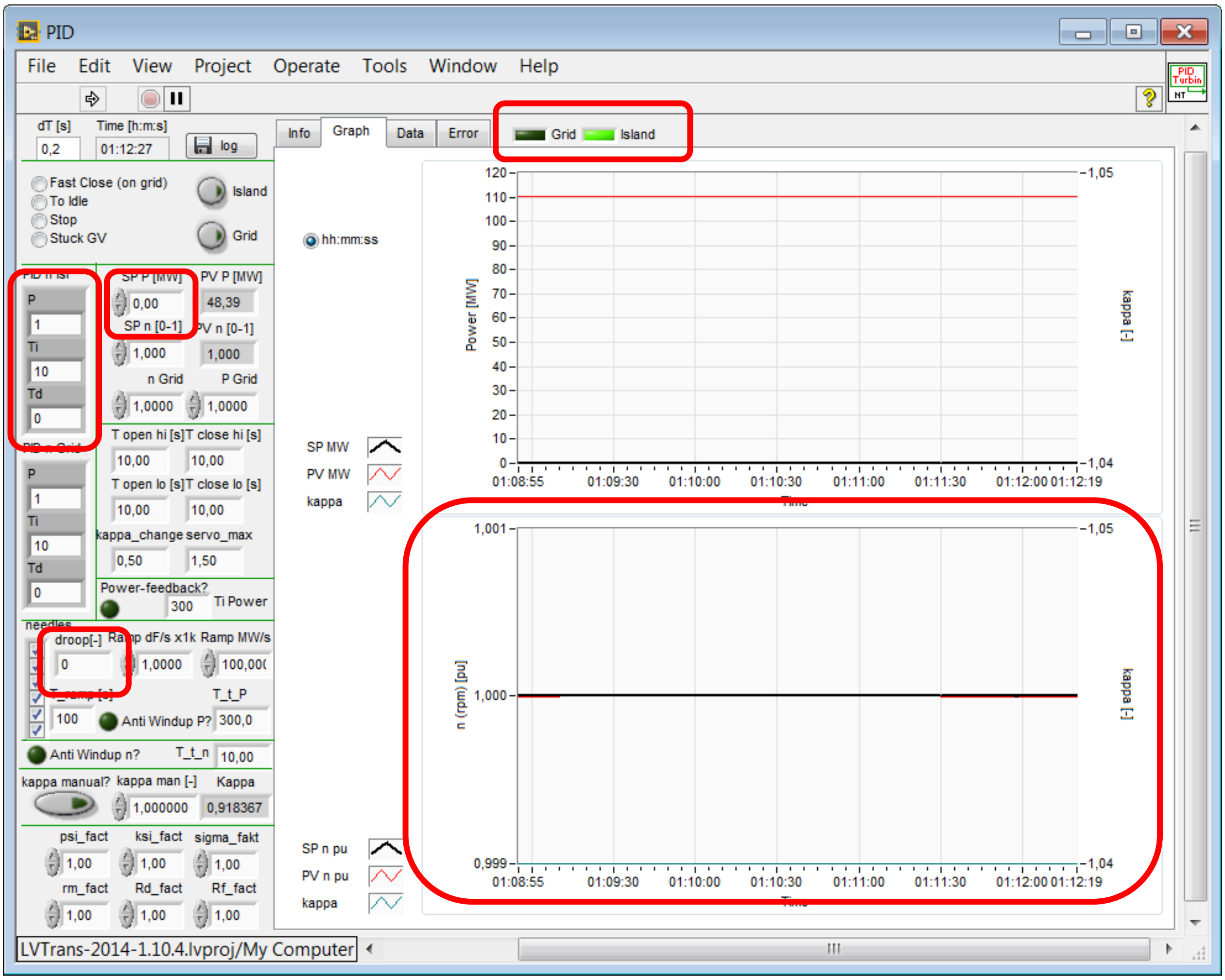


Figure 11.1 PID as set for point 3

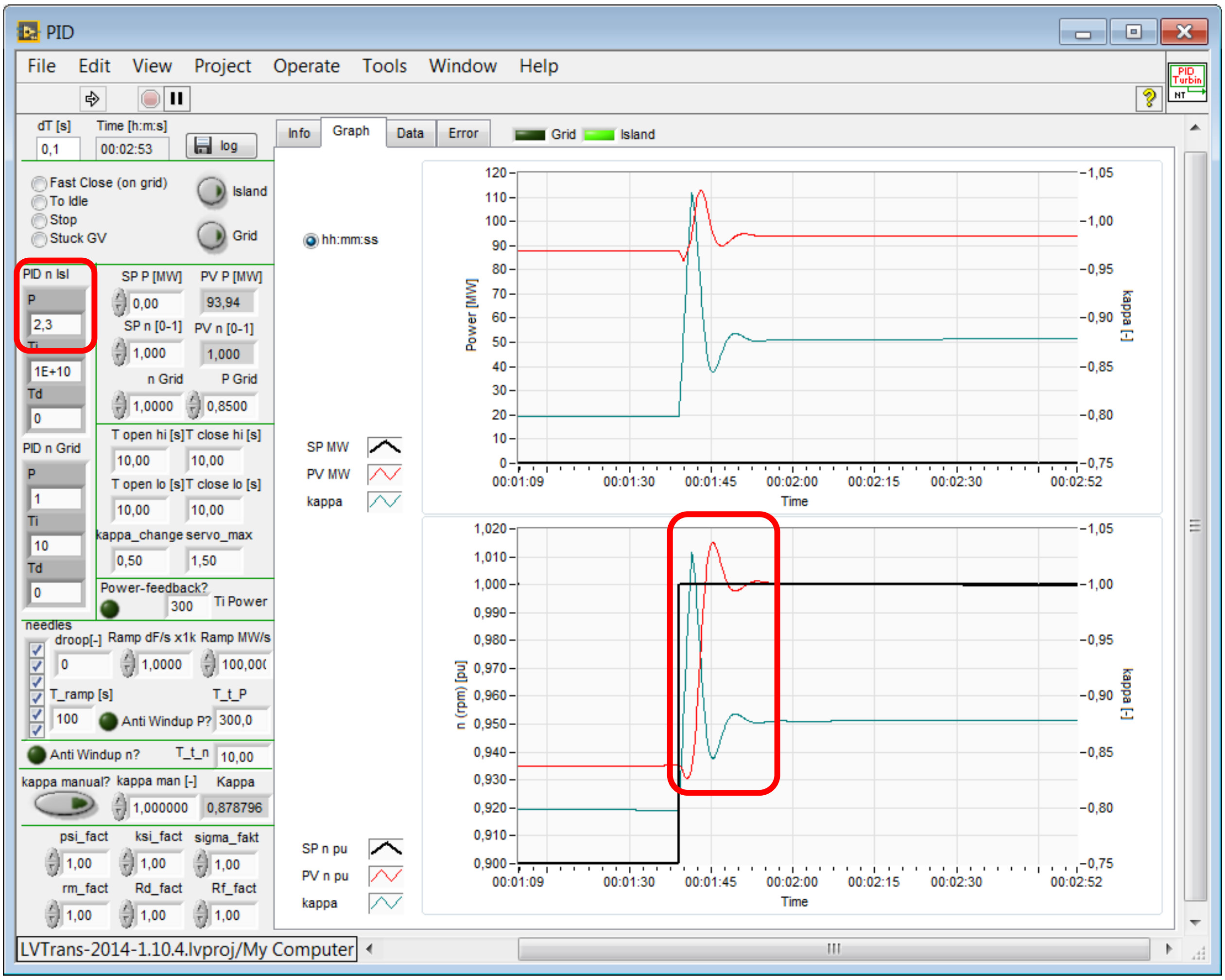


Figure 11.2 The governor as set for point 6

Figure 11.3 The governor set for point 7

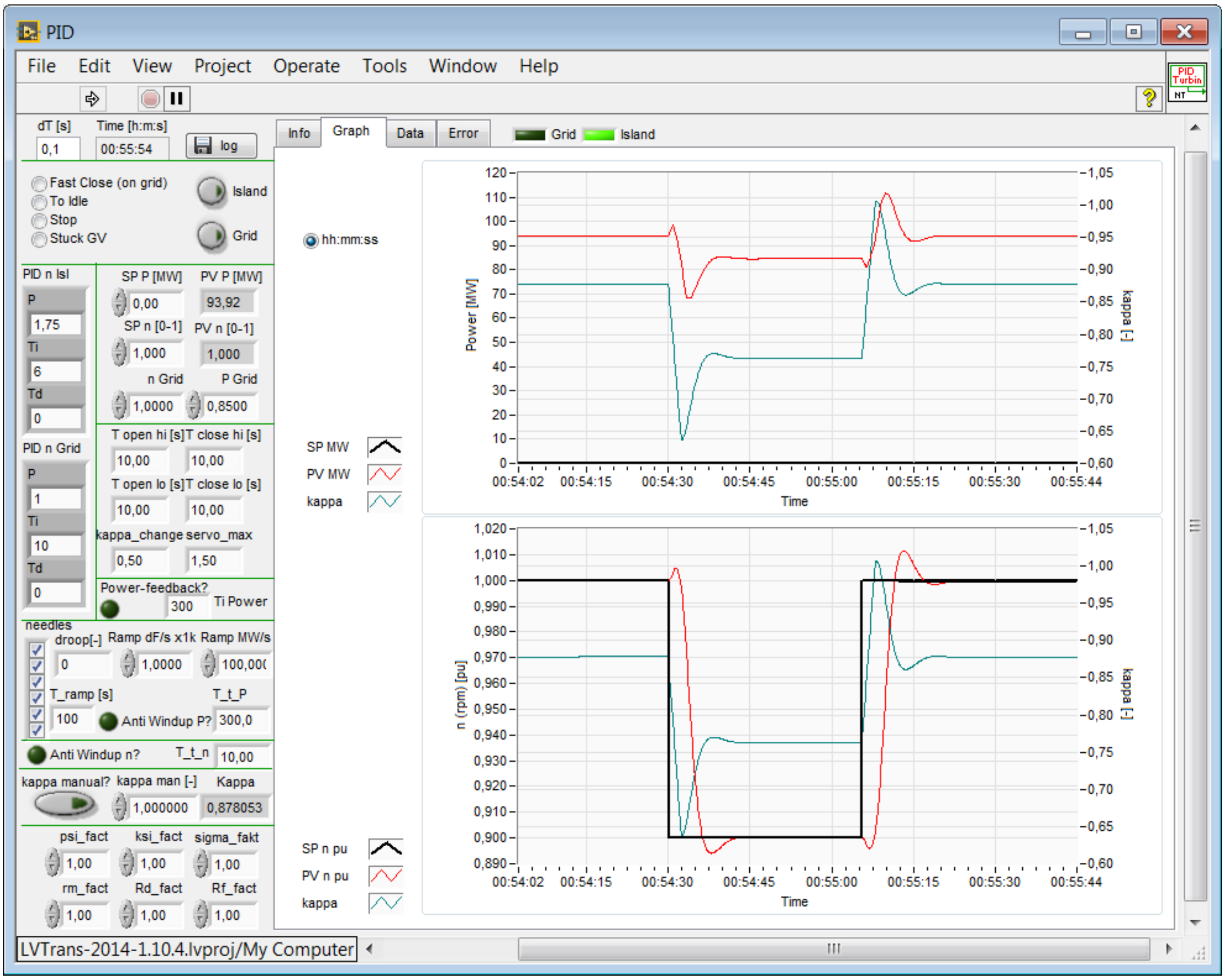


Figure 11.4 Perfectly tuned PI governor as in step 8.

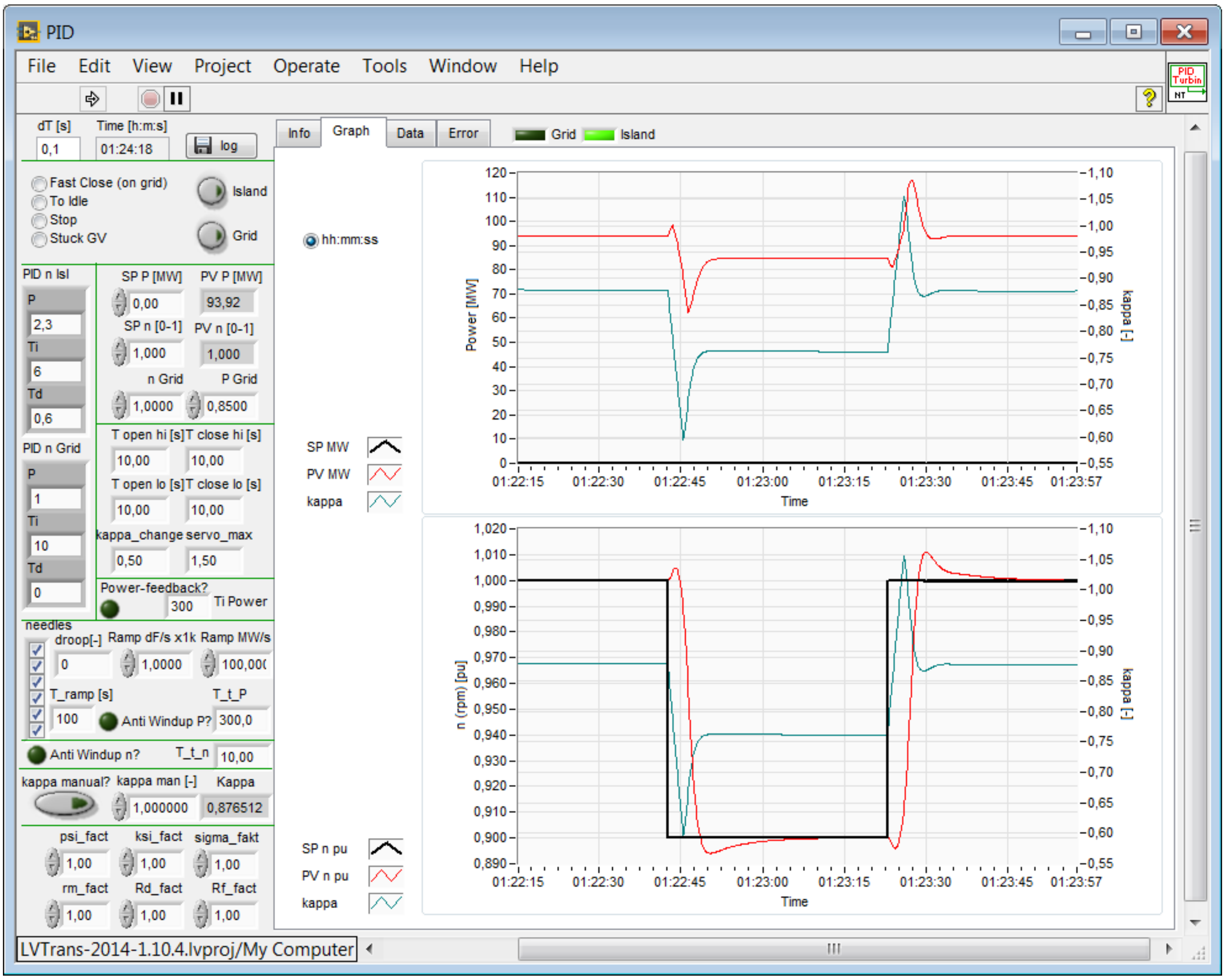


Figure 11.5 Perfectly tuned PID governor as in point 9.

