# HydroSimulator

v4.2

*Quick Guide*

*Developer: Gerard Sanz Estapé*

*gerard.sanz@upc.edu*

*May 2013*

*SAC – UPC*

**NOTE**

This software is under development. Comments and corrections are welcomed.

A FAQ section is included in this guide with all the received from the users.

# Index

# 1. HydroSimulator

HydroSimulator consists of two basic folders, complemented by extra modules. The basic folders are:

- EPA_TOOLKIT: EPANET Toolkit, required for simulating the water networks.
- HydroSimulator_Basic: Main folder with the basic functions of the software.

Two extra folders with a pressure control module and a demand pattern calibration module are also available:

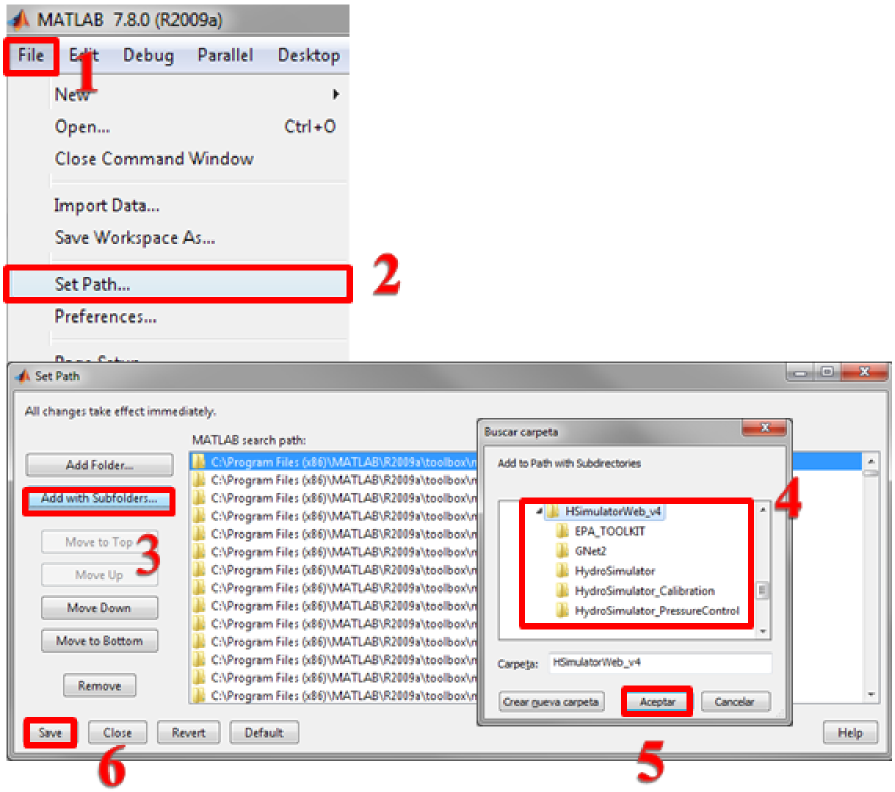
- HydroSimulator_PressureControl
- HydroSimulator_Calibration

**Important note:** The path to the extra folders cannot contain spaces.

The used water distribution networks are located in independent folders:

- GNet2
- GNet3

In order to make all the modules and the toolkit work, the path of all the folders has to be defined in Matlab, as shown next:

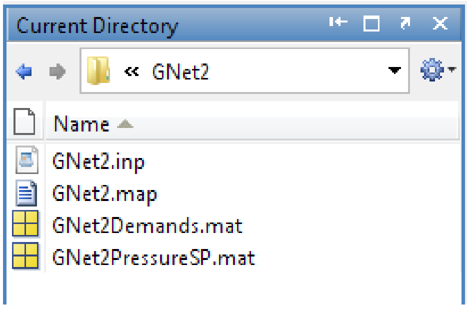


# 2. Use of HydroSimulator

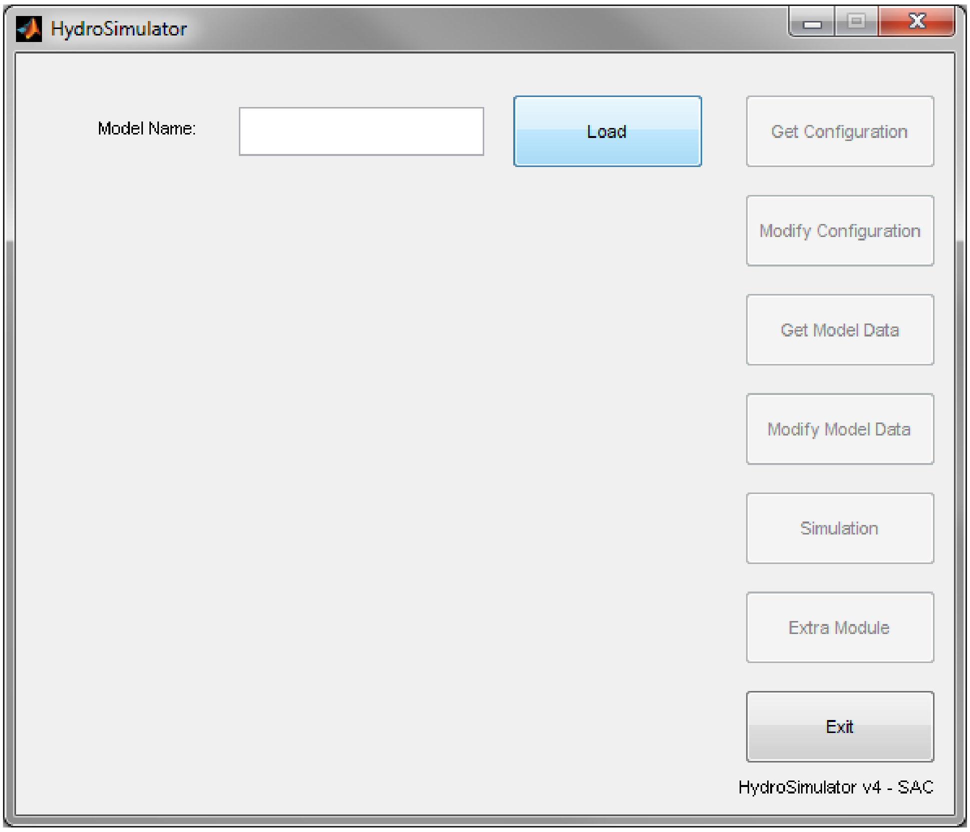
This document presents a demonstration of how the HydroSimulator interface works.

## 2.1. Starting the application

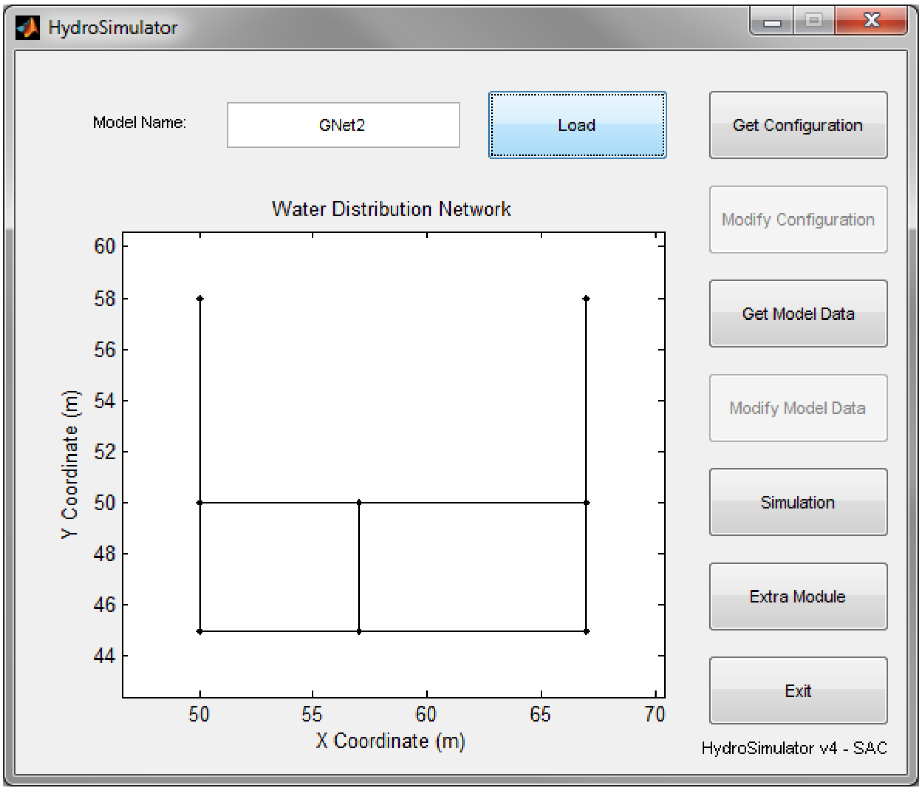
The working folder is always the one where the network data is located. In this case this folder is called: GNet2.



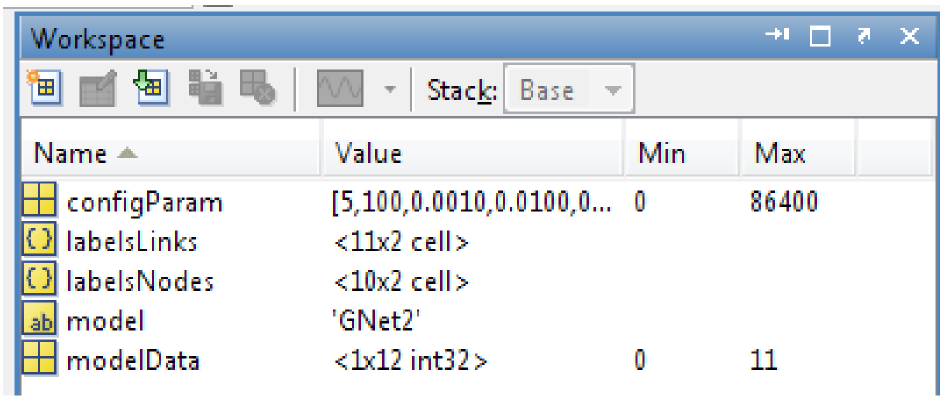

Execute the simulator by writing "HydroSimulator" in the command window. The following window appears:



Load the used network (GNet2). The .inp file and the .map file are required:

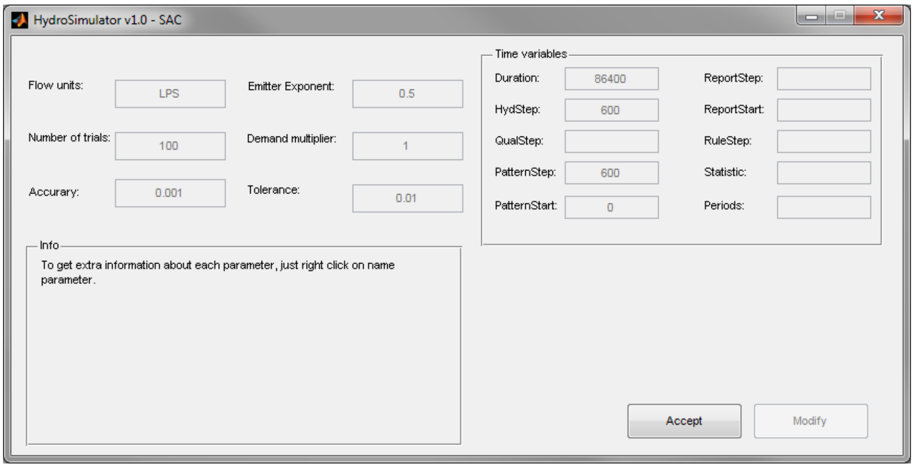



Some information about the network is stored in the workspace:

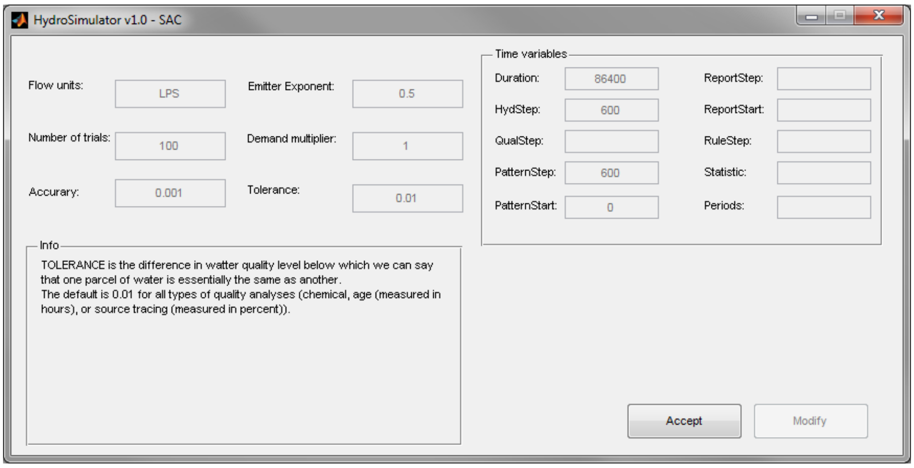



- configParam: Array with information for the “Get Configuration” module.
- labelsLinks: Labels and types of all the network’s links.
- labelsNodes: Labels and types of all network’s nodes.
- model: Name of the loaded water distribution network model.
- modelData: Array with information for the “Get Model Data” module.

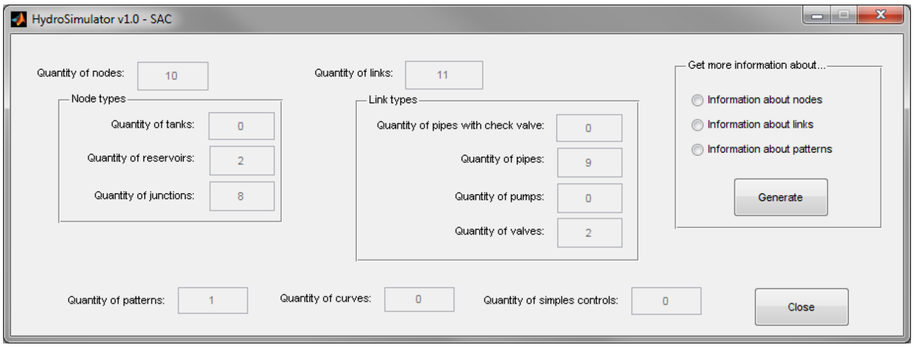
Now it is possible to obtain the simulation parameters and the model data. When clicking the “Get Configuration” button the following window appears:



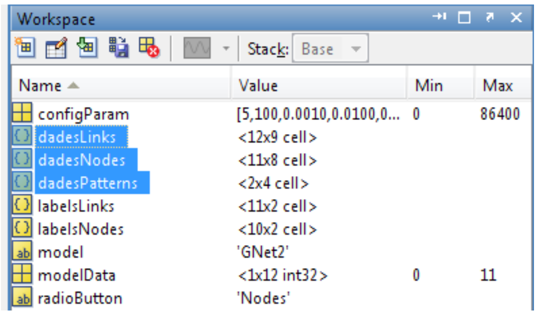

Right click on the parameter name in order to obtain more detailed information:




Click “Get Model Data” to obtain general information about nodes, links, etc.:



Select desired radio button and click “Generate” to obtain detailed information about each type of element:

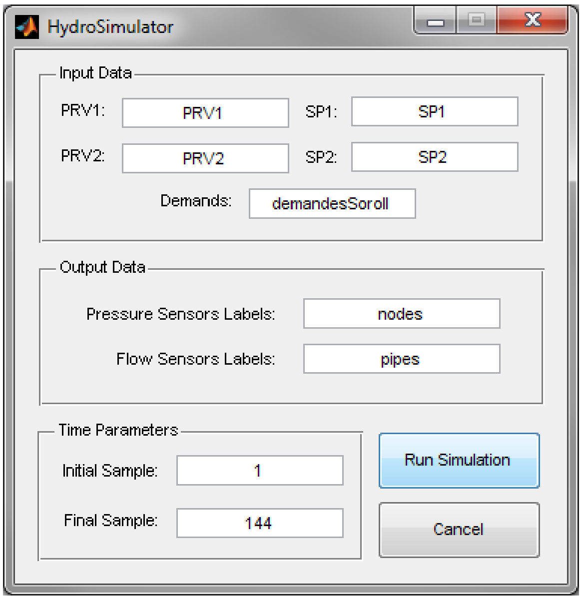



- dadesNodes: Labels, indexes, base demands, associated patterns, emitter coefficients, elevations, initial levels and types of the network junctions, tanks and reservoirs.
- dadesLinks: Labels, indexes, diameters, lengths, roughness, minor loss coefficients, initial states, initial configurations and types of the network pipes, valves and pumps.
- dadesPatterns: Labels, indexes, lengths and values of the network patterns.

Click “Simulation” to set the simulation parameters. Before running the simulation, load the .mat files called “GNet2Demands.mat” and “GNet2PressureSP.mat”. Define in the workspace the cell “nodes” and “pipes” as follows:

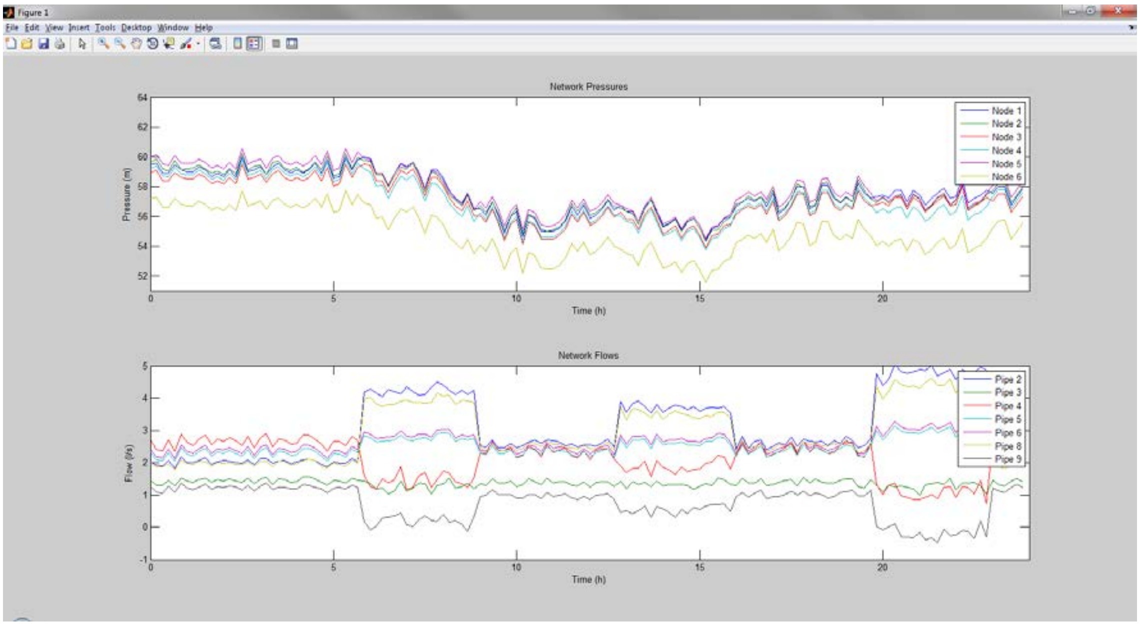
```
>> nodes=labelsNodes(1:6,1)
>> pipes=labelsLinks(1:7,1)
```

Define the simulation parameters as shown in the next figure:



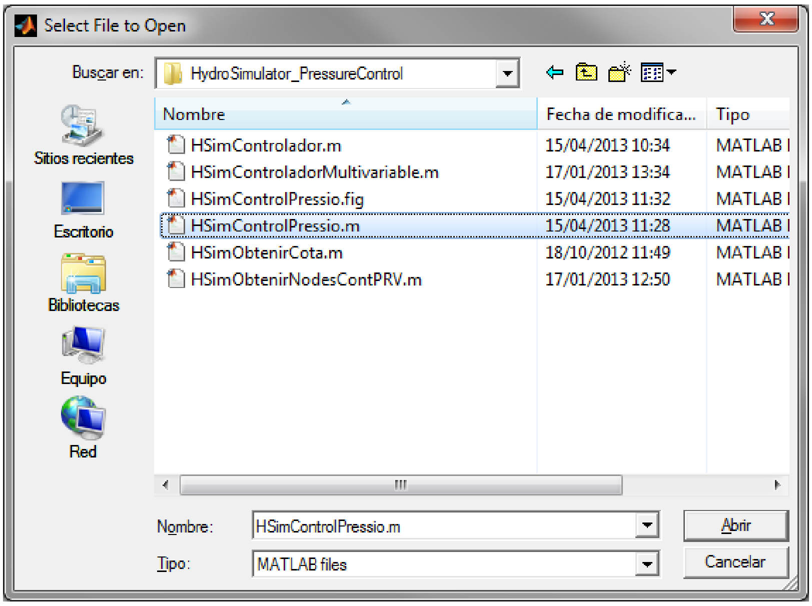
After running the simulation, matrices “Flows” and “Pressures” are stored in the workspace, containing the measured flows and pressures of the defined links “links” and nodes “nodes”. Now it is possible to plot the pressures using Matlab functions:

```
>> x=0:143; x=x/6;
>> subplot(2,1,1)
>> plot(x,Pressures')
>> xlabel('Time (h)'); ylabel('Pressure (m)'); axis([0 24 51 64])
>> legend('Node 1','Node 2','Node 3','Node 4','Node 5','Node 6')
>> title('Network Pressures')
>> subplot(2,1,2)
>> plot(x,Flows')
>> xlabel('Time (h)'); ylabel('Flow (l/s)'); axis([0 24 -1 5])
>> legend('Pipe 2','Pipe 3','Pipe 4','Pipe 5','Pipe 6','Pipe 8','Pipe9')
>> title('Network Flows')
```

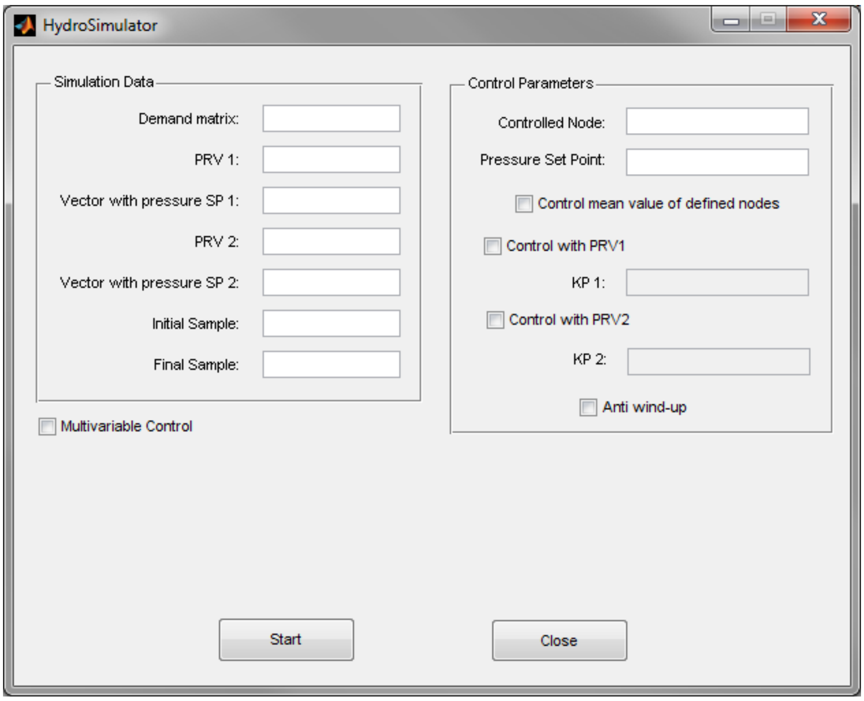


# 3. Working with extra modules: Pressure control

For running extra modules, click the "Extra Module" button. A navigation window appears. Navigate to the HydroSimulator_v4\HydroSimulator_PressureControl folder and open HSimControlPressio.m:




The following module is shown:

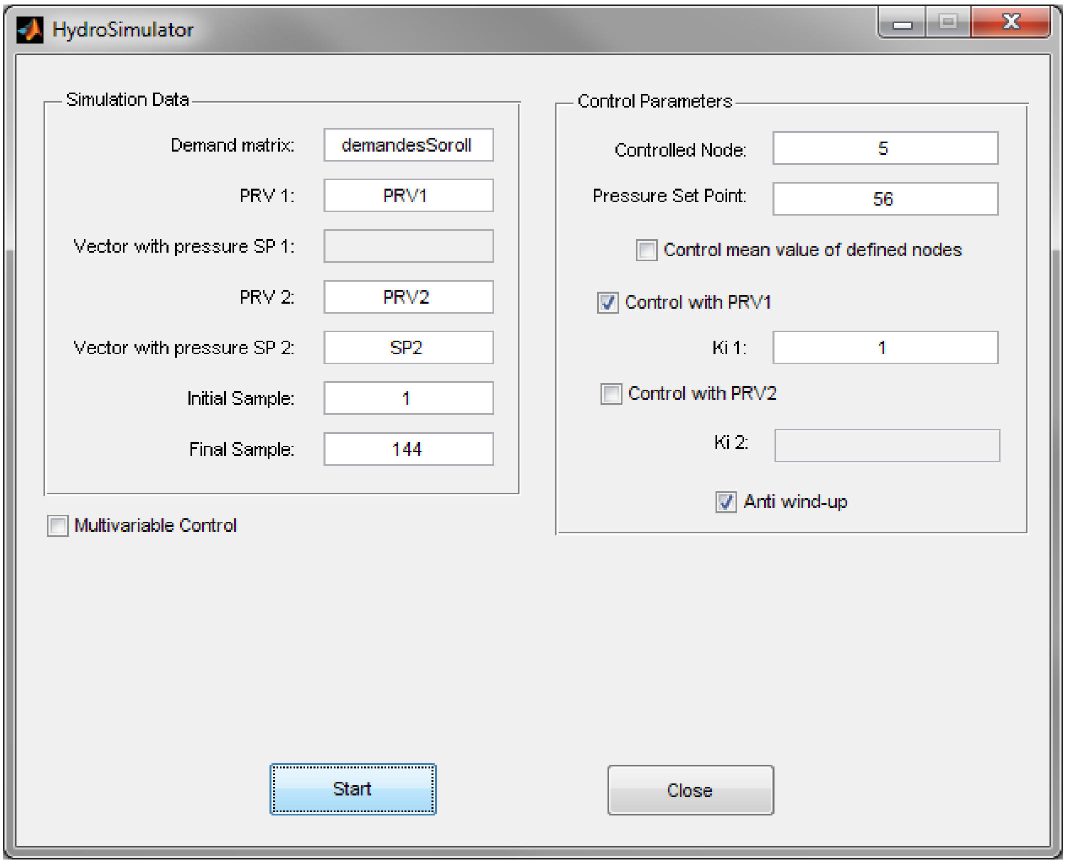


Different types of control can be performed:

- Monovariable control with PRV1.
- Monovariable control with PRV2.
- Monovariable control with PRV1 and PRV2 (Split-range).
- Monovariable control with PRV1 and/or PRV2 of mean pressure value of several nodes.
- Multivariable control.

### 3.1.1. Monovariable control with PRV1

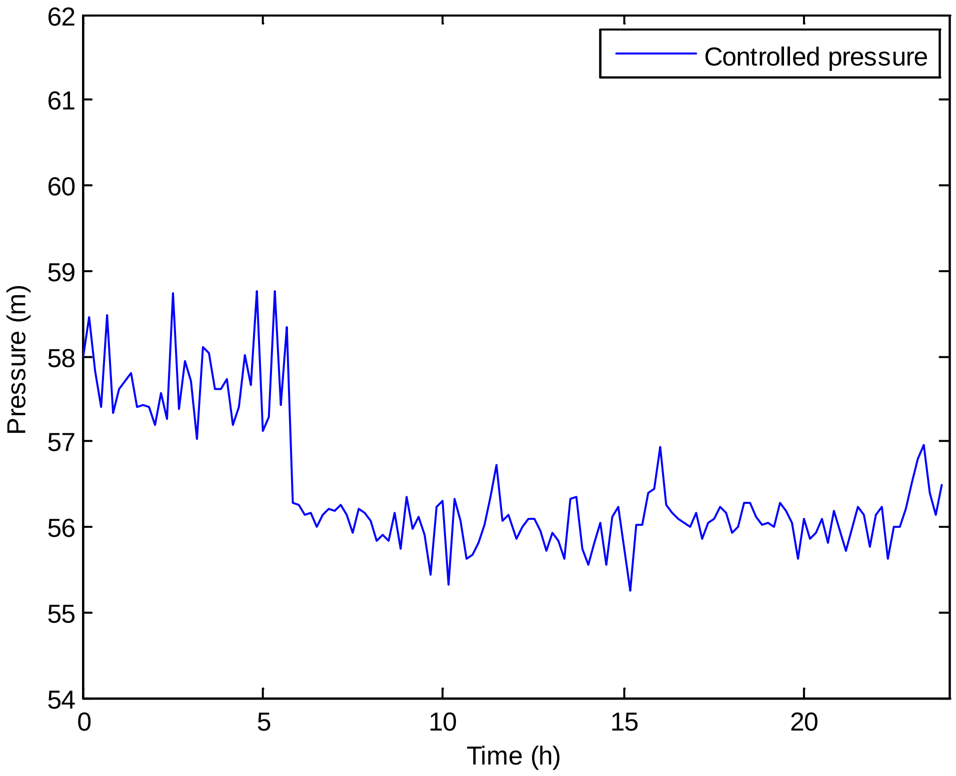
Set parameters as shown in next figure and Start the pressure control algorithm:



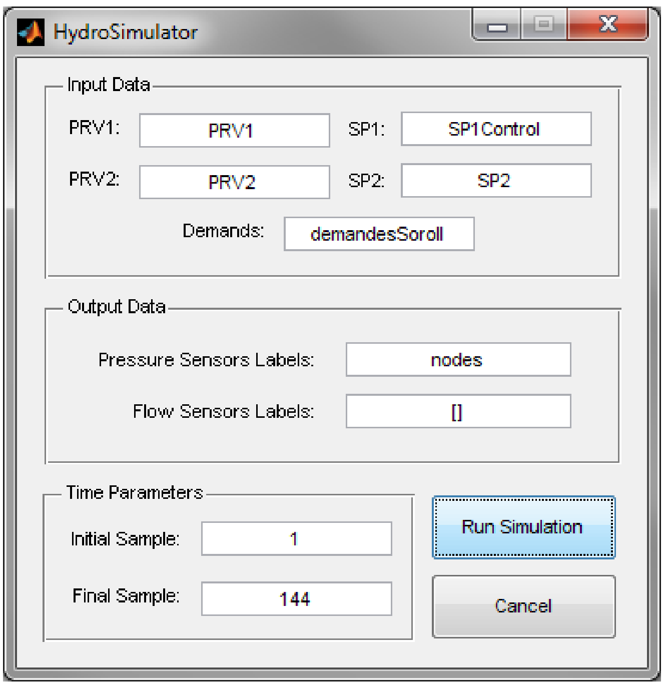

When the simulations finish, some information is stored in the workspace:

- ControlledNodePressure: Pressure at the controlled node.
- SP1Control: Set point in the controlled valve 1.
- SP2Control: Set point in the controlled valve 2.

Again, the pressure at the controlled node can be represented:

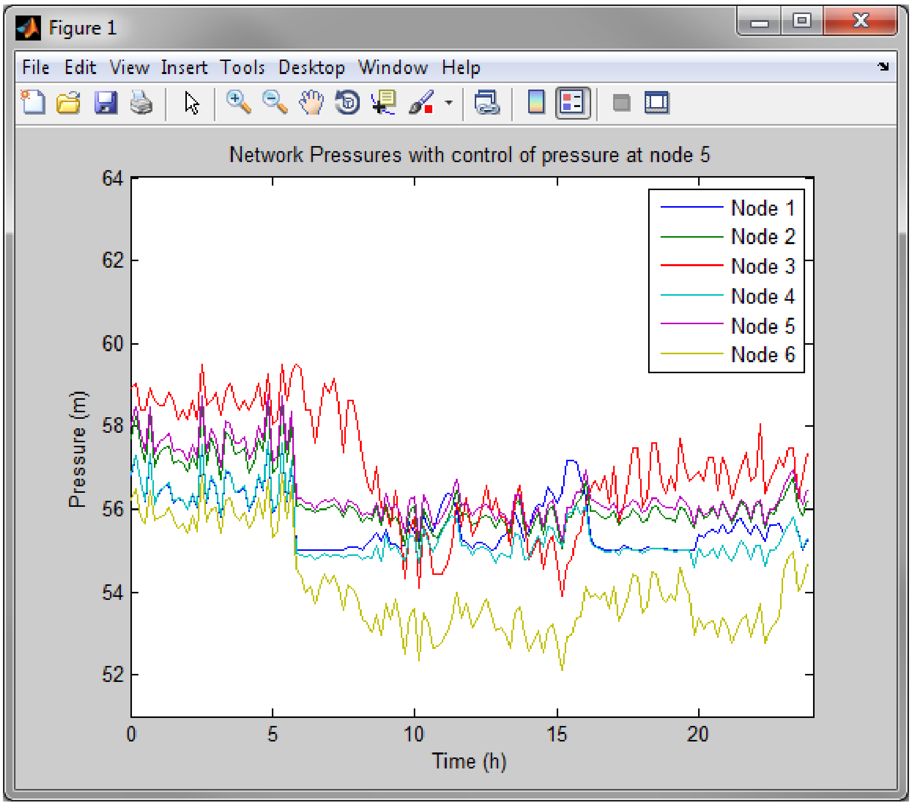



It can be seen that most time pressure is controlled at the desired set point. SP1Control can be used in the simulation module as Set Point for PRV1 in order to observe the resulting pressures in the rest of the nodes. Close the Extra Module window. Open the simulation module. Set parameters as shown in next figure:



Now it is possible to plot these resulting pressures using Matlab functions:

```
>> x=0:143;
>> x=x/6;
>> plot(x,Pressures')
>> xlabel('Time (h)')
>> ylabel('Pressure (m)')
>> axis([0 24 51 64])
>> legend('Node 1','Node 2','Node 3','Node 4','Node 5','Node 6')
>> title('Network Pressures with control of pressure at node 5')
```