11.2 PID TUNING - METHOD 2

This method is not used anymore. It is however a common method and works well enough and is included for completeness. It can only be used for PI, not for PID.

1. Set the PID in Island mode and approximately 85-100% P Grid (0.85-1.0), **SP P [MW] =0 og droop = 0**.
2. Set Kp = 1 and Ti = 10 (eventually adjust Ti up from 10, and Kp down from 1 until the system becomes stable). Set Td = 0.
3. Wait until everything is stable (PV n = 1 and stable).
4. Set Ti = 10^{10} (large value making it a P governor)
5. Do changes in SP n of 10% (0.1) and study the response of PV n (up and down around 1.0) as with the previous method. The point is to find a Pc (P critical) that transforms the system from underdamped to overdamped. Overdamped is when there are no oscillations after the step, underdamped is when more than one top is seen.
6. Increase/decrease Kp until you can see only one top. This requires a bit of practice and can be very difficult on some plants. See Figure 11.6
7. Set Kp = 1.2 * Pc (P = 1,2 * 1,5 = 1,8 in this example), see Figure 11.7
8. Decrease Ti until the overshoot is OK. As with the other method 10% is good.

9. For PID, keep the Ti from point 8 but with 15-20% overshoot. Then increase Td until the overshoot is OK (10% OK)[18].

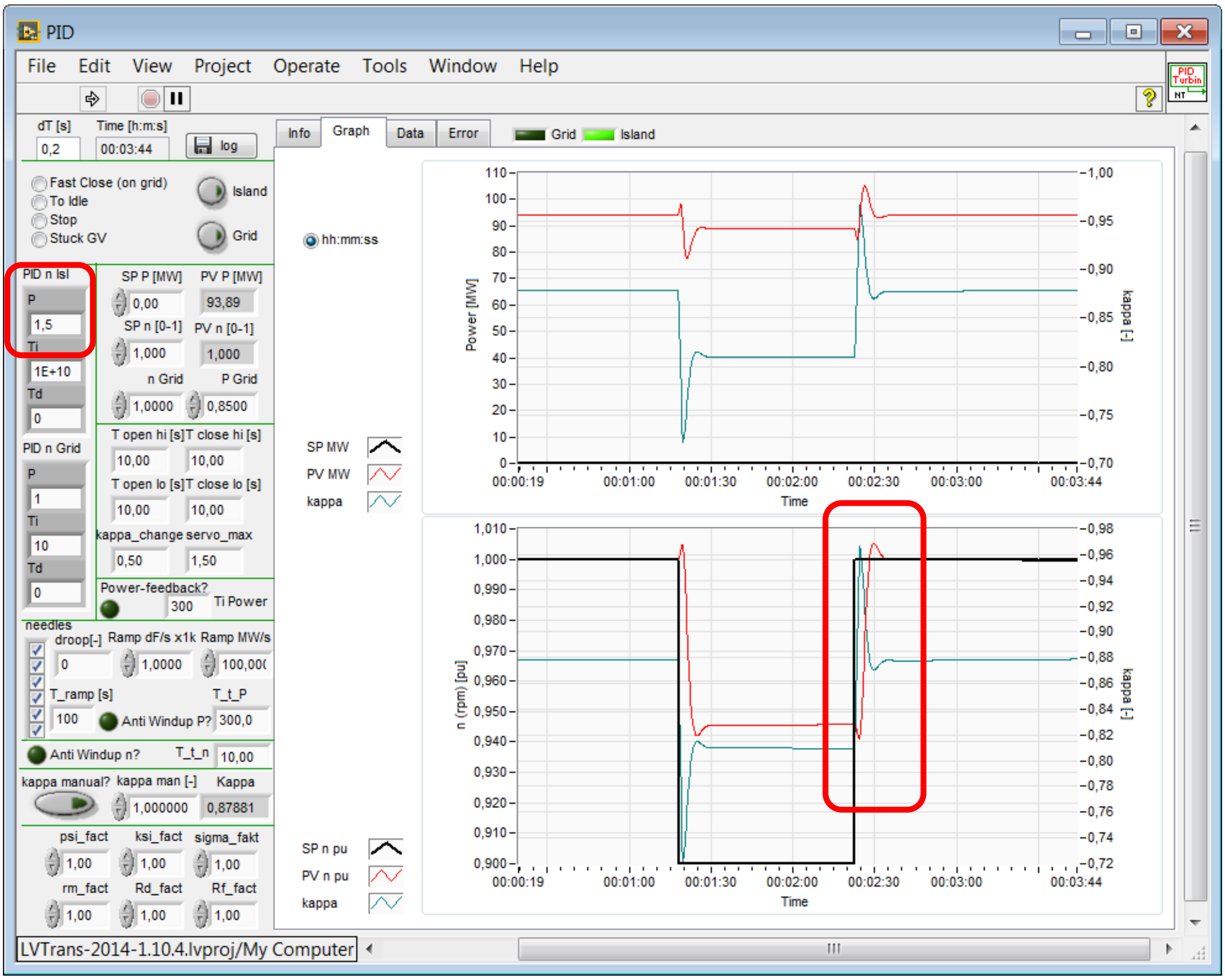


Figure 11.6 The governor as in point 6

[18] This is rather fudgy and does not always work, which is why the method only really work for PI governors.

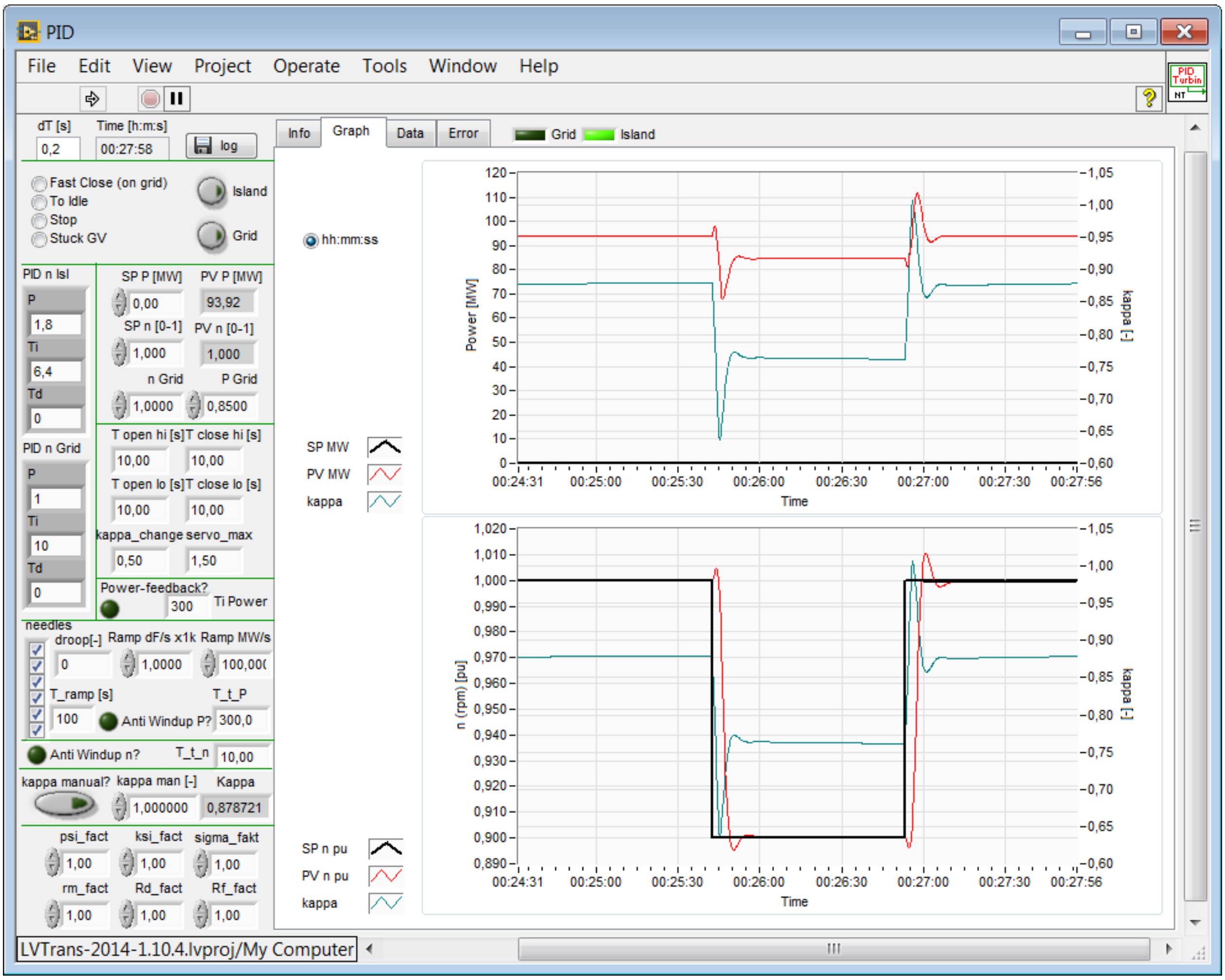


Figure 11.7 Finished tuned PI, point 8 after method 2

11.3 ANTIWINDUP

Anti-windup can normally be turned off, but when the servo is very slow, anti-windup will greatly enhance the governing.

Note! Never use anti-windup if the grid code specifies calculations with a linear governor.

The reason anti-windup works is because the servo reacts much slower than the governor would like it to act. This typically happens if the servo is slower than about 10s, and is a must if the servo is at 20s. What happens then is the integrator «winds up» when moving along the ramp of a slow servo. The governor will therefore overshoot, sometimes massively, before the integrator "winds it back" again.

Figure 11.8 shows the behavior without anti-windup when the load on the isolated grid goes from 100% to 20%. Figure 11.9 shows the same, but this time with anti-windup (no other changes are done). The improvement is rather large (see the lower graphs of speed, n)

Anti-windup is set by hooking off for anti-windup and setting an appropriate T_t_n. T_t_n must be less than Ti. A good value is 1/10 to 1/5 of Ti. Anti-windup can be set at the end, but really should be included into the tuning process, for instance when you start tuning Ti using the procedures above. A word of caution is that the effect of anti-windup may not be relevant in the particular grid code used. When including anti-windup in the tuning process, one may create parameters that are unstable without anti-windup. Often the open loop AFF diagram is used to verify stability. In open loop, anti-windup usually has no effect.

All governors have *some* kind of anti-windup, but it is not clear how this is implemented in one particular governor from one particular manufacturer. There are several ways this can be done, LVTrans use one method.

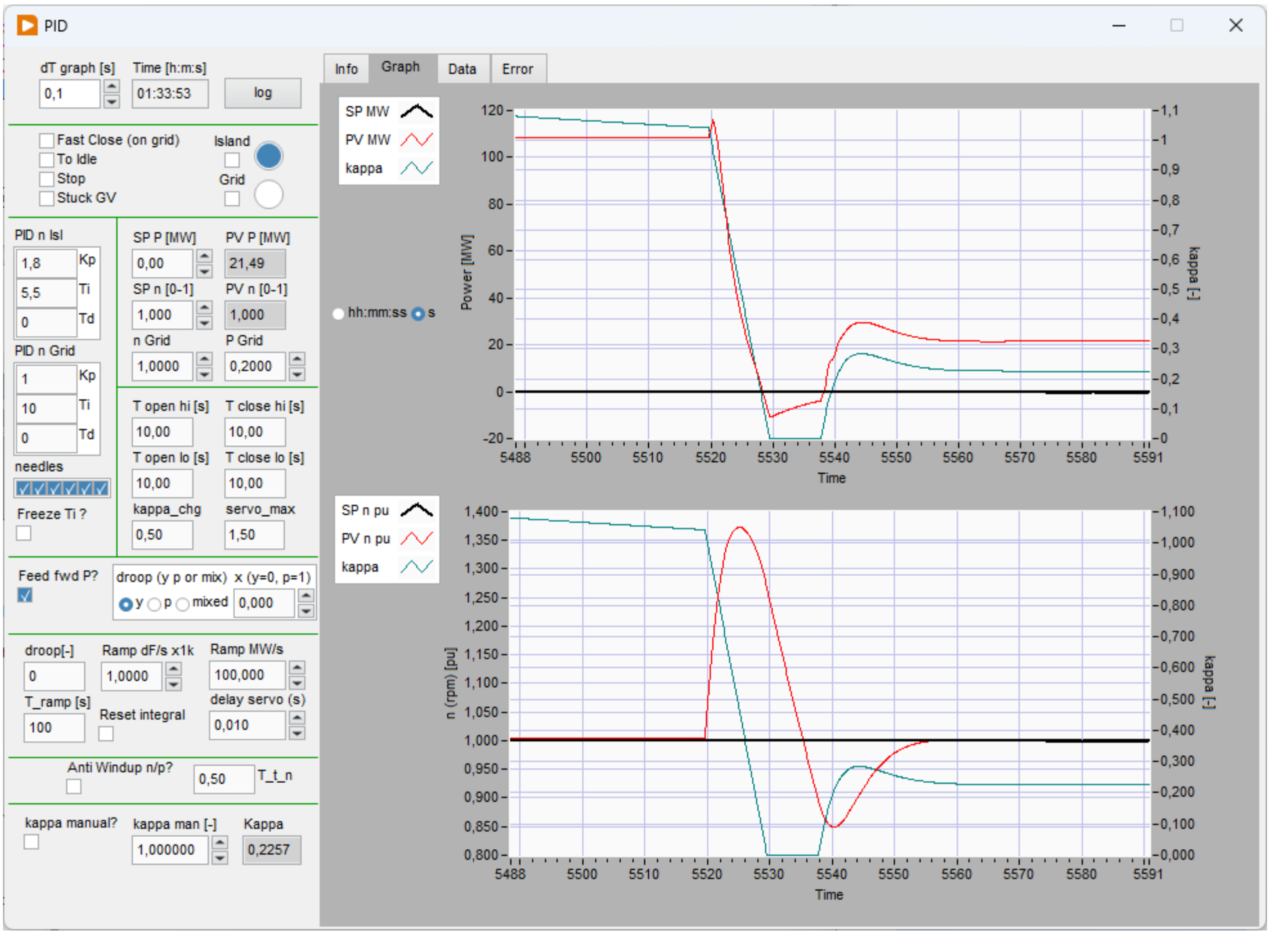


Figure 11.8 From 100% to 20% load on isolated grid without anti-windup

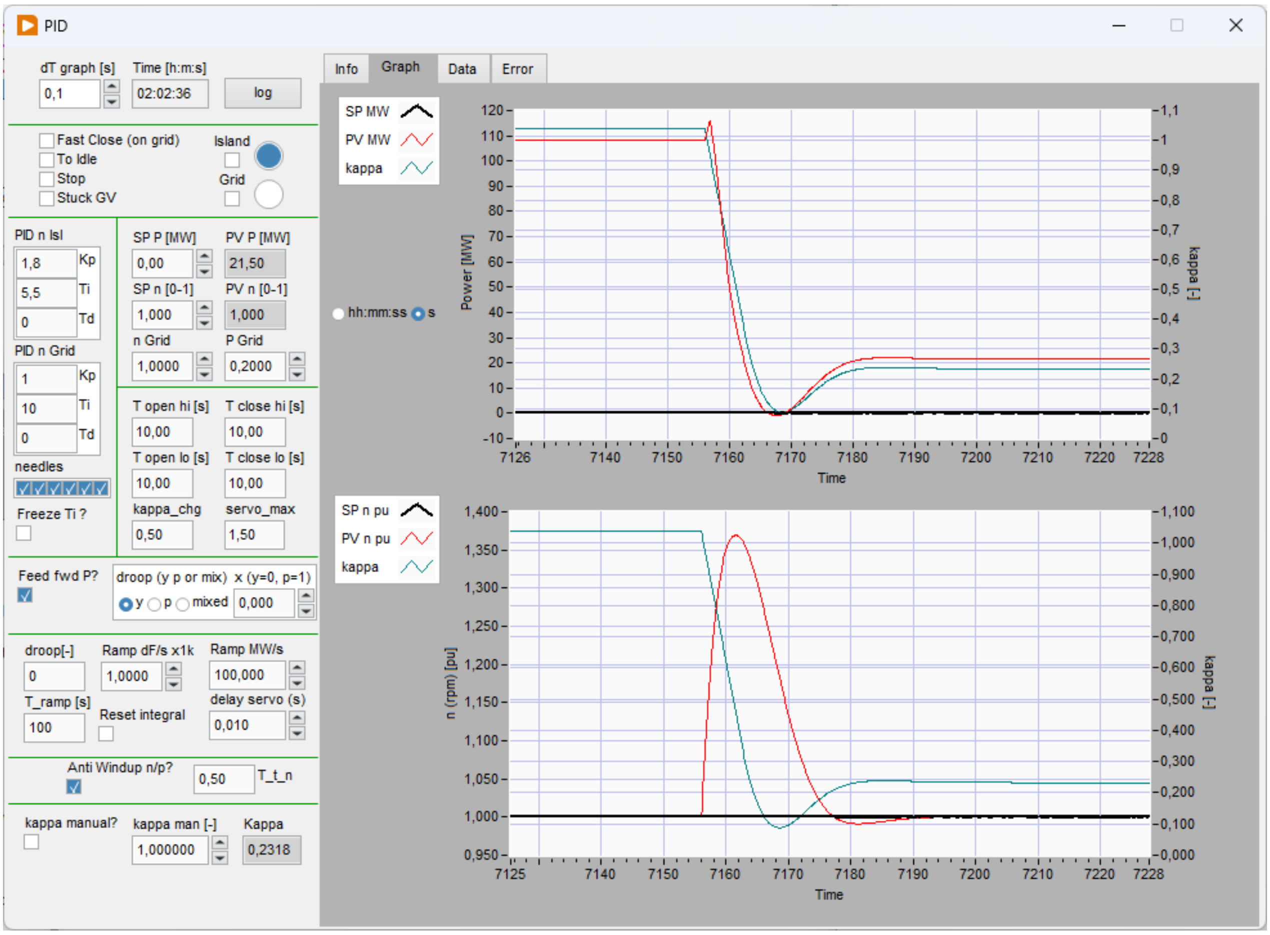


Figure 11.9 From 100% to 20% load on isolated grid with anti-windup and no other changes.

11.4 AFF DIAGRAMS

To plot AFF diagrams, a separate vi is made. This must be used alongside special versions of PID governors. The AFF vi is name AFF Control.vi and is in the topmost folder of LVTrans installation. AFF Control.vi is a general-purpose frequency response analysis tool and can be used to plot several different AFF diagram with several different points for exiting and measure.

The AFF version of the PID is found by right clicking on the governor in the block diagram, chose “select type” and chose an AFF type. The AFF type use the same data file as the normal type. A PID standard AFF is shown in Figure 11.10.

The AFF Control.vi is shown in Figure 11.11 To start it push the white arrow in the upper left corner.

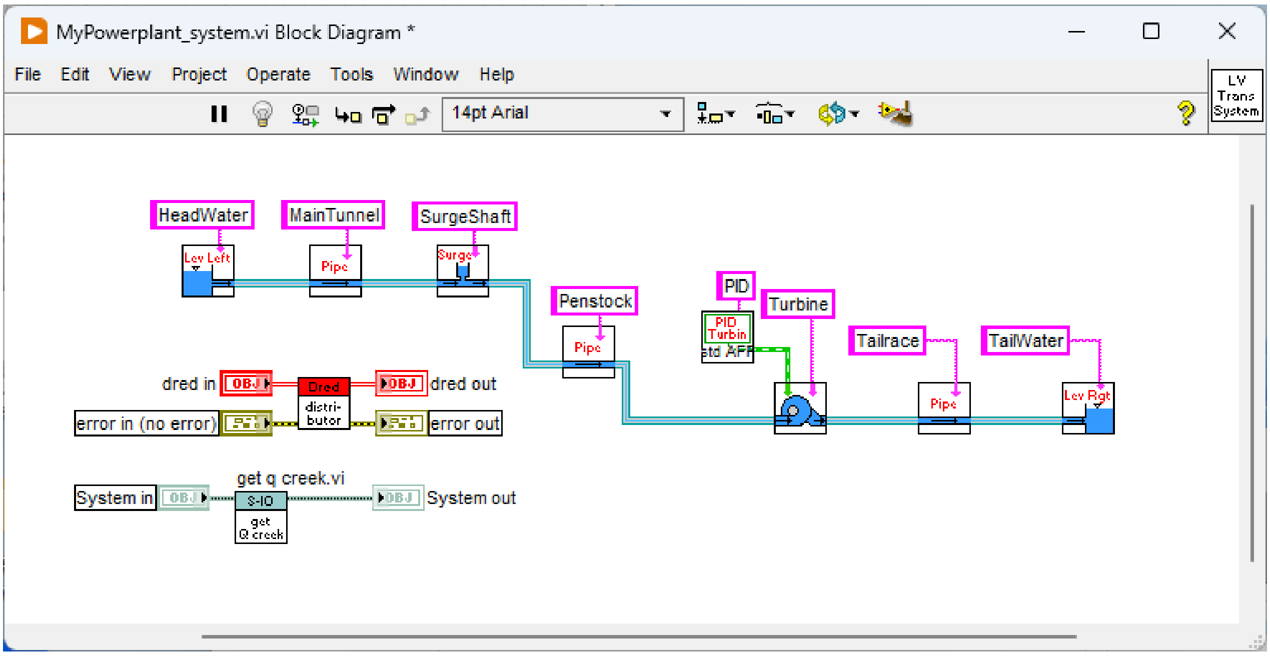


Figure 11.10 The system block diagram with a PID AFF for analysis.

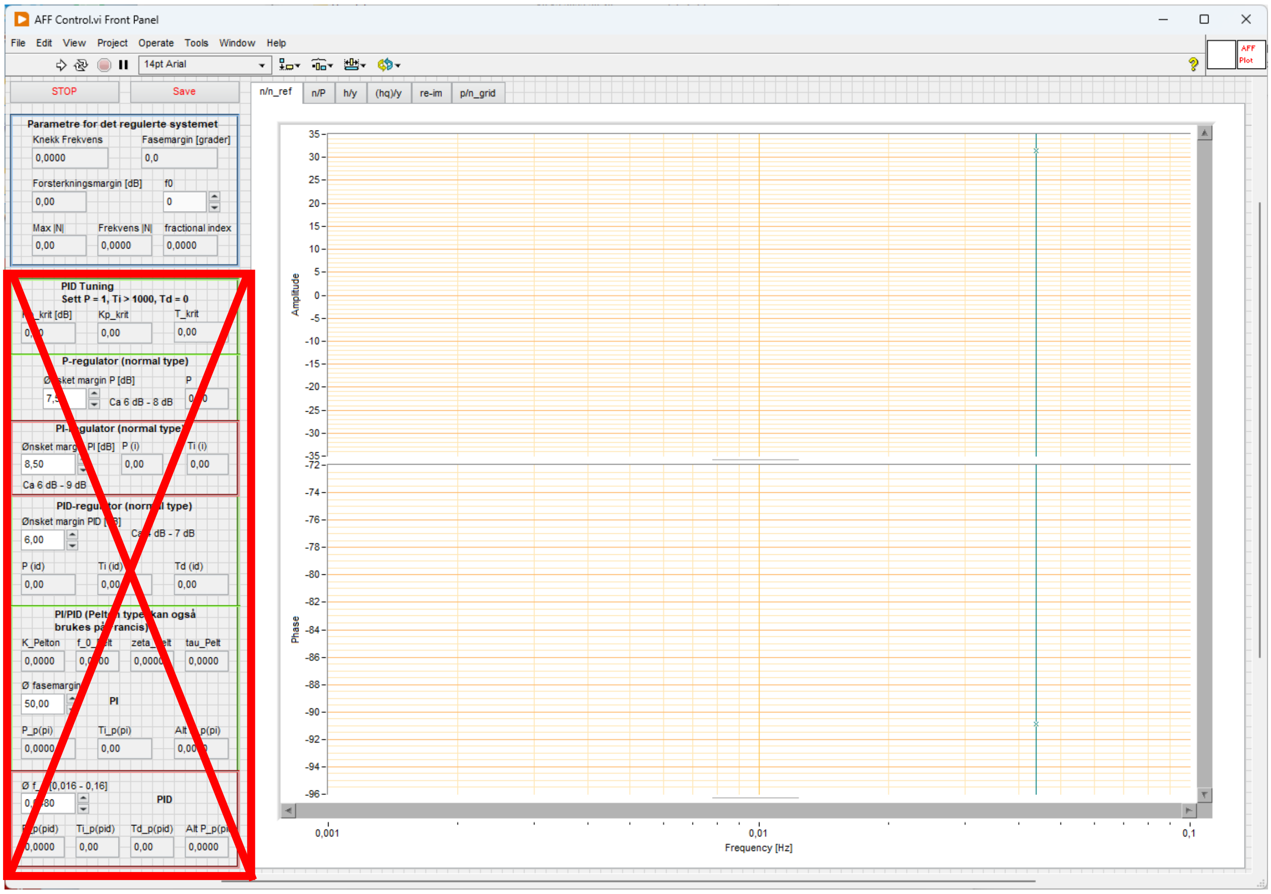


Figure 11.11 The AFF Control.vi What's inside the red X is still functional, but never used anymore. It's the old autotune routines.

The way it all works is:

- Pick a start and end frequency for analysis

- Start the analysis. LVTrans will then run a series of oscillations for each frequency. This may take some time.
- At the end of each frequency run, FFT of the data is done, and sent to the AFF Control for plotting.

This is best shown with examples.

11.5 AUTOTUNING

The autotune functions are hardly ever used because they are much too "theoretically academic" to really give any usable results in real life. ***This chapter can be disregarded completely****, but it shows some usage of the AFF control.vi. Autotuning is also interesting all by itself. Perhaps some trial-and-error autotuning is possible in the future, or a combination?*

The plant is the same Francis plant as before. Start LVTrans with a time step of 0.02 s. Then double click on the PID. Open the Graph tab. Unhook «kappa manual», set the PID in Island mode, set SP P [MW] and droop to 0.0 The set P Grid to 0.85. Now set the Kp=1 and Ti=10 if they aren't already. It should look something like Figure 11.12 Let it stabilize.

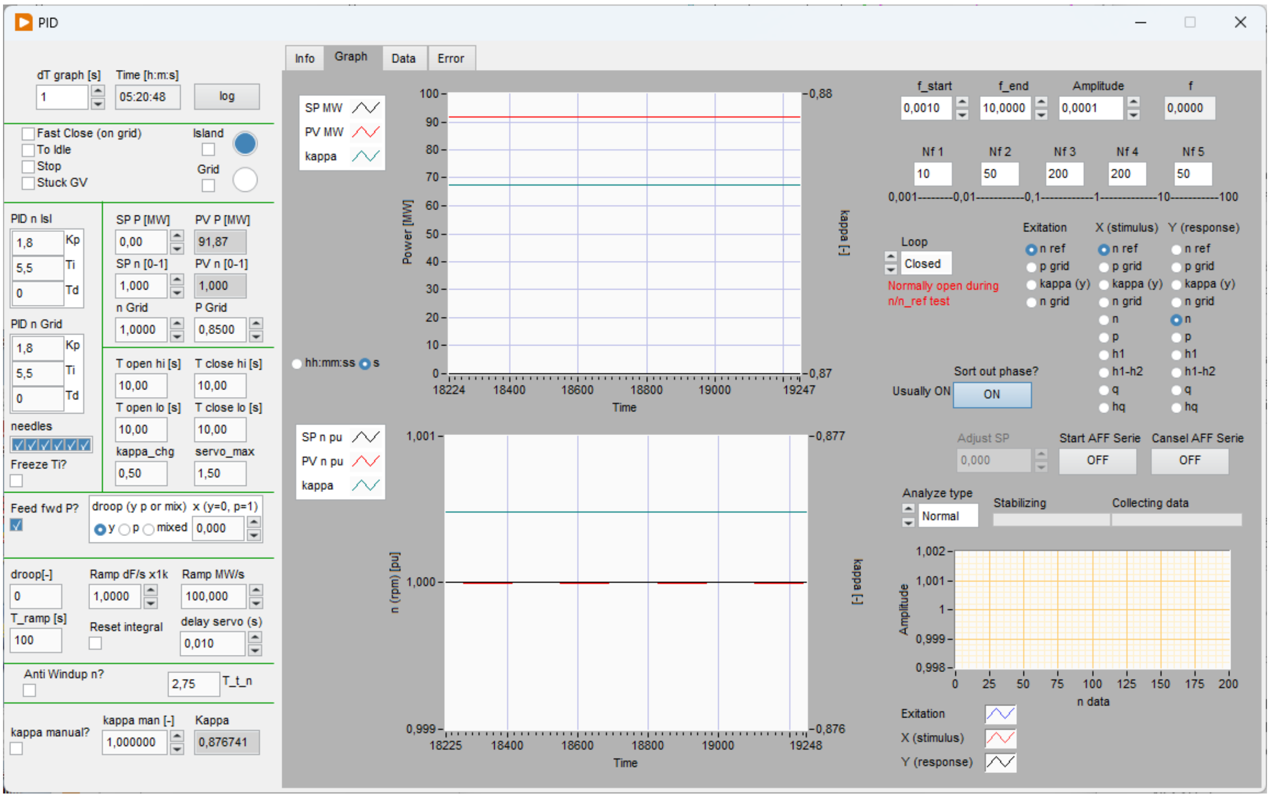


Figure 11.12 PID AFF with values set as described.

Now set the Kp=1 and Ti=10^{10} (very large number, making it a P governor) and set the "loop" to "open". Nothing much should happen, but it should look like Figure 11.13

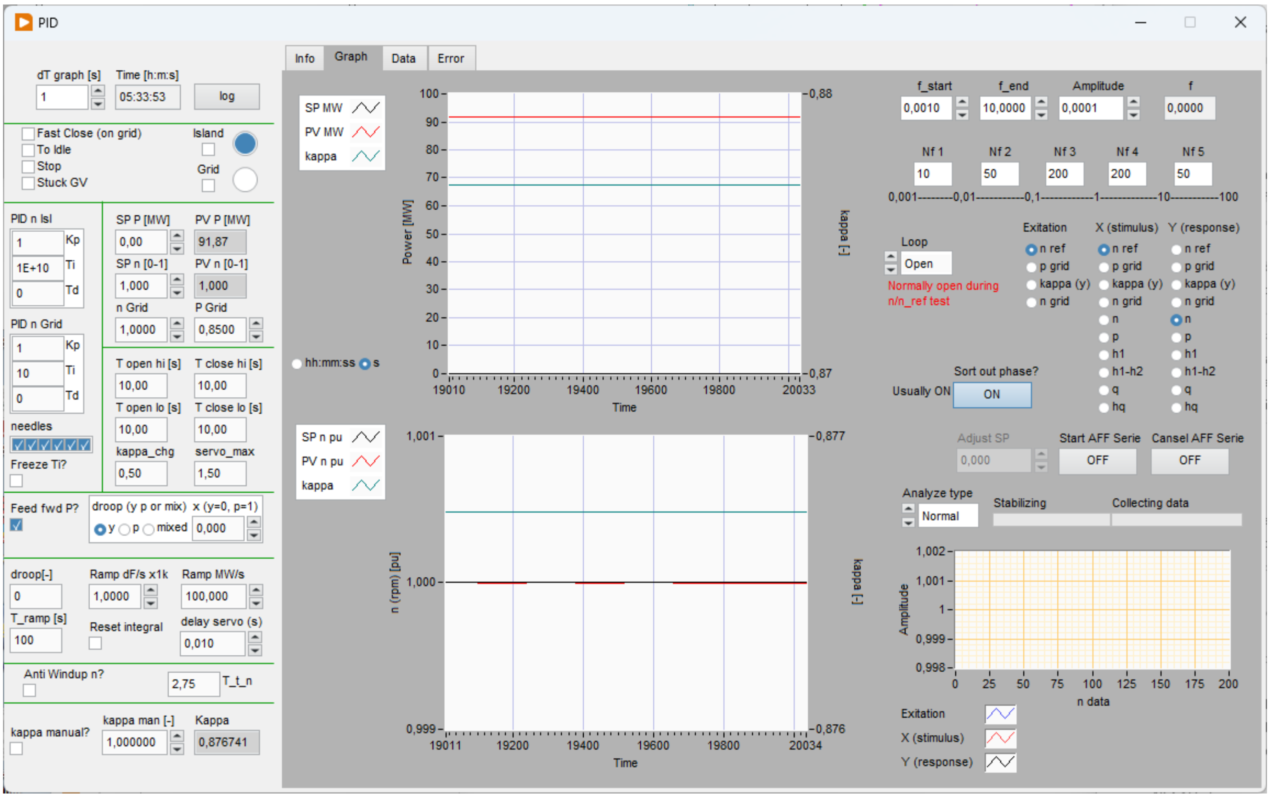


Figure 11.13 PID AFF with loop open, Kp=1 and Ti=very large

Push the button Start AFF series. The governor will start the series of frequency response calculations, frequency by frequency from f_start to f_end. This takes some time. The governor should look something like Figure 11.14.

If not already open, open AFF Control.vi and start it by pushing the white arrow. It will collect values when they arrive and plot them. This should look something like Figure 11.15.

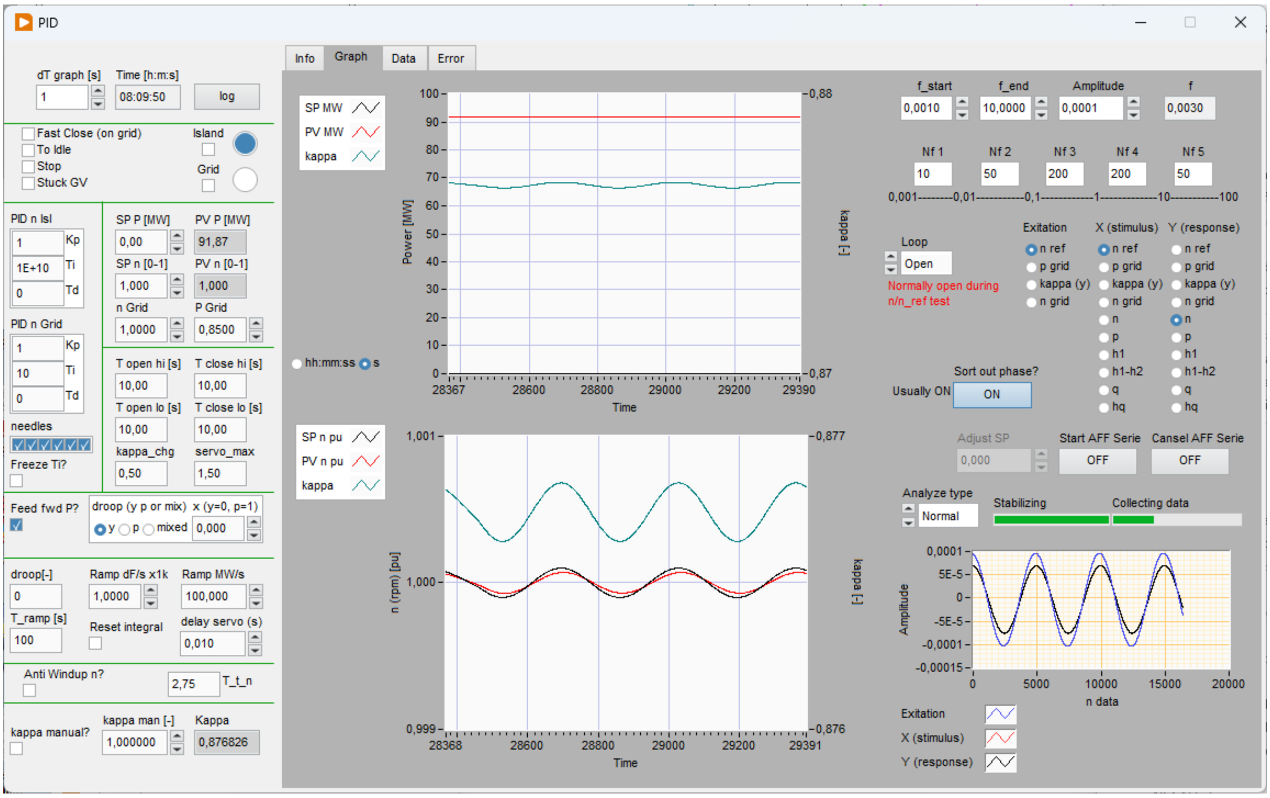


Figure 11.14 Frequency response analysis running.

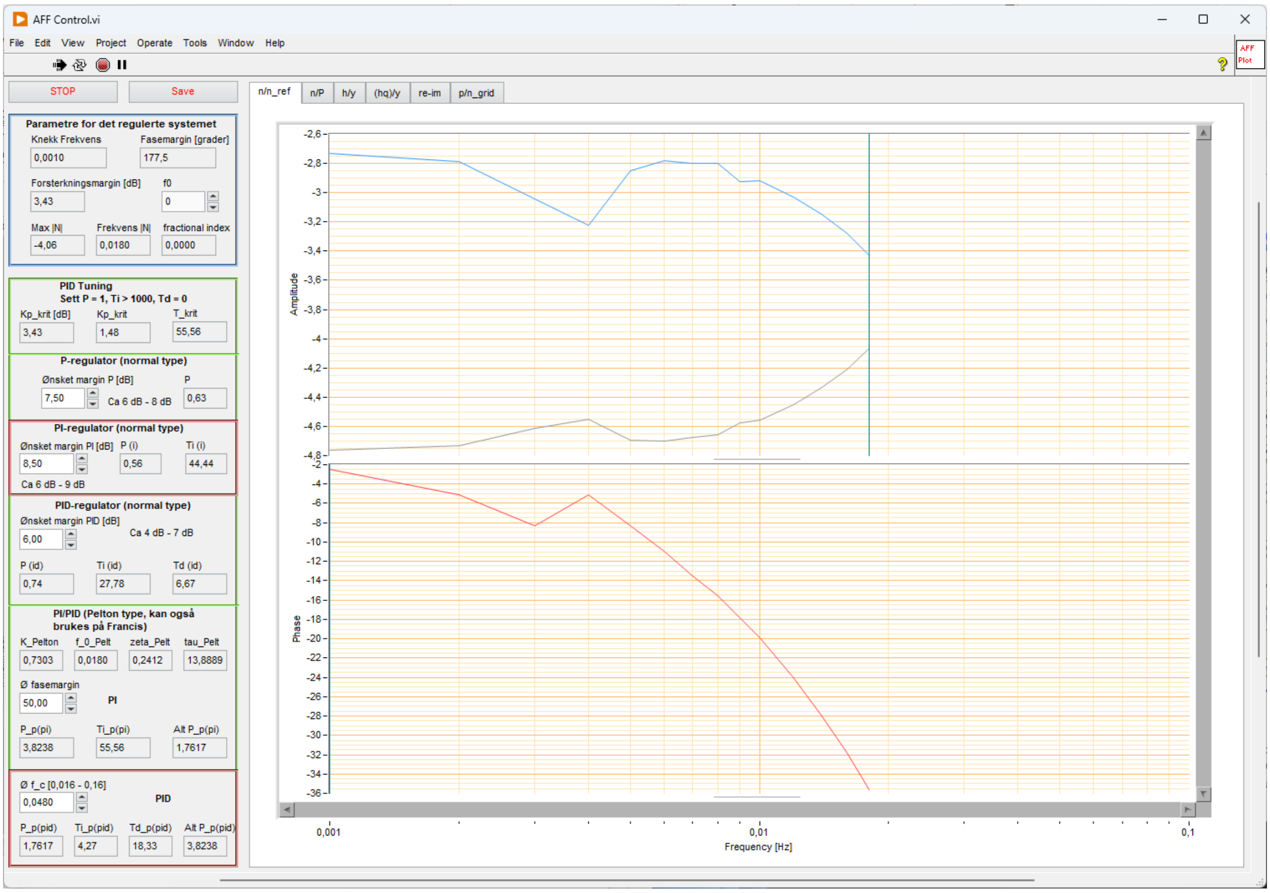


Figure 11.15 AFF Control vi collecting values when they arrive

Note! If the time step is 0.1 s, the highest frequency will be 5 Hz, and so on. To get values all the way up to 10 Hz requires a time step of 0.05 or smaller.

When finished it will look something like Figure 11.16.

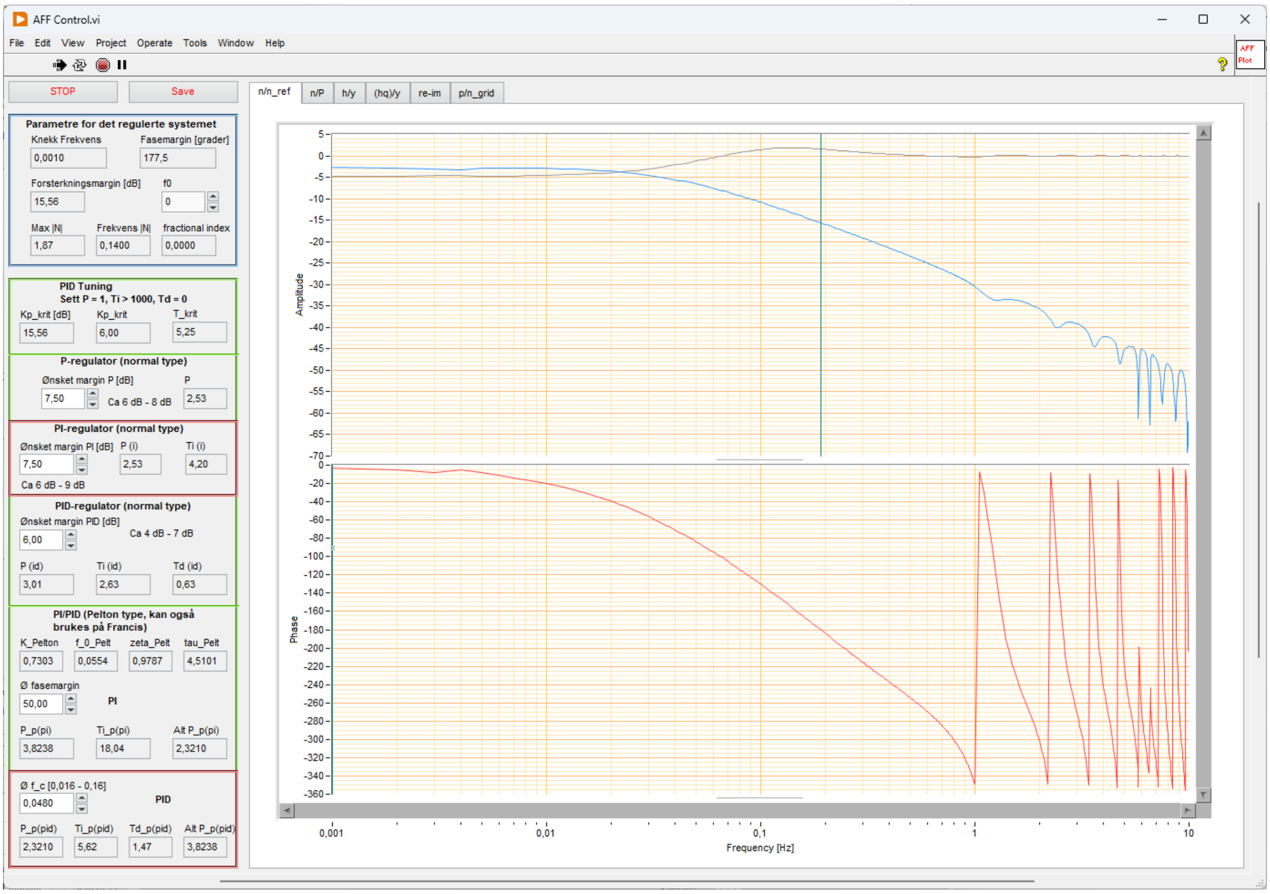


Figure 11.16 AFF Control finished collecting data

Since the frequency response is done with Kp=1, Ti=very large and Td=0 (pure P governor with gain 1.0), we can see what values are created. 5 different sets are created.

- P, PI and PID based on optimal time response (classical tuning in the s-plane).
- PI based on 1st order approximation.
- PID based on second order approximations.

For Francis turbines, only the two encircled with red will work (somewhat…). The one that most often work is PI normal type (the first red one). Here the wanted gain margin can be adjusted.

The lowest red one gives the most «optimal» parameters for a Francis. Here the cross over frequency can be adjusted, "Ø f_c" from 0.016 to 0.16 which will give a large band of possible Kp. We have two possibilities:

1. PI: Kp = 2,53 ; Ti = 4,2 ; Td = 0 (Kp = 2.1; Ti = 6.6; Td = 0)
2. PID: Kp = 2.32 ; Ti = 5,62 ; Td = 1,47 (Kp = 2.5; Ti = 6.6; Td = 0.3)

For comparison an optimum PI and PID using the trial-and-error method are shown in parentheses. What can be said here is that the autotune for PI isn't all that bad in this case. It's slightly more aggressive than trial and error. The autotune for PID is OK regarding Kp and Ti. For

Td the value is much higher than for the trial and error. This high Td may very well be theoretically OK, but in LVTrans simulations as well as in real life, this will give lots of unwanted small adjustments that will wear out the servo in no-time.

11.6 AFF ANALYSIS

A normal AFF analysis, or creating an AFF diagram is straight forward. It must however be done according to grid code. This can mean at a certain power and/or at certain value of droop and so on. With zero droop and 85% power is shown earlier. The next example is shown with 2% droop and 85% power.

The governors in LVTrans act as real governors would do, and this may not be intuitive if you are not used to joggling with droop, speed setpoint and power setpoint, especially not on isolated grid. The droop will offset the speed and so on, and so will the power setpoint. Set up the PID as in the previous example, but set the droop to 0.02 (2%) and leave the SP P [MW] at 108 MW. The PID should look something like Figure 11.17

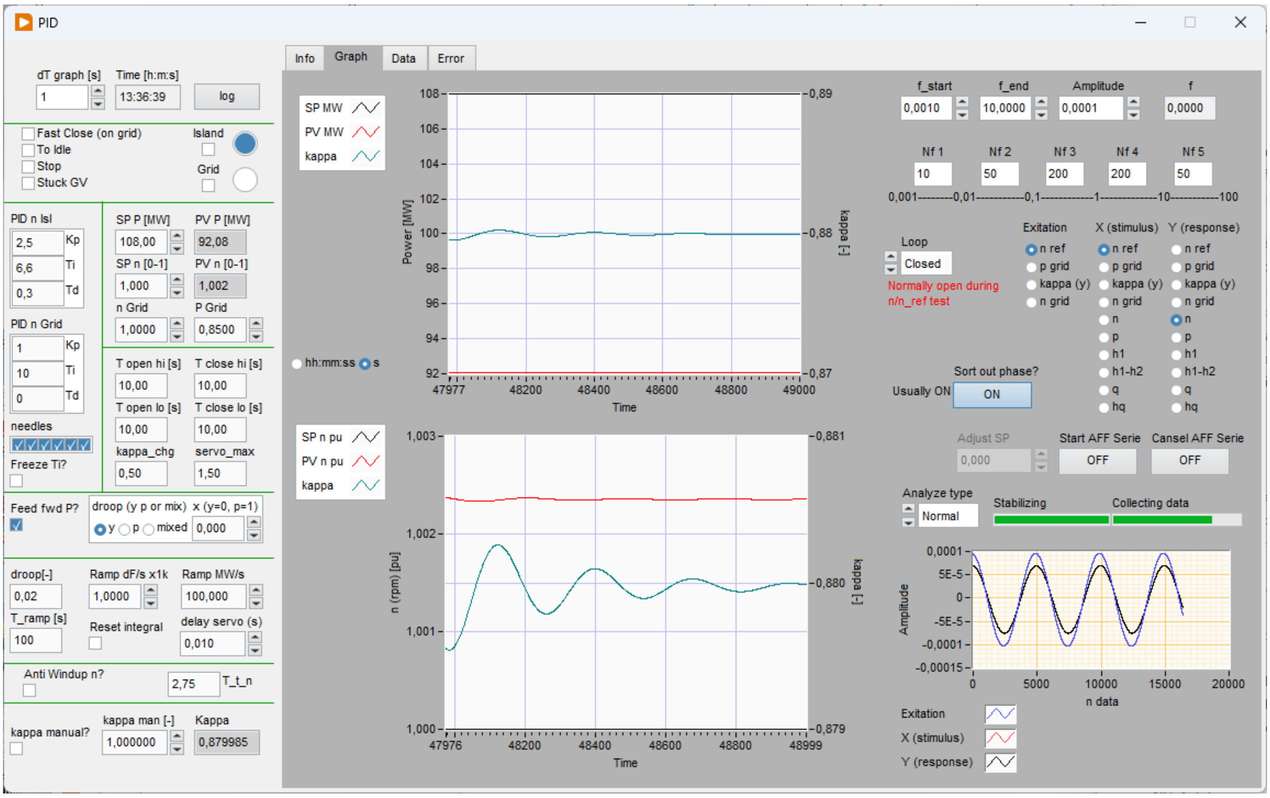


Figure 11.17 PID with P grid of 0.85, droop of 0.02 and SP P of 108 MW

Then set SP P [MW] to the same value that PV P [MW] shows. In this example it shows 92.09 MW, so we set it to 92 MW. After the initial «jolt» the PV n will settle pretty close to 1.000 rpm (in pu). Then set the «Loop» to open. This will most probably create an offset again in the speed. This offset is now adjusted by adjusting the SP P [MW] until the speed is as close to 1.000 as we can. There is no need to overdo this, getting it approximately at 1.000 is enough. The aim is to get the oscillations around 1.0 rpm (pu) instead of 0.95 for instance.

Now we do an AFF series again by clicking on the Start AFF series button.

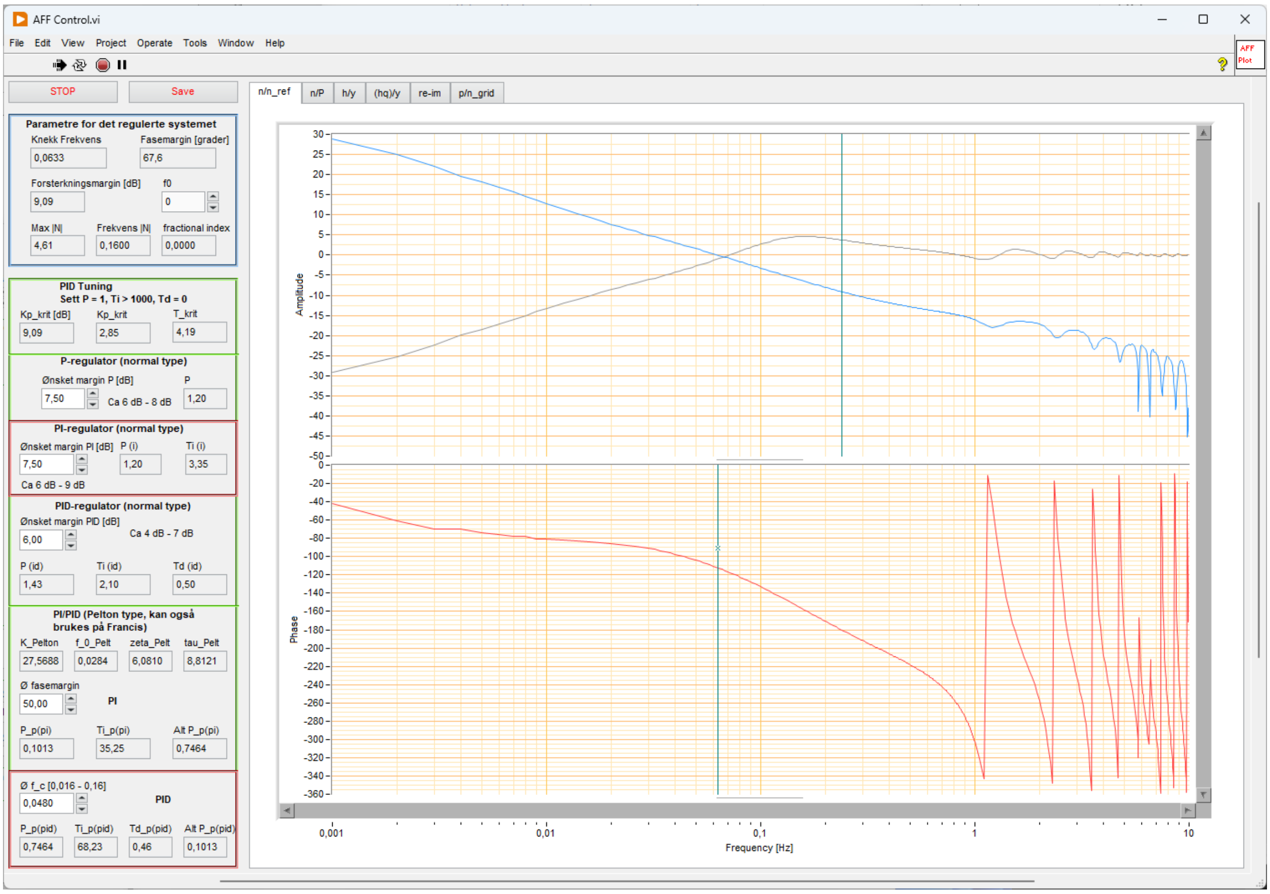


Figure 11.18 AFF plot showing phase margin, gain margin and max|N|.

The AFF plot will automaticalle find phase and gain margins as well as max|N| and crossover frequency. It also has several other plots for different tasks. These tasks are linked to the PID AFF elements.

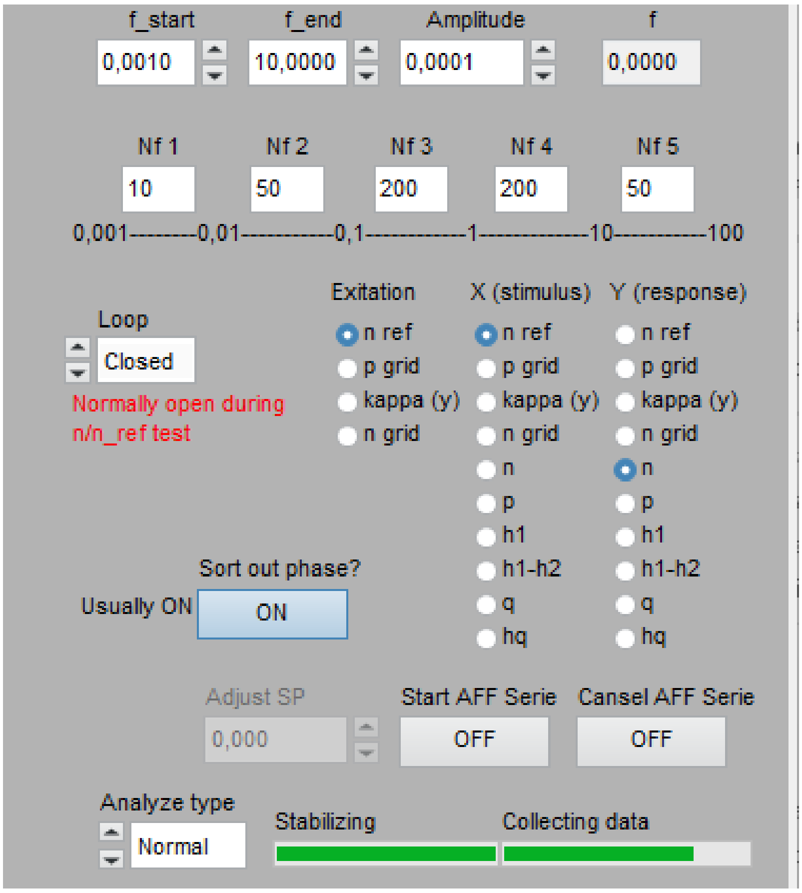


Figure 11.19 Set up of AFF plot in the PID AFF

Figure 11.19 shows the different set ups one can do in the PID AFF. f_start. f_end and amplitude should be obvious. Then there are 5 numbers that can be used to set the number of frequencies within the shown frequency bands.

This AFF is very general and can be used to plot several different frequency response plots.

Excitation is where the system is excited, normally n_ref (speed setpoint).
X (stimulus) is one point of measure, normally also n_ref.
Y (response) is the response, normally n, such that:

$$U = \frac{Y(response)}{X(stimulus)}$$

For normal governing AFF this will be:

$$U = \frac{n}{n\_ref}$$

Where the excitation also is n_ref. The Loop can be set to open and close, referring to an open and closed governing loop with feedback. Then there is “Analyze type”. There are three types, normal, fast and accurate. Fast will only take a few oscillations before the FFT is done, normal will do some more oscillations, and accurate will do even more. The reason for this is that it takes some time to do frequency response like this, and one can do fast analysis to see how it is, then an accurate to get a nice and accurate plot in a report.

The ”sort out phase” button is to change between the phase going from 0 to 360 or -180 to +180. This can be useful for some frequency response analysis, because some are naturally from 0-360 while others are naturally from -180 to + 180. The graphs looks much better choosing the correct one. This button must be pushed before starting the analysis.

Data can also be saved in ordinary text files by pushing the log button (not shown in Figure 11.19).

12 ELEMENTS

The elements are found in the menu, see Figure 4.1. The data for each element will reside in separate data files in the folder named “Data”. The files are Windows ini files, and therefore very simple. They can be edited in the element themselves or can be edited using a text editor.

All elements have a defined positive direction of flow. Positive direction is from left to right, or whatever is indicated on the element icon.

H start and Q start are the starting values, unless otherwise specified. This means the program starts with these values and iterates towards steady state within seconds. H start is HGL (Hydraulic Grade Line) unless otherwise specified. HGL in a pipe is the absolute pressure plus the geodesic height, or level. Unless used to it, it can be a bit confusing, but it’s important to differentiate between HGL and absolute pressure and use them in the correct spot. Normally the program will just chew through badly offset starting values and converge fast to a steady state, but when using open channels, it is very important to start with good values. As a rule of thumb:

- HGL everywhere upstream a turbine is set to head water level.
- HGL downstream a turbine is set to tail water level.

For instance: Head water level is 1000m. The turbine is at 400m and tail water level is at 430 m, all AMSL (datum at sea level). H start upstream turbine is then 1000m, and downstream tube is 400m. However, as mentioned, this is seldom of any importance unless an open surface channel is included somewhere.

The data files are created at first run with **default data**. Some of the data may be way off, and some should not be changed. In the following the data are marked as:

- Bold font : Data must normally be set/changed to fit with the real system
- Italic font : Default data should normally NOT be changed.
- Normal font : Default data is OK in most circumstances.

Some of the variations of elements have no relevance in standard analysis, and they are not described here.

Note! a description of the data is also found in each element by clicking the “Info” tab, it could be slightly outdated, however.

Note! Each element starts with three lines. The first line is required for the Windows ini system and should not be changed. The second line is a typedef version number and should also not be changed. The third line is the name LVTrans gives to the window for the element. This can be changed, but it has no effect on the calculations. These three lines, except the version number, are not included in the explanation below. In addition there is an AUX-IO data for embedded devices and outside the scope of this manual. The AUX-IO is not included below, but MUST be set to FALSE for normal analysis.

12.1 PIPE

Pipe is the element for pipes and tunnels (closed conduit). All pipe elements have to have a non-pipe element at each end.

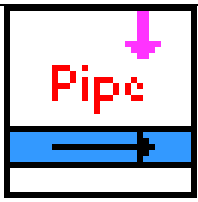
Name	Used for	Icon	Special
Pipe	Pipe, tunnel or any other closed conduit.		Have to have non-pipe elements on each side. Defines geodesic heights in the system.

Table 12.1 Data for Pipe version 00002

Data	Format	Description
L [m]	**value**	**Length of pipe**
Use Diameter?	**TRUE/FALSE**	**TRUE = use pipe diameter instead of cross sectional area**
D [m]	**value**	**Diameter (or equivalent diameter)**
Area [m2]	**value**	**Cross sectional area of the pipe**
Use Dh?	TRUE/FALSE	Use hydraulic diameter
Periphery [m]	value	The periphery of the pipe (when using hydraulic diameter)
f	**value**	**Friction constant as used in the Moody diagram (constant for each time step).**
full_moody?	TRUE/FALSE	Calculate new friction constant based on the Moody diagram for each time step. (more accurate than f above)
Epsilon [m]	value	The roughness in m when using full_moody
ny	*value*	*Kinematic viscosity, v (1e-6 for water at 20 deg C) when using full_moody*
f_max	*value*	*Max value f can take when using full_moody.*
Lambda	*value*	*Dynamic friction constant*
Rho [kg/m3]	value	Density of the fluid
a [m/s]	value	Celerity (speed of sound) in the pipe
Z left [m from datum]	**value**	**Geodesic height on the "left" end**
Z right [m from datum]	**value**	**Geodesic height on the "right" end**
Q start [m3/s]	value	Starting value for calculations
H start [m3/s]	value	Starting value for calculations, EGL from datum

12.2 CONSTANT LEVEL LEFT

Contant Level Left defines the water level (constant level) for a reservoir on the "left" side.

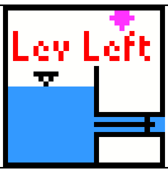
Name	Used for	Icon	Special
Constant Level Left	Defines a constant pressure on the "left" side.		A dam, reservoir and so on.

Table 12.2 Data for const level left, version 00001

Data	Format	Description
H0 [level m]	**value**	**Nominal geodesic level of the reservoir from datum. Can be changed interactively during running.**
Cvp	value	Loss coefficient with positive flow relative to element.
Cvm	value	Loss coefficient with negative flow relative to element.

12.3 CONSTANT LEVEL RIGHT

Constant Level Right defines the water level (constant level) for a reservoir on the "right" side.

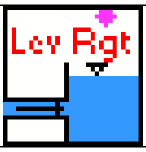
Name	Used for	Icon	Special
Constant Level Left	Defines a constant pressure on the "left" side.		A dam, reservoir and so on.

Table 12.3 Data for const level right, version 00001

Data	Format	Description
H0 [level m]	**value**	**Nominal geodesic level of the reservoir from datum. Can be changed interactively during running.**
Cvp	value	Loss coefficient with positive flow relative to element.
Cvm	value	Loss coefficient with negative flow relative to element.

12.4 SIMPLE CONNECTION

Simple Connection is used to:

- Connect two pipes with different specification
- Define singular losses
- Divide the pipe/tunnel better to fit the terrain for instance.

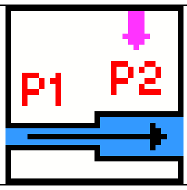
Name	Used for	Icon	Special
Simple Connection	Connect pipes, define singular losses.		

Table 12.4 Data for misc connection version 00001

Data	Format	Description
Cvp	value	Loss coefficient with positive flow relative to element
Cvm	value	Loss coefficient with negative flow relative to element.

12.5 SCREEN

The Screen element is identical to the Simple Connection element. The only difference is the "look" making it easier to identify on the system block diagram.

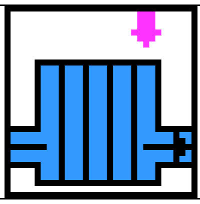
Name	Used for	Icon	Special
Screen	Connect pipes, define singular losses.		Identical to Simple Connection except the look.

Table 12.5 Data for Screen version 00001

Data	Format	Description
Cvp	value	Loss coefficient with positive flow relative to element
Cvm	value	Loss coefficient with negative flow relative to element.

12.6 PID TURBINE

PID Turbine is several different implementations of a turbine governor (select type). The ones used for analysis are usually the "standard" and the "linear". The "standard" implementation is after IEEE and includes both power related droop and opening related droop, as well as a mix. They all use the same data files. PID Turbine have tons of interactive parameters. pu units (per unit) are used throughout except where noted.

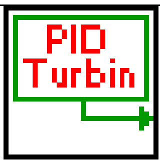
Name	Used for	Icon	Special
PID Turbine	To control and plant modl and the simulations.		Must be used with a Francis or Pelton turbine. Lots of interactive parameters.

Table 12.6 Data for turbine version 00001

Data	Format	Description
Pr [MW]	**value**	**"Rated" power. Must correspond to rated/BEP power of the turbine at given torque and rpm.**
Nr [rpm]	**value**	**"Rated" rpm. Must correspond to rated/BEP rpm of the turbine.**
SP [MW]	**value**	**Initial Set Point of the power**
Kp [PID grid]	value	Proportional constant when run on the common grid
Ti [PID grid]	value	Integral time constant (s) when run on the common grid
Td [PID grid]	value	Derivate time constant (s) when run on the common grid.
Kp [PID island]	value	Proportional constant when run on islandic grid
Ti [PID island]	value	Integral time constant (s) when run on islandic grid
Td [PID island]	value	Derivate time constant (s) when run islandic grid
Tip [PID power]	*value*	*Integral time constant for a special power feedback function (non-standard, not used for "normal" PID type)*
T ramp [s]	value	Time constant for the ramp power setpoint
Rp [droop]	value	Droop (in pu : 0,01 = 1%)
T close high [s]	value	Extrapolated closing time over the changeover point
T close low [s]	value	Extrapolated closing time below the changeover point
T open high [s]	value	Extrapolated opening time over the changeover point
T open low [s]	value	Extrapolated opening time below the changeover point
Kappa change point	value	The point in kappa (pu) wher the closure/opening go from high to low as defined above
a [cnst]	*value*	*Constant for estimating Y/P*[19]
b [cnst]	*value*	*Constant for estimating Y/P*
c [cnst]	*value*	*Constant for estimating Y/P*
servo_max	*value*	*The max opening of the servo in pu.*
manual_default	value	The PID starts per default always in manual. The opening is set here.
power droop?	TRUE/FALSE	Default startup in power elated droop or opening related droop

[19] For better estimating of servo vs power (Y/P) when opening related droop is used and for the power ramp, a second order best fit equation is used where $Y/P = ax^2 + bx + c$ Default is a=c=0 and b=1.

12.7 PID PUMP

PID Pump is an implementation of a standard PID governor for a pump. It can govern based on rpm, pressure, torque, flow and mechanical and hydraulic power. This is done interactively. It also has a check valve that closes for backward flow. The check valve can be removed interactively. It can be run on a set rpm, set free (run away) and run normal with the governor.

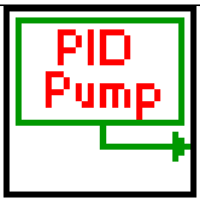
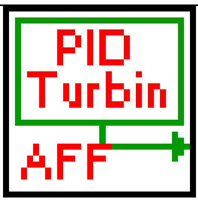
Name	Used for	Icon	Special
PID Pump	Defines a PID governor for a pump.		Must be used with a pump. Lots of parameters and modes can be adjusted interactively. .

Table 12.7 Data for PID pump

Data	Format	Description
Nr [rpm]	**value**	**Rated rpm**
Qr [m3/s]	**value**	**Rated flow**
Hr [m]	**value**	**Rated pressure head**
Tr [Nm]	**value**	**Rated torque**
Kp [PID]	value	Proporsjonal constant
Ti [PID]	value	Integral time constant (s)
Td [PID]	value	Derivate time constant (s)
CheckValve?	[TRUE/FALSE]	Check valve included at startup

12.8 PID TURBINE AFF

PID Turbine AFF is several versions of PID Turbine used for AFF plots, or frequency response analysis. This is explained earlier in the manual. The parameters are identical to ordinary PID Turbines. It's found with "select type". **It is used together with AFF Control.vi in the main folder.**

Name	Used for	Icon	Special
PID Turbine AFF	Stability analysis, AFF plots.		Must be used with Francis or Pelton. Tons of interactive variables. Used together with AFF Control.

12.9 SURGE SHAFT STANDARD

Surge Shaft Standard is a constant area surge shaft with friction and losses. It can be used when the cross sectional area does not change with level.

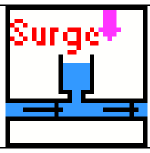
Name	Used for	Icon	Special
Surge Shaft Standard	Simpler surge shafts with constant cross sectional area.		Have to stand between two pipes.

Table 12.8 Surge Shaft Standard, version 00002

Data	Format	Description
Use Area?	**TRUE/FALSE**	**Shall the area be used, or the diameter?**
Area [m2]	**value**	**Cross sectional area of the surge shaft**
D [m]	**value**	**Equivalent diameter. Used if Use Area is FALSE**
f	value	Friction constant for the shaft according to Moody diagram
Cvp	value	Loss coefficient for the flow entering up into the shaft
Cvm	value	Loss coefficient for the flow exiting down from the shaft
H start [m from datum]	value	Start value for the level in the shaft.

12.10 SURGE SHAFT VARIABLE

A surge shaft with variable cross sectional area, and with a weir at the top.

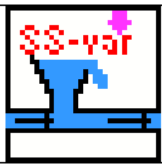
Name	Used for	Icon	Special
Surge Shaft Variable	Surge shaft with variable cross sectional area and/or a weir at the top.		Have to stand between two pipes.

Table 12.9 Surge Shaft Variable, version 00001

Data	Format	Description
Cvp	value	Loss coefficient for the flow entering up into the shaft
Cvm	value	Loss coefficient for the flow exiting down from the shaft
H start [m from datum]	**value**	**Start value for the level in the shaft.**
B [m width of weir]	value	*Width of the weir at the top*
C-cnst	value	*Coefficient for the weir*
H weir [m from floor]	**value**	***The level of the weir. Measured from the (tunnel) floor of the bottom of the shaft***
L0	**[m]**	**Level of lowest profile, measured from the (tunnel) floor.**
…		
L9	**[m]**	**Level of highest profile, measured from the (tunnel) floor.**
A0	**[m2]**	**Areal at the lowest profile point**
…		
A9	**[m2]**	**Areal at the highest profile point**

The cross sectional area profile is described with 10 pairs of Lx:Ax. Between these pairs a linear interpolation is done. All the pairs have to be used. If the level is above L9, then A9 is used for all higher levels. If the level is below L0, then A0 is used for all lower levels. This way one can simulate also if the shaft is fully emptied and the level start to reach into the tunnel.

L0 til L9 are levels **relative** to the floor of the (tunnel) beneith the shaft. This geodesic level are usually described in most civil drawings. The Z left and Z right in connecting pipes define where this level is in absolute terms. Start with L0 = 0 for instance. As an example : L0 = 0, L1 = 5, L2 = 20, A0 = 50, A1 = 10 A2 = 10. The area at the bottom of the shaft is 50 m2. From there, linear interpolation is done to L1 where the are is 10 m2. From L1 to L2 the area is constant 10 m2.

All L og A values must be used. If the shaft is described well enough with 3 pairs, then simply continue upward. In the example above insert L3 = 21, A3 = 10, L4 = 22 A4 = 10 and so on. No L values can have identical values. They can however be very close.

The weir is always calculated. Enter the correct H weir and the width. If no weir exists, then be sure to enter a H weir above any anticipating surge level.

12.11 SURGE SHAFT VARIABLE QIN

A surge shaft with variable cross-sectional area, a weir at the top, and with Qin possibilities, making it a kind of more elaborate creek shaft.

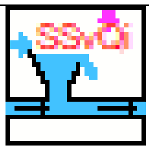
Name	Used for	Icon	Special
Surge Shaft Variable Qin	Surge shaft with variable cross sectional area and/or a weir at the top and a Q in.		Have to stand between two pipes. Works also as a more elaborate creek shaft.

Table 12.10 Surge Shaft Variable, version 00001

Data	Format	Description
Cvp	value	Loss coefficient for the flow entering up into the shaft
Cvm	value	Loss coefficient for the flow exiting down from the shaft
H start [m from datum]	**value**	**Start value for the level in the shaft.**
B [m width of weir]	value	*Width of the weir at the top*
C-cnst	value	*Coefficient for the weir*
H weir [m from floor]	**value**	***The level of the weir. Measured from the (tunnel) floor of the bottom of the shaft***
L0	**[m]**	**Level of lowest profile, measured from the (tunnel) floor.**
…		
L9	**[m]**	**Level of highest profile, measured from the (tunnel) floor.**
A0	**[m2]**	**Areal at the lowest profile point**
…		
A9	**[m2]**	**Areal at the highest profile point**
X*Qinn	**value**	**The fraction X of the flow set in the main program for all creek shafts in the system.**
Individual Q?	[TRUE/FALSE]	If true, then X*Qin is NOT used. Instead, the Q is set in the element itself when running

See chapter 12.10 and chapter 12.12 for explanation. This is essentially only a normal Surge Shaft Variable with creek shaft capabilities. Flow into it can be specified.

12.12 CREEK SHAFT NORMAL

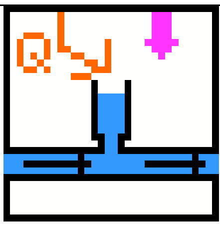
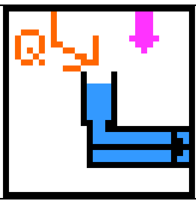
Name	**Used for**	**Icon**	**Special**
Creek shaft normal	Creek shafts/brook intakes.		Must be set between two pipes. Flow into the element is positive. Default the flow is a set fraction of the flow defined in the main program.

Table 12.11 Creek Shaft Normal, version 00003

Data	**Format**	**Description**
Use Area?	**TRUE/FALSE**	**Shall the area be used, or the diameter?**
Area [m2]	**value**	**Cross sectional area of the surge shaft**
D [m]	**value**	**Equivalent diameter. Used if Use Area is FALSE**
f	value	Friction constant for the shaft according to Moody diagram
X*Qinn	**value**	**The fraction X of the flow set in the main program for all creek shafts in the system.**
Cvp	value	Loss coefficient for the flow entering up into the shaft
Cvm	value	Loss coefficient for the flow exiting down from the shaft
H start [m from datum]	value	Start value for the level in the shaft.
Individual Q?	[TRUE/FALSE]	If true, then X*Qin is NOT used. Instead the Q is set in the element itself when running

The reason behind X*Qin is that the flow in creek shafts is very seldom measured. The value is given as a prediction from perspiration, melting of snow, drainage and so on within the area the creek shafts of the power plant collects water. X*Qin can be any value, also much larger than 1.

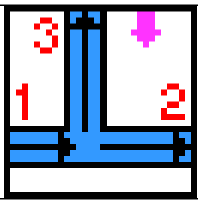
12.13 CREEK SHAFT NORMAL Q1

Name	Used for	Icon	Special
Creek shaft normal Q1	Creek shaft/brook instakes connected at the start of a pipe.		Is connected at the "left" end of a pipe.

The Creek Shaft Normal Q1 has identical data file to Creek shaft normal. See Table 12.11

12.14 T 1IN 2OUT N AND T 1IN 2OUT S

T 1in 2out N and T 1in 2out S are diverging branches with one inlet and 2 outlets. The difference between N and S is simply the way the third branch exits in the block diagram. In all other aspects they are identical. The reason for S and N is to give flexibility in the system block diagram.

Name	Used for	Icon	Special
T 1in 2out N	Diverging branch type N.		Have to stand between 3 pipe elements.

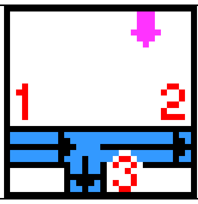
Name	Used for	Icon	Special
T 1in 2out S	Diverging branch type S.		Have to stand between 3 pipe elements.

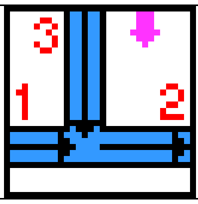
Table 12.12 T 1in 2out, version 00001

Data	Format	Description
Cv1	value	Loss coefficient from 1 to the middle
Cv2	value	Loss coefficient from the middle to 2
Cv3	value	Loss coefficient from the middle to 3

Note! The element has the same loss coefficient for positive and negative flow, and these coefficients are constant. This is therefore a simplistic way of loss calculations, but it will cover most practical situations well enough.

12.15 T 2IN 1OUT N AND T 2IN 1OUT S

T 2in 1out N and T 2in 1out S are converging branches with 2 inlets and 1 outlet. They are the converging variants from the previous chapter.

Name	Used for	Icon	Special
T 2in 1out N	Converging branch type N.		Have to stand between 3 pipe elements.

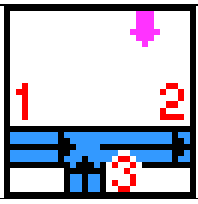
Name	Used for	Icon	Special
T 2in 1out S	Converging branch type S.		Have to stand between 3 pipe elements.

Table 12.13 T 2in 1out, version 00001

Data	Format	Description
Cv1	value	Loss coefficient from 1 to the middle
Cv2	value	Loss coefficient from the middle to 2
Cv3	value	Loss coefficient from 3 to the middle

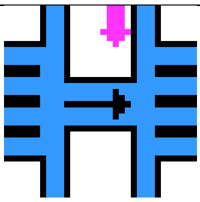
Note! The element has the same loss coefficient for positive and negative flow, and these coefficients are constant. This is therefore a simplistic way of loss calculations, but it will cover most practical situations well enough.

12.16 T ADAPTIVE

T Adaptive defines an adaptive converging/diverging T. It can have 0 to 6 inlets pipes connected and 0 to 6 outlet pipes connected. It does not matter which one of the inlet/outlets are used. This T does not have any associated loss coefficients.

Since it can have 0 to 6 connections, connecting it at the end/start of one pipe makes this element act as a dead end closed pipe.

Table 12.14 T Adaptive, version 00001

Name	Used for	Icon	Special
T Adaptive	Converging and diverging T, very flexible.		Connects to 0-6 pipes in each end. No associated loss coefficients.

Other than default typedef etc, this element has no data to configure.

12.17 FRANCIS

The Francis element represents a Francis turbine. Using a special excel file to find the data is explained earlier in the manual.

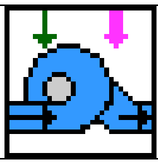
Name	Used for	Icon	Special
Francis	Defines the Francis turbine.		Has to be used together with PID turbine or PID turbine AFF.

Table 12.15 Data for the Francis element, version 00001

Data	Format	Description
Q BEP [m3/s]	**value**	**Q BEP or Q rated**
H BEP [m]	**value**	**H BEP or H rated**
H BEP design [m]	**value**	**H BEP/rated if the turbine is run a bit "off design" (if for instance H BEP is different from the net head of the plant)**
N rated [rpm]	**value**	**Rated rpm**
T BEP [Nm]	**value**	**Torque at Q, H BEP or rated**
E BEP [Nm]	**value**	**Electrical torque at Q, H BEP or rated (set it equal to T BEP)**
a1 BEP [deg]	**value**	**Design inlet angle, α1, at BEP or rated**
b1 BEP [deg]	**value**	**Design outlet angle, β1, at BEP or rated**
r1 [m]	**value**	**Radius of turbine wheel at inlet**
r2 [m]	**value**	**Radius of turbine wheel at outlet**
Ta [s]	**value**	**Time constant, Ta, for the rotating masses (turbine + generator)**
Twt turbine [s]	*value*	*Inertia water time constant for the water inside the turbine element.*
Rq	*value*	*Design constant*
Rm	*value*	*Design constant*
Rd	*value*	*Design constant*
eta hydraulic BEP	value	Hydraulic efficiency
eta total BEP	value	Total efficiency
N turbines	*value*	*Number of turbines. Usually this is 1. It can however also be set to a higher value, for simplified simulation of several turbines at once.*
Pole pairs	**value**	**The number of pole pairs in the generator**
D grid	*value*	*Damping coefficient on the grid/generator*
Delta r [rad]	*value*	*The rated pole angle in electrical degrees*
F grid	**value**	Grid frequency, 50 or 60 Hz
eve_mod	*value*	*0,2 gives "better" runaway speed for high head Francis*

Important notes:

- The data used must be consistent. Given a Q BEP and H BEP, then this gives a set of other data that must be used together. This must also be consistent with P BEP for the governor element.
- If Q rated is used instead of Q BEP, then all other values must also be rated. This includes closing and opening times of the turbine servo.
- Eta total BEP should be slightly smaller than eta hydraulic BEP (by 0.01).

12.18 PELTON

The Pelton turbine in LVTrans.

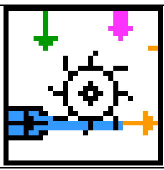
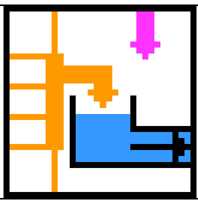
Name	Used for	Icon	Special
Pelton	Defines a Pelton type turbine.		Must be used together with PID Turbine or PID Turbine AFF. Can also be used with Pelton Sump.

Table 12.16 Pelton data, version 00001

Data	Format	Description
Q BEP [m3/s]	**value**	**Q BEP or Q rated**
H BEP [m]	**value**	**H BEP or H rated**
N rated [rpm]	**value**	**Rated rpm**
T BEP [Nm]	**value**	**Torque at Q, H BEP or rated**
E BEP [Nm]	**value**	**Electrical torque at Q, H BEP or rated (set it equal to T BEP)**
Ta [s]	**value**	**Time constant, Ta, for the rotating masses (turbine + generator)**
Rm	*value*	*Design constant*
eta_bep	value	Hydraulic efficiency
N turbines	*value*	*Number of turbines. Usually this is 1. It can however also be set to a higher value, for simplified simulation of several turbines at once.*
Pole pairs	**value**	**The number of pole pairs in the generator**
D grid	*value*	*Damping coefficient on the grid/generator*
Delta r [rad]	*value*	*The rated pole angle in electrical degrees*
F grid	value	Grid frequency, 50 or 60 Hz
n needles	value	The number of injectors for the turbine
eta_3	*value*	*Design constant*
q_max	*value*	*Design constant*

12.19 PELTON SUMP

Pelton Sump is the "sump" or reservoir often placed after one or several Pelton turbines to collect the water. For most calculations, it is unnecessary to include pipes and tunnels after a Pelton, since the water conduit is broken at this point. For those circumstances where calculation of the tail race from a Pelton is warranted after all, then this element must be used.

Name	Used for	Icon	Special
Pelton Sump	Defines the reservoir behind one or several Pelton turbines.		Must be used with Pelton element. From 1 to 6 individual turbines can be connected.

Data	Format	Description
Use Area?	**TRUE/FALSE**	**Use area or diameter to set the surface area.**
Area [m2]	**value**	**Area of the surface.**
D [m]	value	Equivalent diameter of the surface area.
f	value	Friction coefficient, f, of the reservoir.
Cvp	value	Loss coefficient for water flowing up into the reservoir
Cvm	value	Loss coefficient for water flowing out of the reservoir
H start [m from datum]	value	Start value of the level when calculations start

12.20 PUMP CENTRIFUGAL

A pump (centrifugal) using suter curves according to Wylie and Streeter.

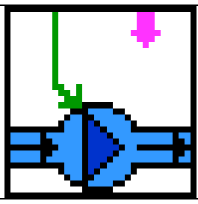
Name	Used for	Icon	Special
Pump Centrifugal	Sumulating a centrifugal pump described with Suter curves.		Have to be used with PID Pump. Needs Suter curves.

Table 12.17 Pump Centrifugal, version 00001

Data	Format	Description
Q rated [m3/s]	**value**	**Q rated for the pump**
H rated [m]	**value**	**H rated for the pump**
N rated [rpm]	**value**	**Rated rpm for the pump**
T rated [Nm]	**value**	**Rated mechanical torque**
P rated [MW]	**value**	**Rated power**
I0 [inertia]	**value**	**Polar (rotational) moment of inertia**
N pumps	**value**	The number of pumps (usually only one per element)
Suter-file.txt	**string**	**The name of the file with the suter curves.**

Suter-curves is a method to transform torque and H-Q diagrams to a format suitable for simulations. Two examples are in the folder named useful\suter. These can be copied to the Data folder for the system. Exactly what Suter curves are and how to make them is best described in Wylie and Streeter.

Note! If LVTrans does not find a Suter file, it will use a default Suter curve based on data from Wylie and Streeter (this is the same data as in the "useful\suter" folder). **This may not at all be what you want however**, since this curve most probably will not describe the pump you want to model. This may still useful, because in many cases you are not interested in the pump transients, but simply want one or more pumps in the system to supply flow and pressure in a "pump like" fashion. This can be done because the Suter curves themselves are non-dimensional, and the pump is scaled with the rated values in the data file.

The file containing the Suter curves in LVTrans must be formatted in a special fashion as shown in Figure 12.1 (this is the default data, and the curves in the "useful\suter" folder). There are five columns. Column 1 is the "X" data. It runs from zero to 2*π, representing a full circle, all 4 quadrants. Columns two and three are the corresponding WH and WB representing head and torque in "Suter format". Columns four and five are the same but smoothed more.

LVTrans only reads column 1, 4 and 5, that is X, WH(ave5) and WB(ave5)[20]. When making a Suter file from scratch, be sure to also include those two extra columns. Also include the names in the first row (LVTrans will read the entire file and delete the first row). The file is otherwise a standard tab delimited text file. Any number of rows can be used.

X	WH(ave3)	WB(ave3)	WH(ave5)	WB(ave5)
0,000000	0,164254	-0,948942	0,152527	-0,955347
0,285599	0,309847	-0,424830	0,308869	-0,438380
0,571199	0,374925	-0,100963	0,374720	-0,097160
0,856798	0,426420	0,245526	0,431562	0,245565
1,142397	0,543534	0,602261	0,541032	0,597620
1,427997	0,631000	0,799000	0,633000	0,800000
1,713596	0,752333	0,885333	0,751400	0,876000
1,999195	0,856000	0,800000	0,856000	0,792400
2,284795	0,961667	0,582667	0,963200	0,581000
2,570394	1,091333	0,373667	1,093200	0,381200
2,855993	1,232667	0,340333	1,226800	0,344800
3,141593	1,242031	0,483844	1,240853	0,489306
3,427192	1,121949	0,657088	1,114802	0,648769
3,712791	0,826895	0,641204	0,819510	0,627954
3,998391	0,367907	0,418965	0,360437	0,412582
4,283990	-0,188590	0,021498	-0,183956	0,013666
4,569589	-0,678444	-0,507757	-0,670155	-0,507433
4,855189	-1,155058	-1,309028	-1,117715	-1,262142
5,140788	-1,201711	-1,773341	-1,194588	-1,766519
5,426387	-1,016465	-1,952184	-1,014921	-1,945745
5,711987	-0,715865	-1,769065	-0,692466	-1,773994
5,997586	-0,135086	-1,444664	-0,159626	-1,435034
6,283185	0,164254	-0,948942	0,152527	-0,955347

Figure 12.1 Formatting and default Suter data for pump

A note on smoothing of Suter curves. Suter curves are “circular” in that X runs from 0 to 2π, then starts from 0 again. It can go in either direction. It’s more like polar coordinates where X is the angle and WH and WB is x and y coordinates describing a circular shape. Smoothing is therefore best done on the Suter curves, not on the torque, head and flow curves obtained from the pump diagram. For transient calculation of pumps, smoothness of the (Suter) curves are much more important than being 100% correct at every point.

[20] The reason for this is from a long time ago. The pump element was one of the first elements that was made in LVTrans after the pipes. Excel was used to create the files by reading from hill charts. The first row of WH and WB was always way too “bumpy” to produce good results. First order smoothing (ave3) gave much better results, but second order (ave5) was even better. The final excel format has stuck since then.

12.21 VALVE INTERNAL SERVO

Valve Internal Servo is used to describe a valve with a first order servo, typically a gate or any kind of valve.

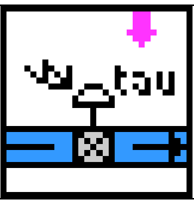
Name	Used for	Icon	Special
Valve Internal Servo	All kind of valves, gates and so on.		Opening and speed can be adjusted interactively.

Table 12.18 Valve Internal Servo, version 00001

Data	Format	Description
Cvp	value	Loss coefficient for positive flow direction.
Cvm	value	Loss coefficient for negative flow direction.
Closing time high [s]	value	Extrapolated closing/opening time when opening larger than switch-over.
Closing time low [s]	value	Extrapolated closing/opening time when opening smaller than switch-over.
Switch-over	value	Value in pu where the servo switches closing/opening time
Tk_s	value	Time constant for 1st order servo
tau_0	value	Valve opening in pu at start of simulation.
use-cv-file?	TRUE/FALSE	Use a separate file for Cv values?
cv-file-name	string	Name of the file with Cv values
D [m]	**value**	**Inner diameter of valve (for calculations of velocity in for instance pipe break valves)**

An example of a Cv file is found in the “useful\cv” folder. The Cv file itself is merely to columns where the first column is opening in pu (0.0 to 1.0) and the second column is the corresponding Cv value. The example file is for a typical gate valve. The format is tab delimited. Any number of rows can be used. Data for the file can be taken from valve manufacturers or hand books. Some Excel work is normal to get the Cv values, and to get them in the wanted format for LVTrans.

```
0,011111111    262,4550153
0,022222222    67,12346385
0,033333333    30,95682868
0,044444444    18,31729039
0,055555556    12,48885028
0,066666667    9,346294219
0,077777778    7,476091196
0,088888889    6,287810797
0,1  5,49942534
0,111111111    4,962439074
0,122222222    4,592653186
0,133333333    4,339464254
0,144444444    4,171001147
0,155555556    4,066406611
0,166666667    4,011591976
0,177777778    3,996787368
0,188888889    4,015069303
0,2  4,061444241
…
…
0,811111111    42,79879185
0,822222222    44,85624295
0,833333333    47,02358727
0,844444444    49,30796973
0,855555556    51,71710409
0,866666667    54,25932641
0,877777778    56,94365438
0,888888889    59,77985334
0,9  62,77850968
0,911111111    65,95111262
0,922222222    69,31014561
0,933333333    72,86918846
0,944444444    76,64303169
0,955555556    80,64780493
0,966666667    84,90112113
0,977777778    89,42223895
0,988888889    94,23224602
1    99,35426597
```

Figure 12.2 Example of the Cv file format

12.22 PRV

The PRV (Pressure Release Valve) is a variant of the Valve Internal Servo. It can be found by "Select-type" by right clicking on a valve.

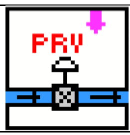
Name	Used for	Icon	Special
PRV, Pressure Release Valve	PRV in connection with a turbine and PID.		Must be connected to a Turbine governor (PID).

Table 12.19 PRV, version 00001

Data	Format	Description
Cv	value	Loss coefficient.
Opening time [s]	value	Extrapolated opening time.
Closing time [s]	value	Extrapolated closing time.
Closing time low [s]	value	Extrapolated closing/opening time when opening smaller than switch-over.
Tk_s	value	Time constant for 1st order servo
tau_0	value	Valve opening in pu at start of simulation.
Max y speed	value	The speed is extrapolated closing time from the PID
use-cv-file?	TRUE/FALSE	Use a separate file for Cv values?
cv-file-name	string	Name of the file with Cv values
D [m]	**value**	**Inner diameter of valve (for calculations of velocity in for instance pipe break valves)**

Max- y speed needs some explanation. This is the speed at which the servo on the guide vanes moves. This is taken from the "kappa" of a PID. If the main servo closes with an extrapolated time of 4 s, then setting max y speed to 4.1 s will open the PRV when the guide vanes closes at max speed. It will not open if the guide vanes closes at for instance 5 s. This is approximately how the PRV work.

It can also use a Cv file for better accuracy. This is exactly as a normal valve.

Note! For the time being the PRV does not fully represent a normal PRV. It should also have a "relax time" for which it stays open before it closes again. This, or a similar functionality, has not yet been modelled. It must therefore be used with caution.

There is an example plant with a PRV in the user folder.

12.23 OPEN CHANNELS

Open channels are free surface channels. They are calculated using finite difference methods, not MOC. This, and the very nature of open channel flow, makes them much fiddlier regarding stability.

A trick to make open channel start in a stable manner is to start at zero flow.

There are three different versions. They all use the same parameters as shown in Figure 12.3 and Figure 12.4.

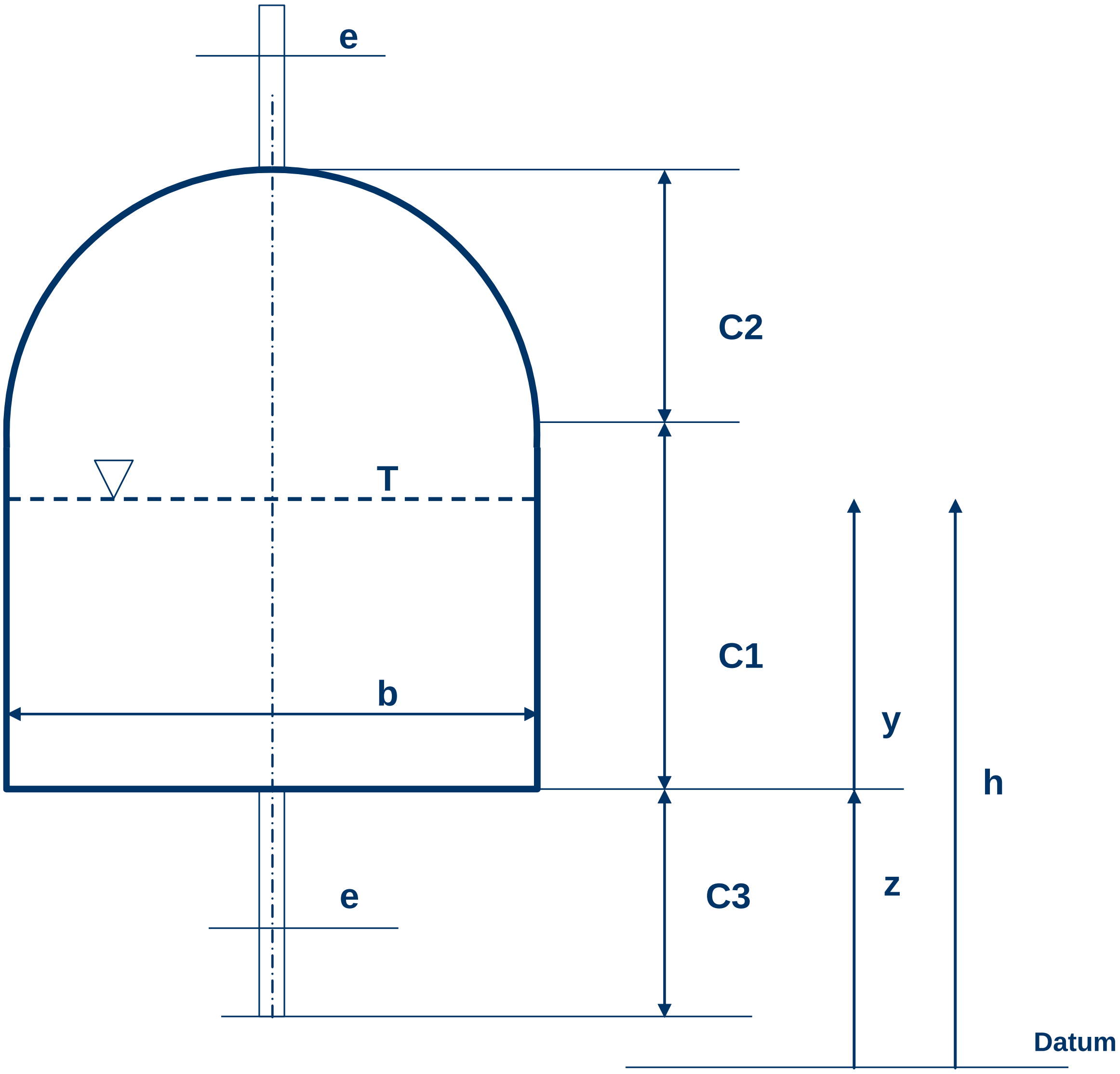


Figure 12.3 Open channel cross sectional parameters

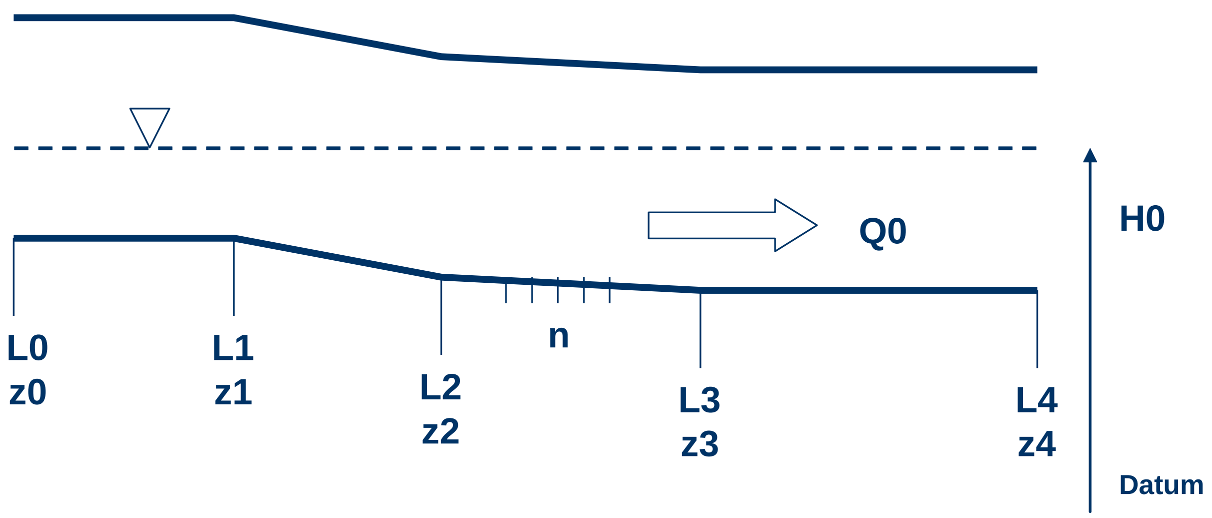


Figure 12.4 Open channel length vise parameters.

The Manning number is defined as:

$$Q = \frac{k_n}{n} A R^{2/3} S_f^{1/2}$$

Here n = Manning number, kn = 1.0 (SI enheter). Typical values for n is 0,013-0,015 for concrete and 0,025-0,045 for blasted rock. However, to find the Manning number precisely for arbitrary channels is literally impossible without measurements. If no measurements exist, one has to use ”best guess”.

12.24 OPEN CHANNEL NL NORMAL

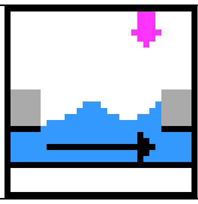
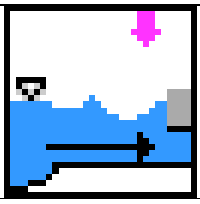
Name	Used for	Icon	Special
Open Channel NL normal (NL = No Level).	Free surface channel between two pipes.		Have to stay between two pipes.

Table 12.20 Open channel parameters (see figures)

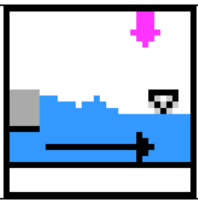
Data	Format	Description
Mannings	value	Manning number, Se definiton
B [m]	**value**	**Width of channel**
C1 [m]	**value**	**Height of square shaped part.**
C3 [m]	value	Depth of numerical "fixing part"
Q_start	**value**	**Flow at start of calculations.**
H_start	**value**	**Water level at start, measured from datum.**
e [m]	*value*	*Width of numerical "fixing part"*
N [-]	value	Number of elements in each sub-element.
delta	value	Numerical constant (0,5 – 1,0)
L0 [m]	**value**	**Parameter for profiling** **NOTE! L0 has to be zero**
L1 [m]	**value**	**Parameter for profiling**
L2 [m]	**value**	**Parameter for profiling**
L3 [m]	**value**	**Parameter for profiling**
L4 [m]	**value**	**Parameter for profiling**
z0 [levelg]	**value**	**Parameter for profiling**
z1 [levelg]	**value**	**Parameter for profiling**
z2 [levelg]	**value**	**Parameter for profiling**
z3 [levelg]	**value**	**Parameter for profiling**
z4 [levelg]	**value**	**Parameter for profiling**

12.25 OPEN CHANNEL LL NORMAL

Name	Used for	Icon	Special
Open Channel LL Normal (LL = Level to the Left).	Free surface channel with a reservoir to the «left».		Have to have a pipe to the «right». Level in the reservoir can be adjusted interactively.

All parameters as in Table 12.20.

12.26 OPEN CHANNEL LR NORMAL

Name	Used for	Icon	Special
Open Channel LR Normal (LR = Level to the Right).	Free surface channel with a reservoir to the «rightt».		Have to have a pipe to the «left». Level in the reservoir can be adjusted interactively

All parameters as in Table 12.20.

13 SAVING AND COPYING

All the data and files will reside inside the system folder for the actual system. Eventual documentation, calculations etc can also be stored in that folder. A zipped folder (with no additional data or documentation) typically has the size of 100-200 kB. This can easily be sent my mail[21] to others.

Note! All systems (plants) must reside under “User Files” or LVTrans may not find the needed files to execute.

[21] At least that used to be the case. Internet security may prevent this.