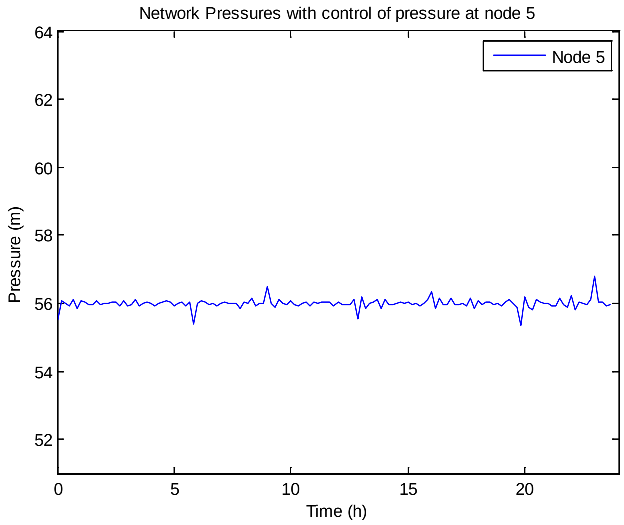



### 3.1.2. Monovariable control with PRV2

This type of control works similar to the previous one, but this time the valve used for control the pressure in a node is PRV2. Check the box below PRV2 id and run the algorithm as in the previous case.

### 3.1.3. Split-range control

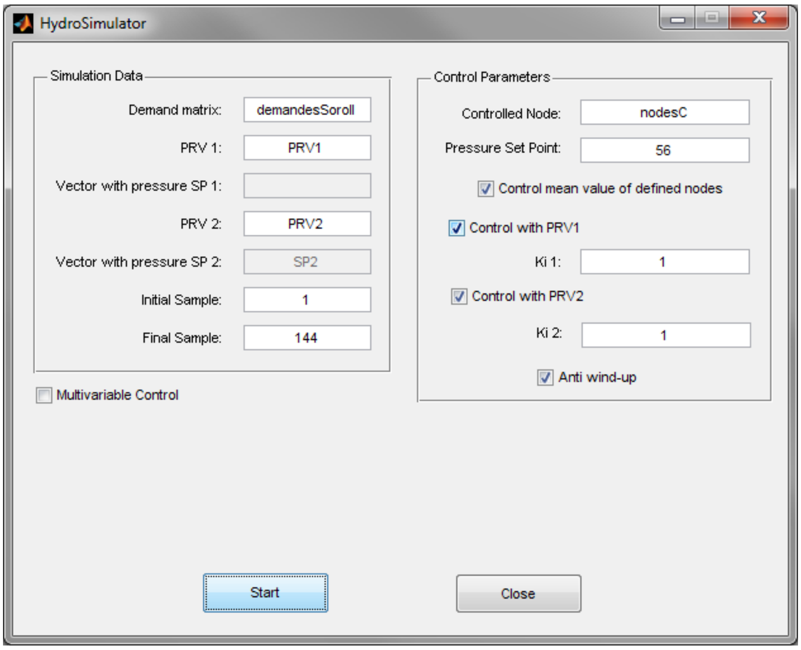
In this type of pressure control, the two actuators (pressure reduction valves) are used. Check both PRV boxes and observe how the result improves:



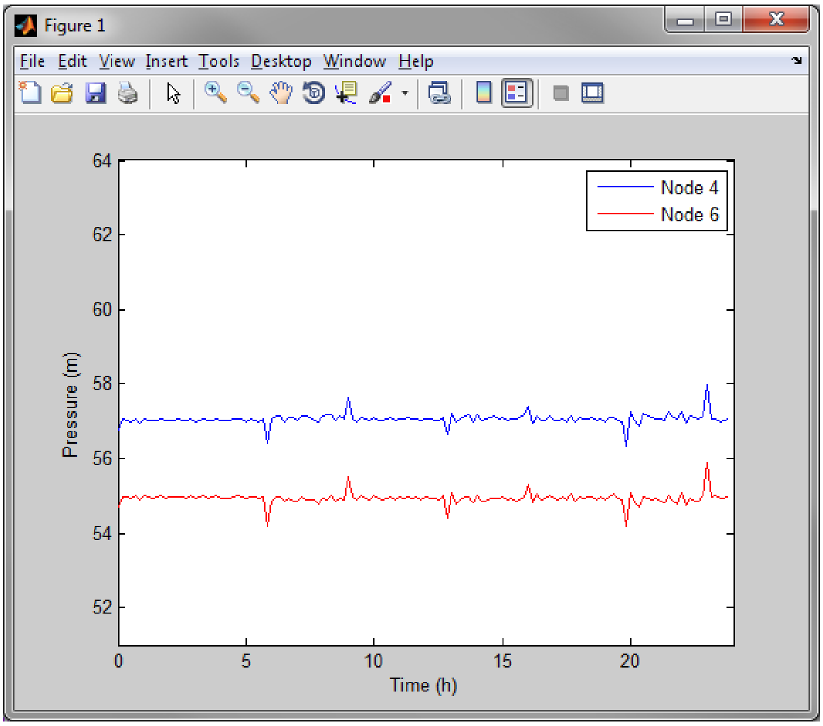

### 3.1.4. Monovariable control of mean value

This type of control is the same as the previous ones, but now the node label is a cell with more than one node id. The mean pressure of these nodes is controlled by PRV1 and/or PRV2. For example, for controlling the mean pressure value of nodes 4 and 6:

```
>> nodesC={'4' '6'}
```

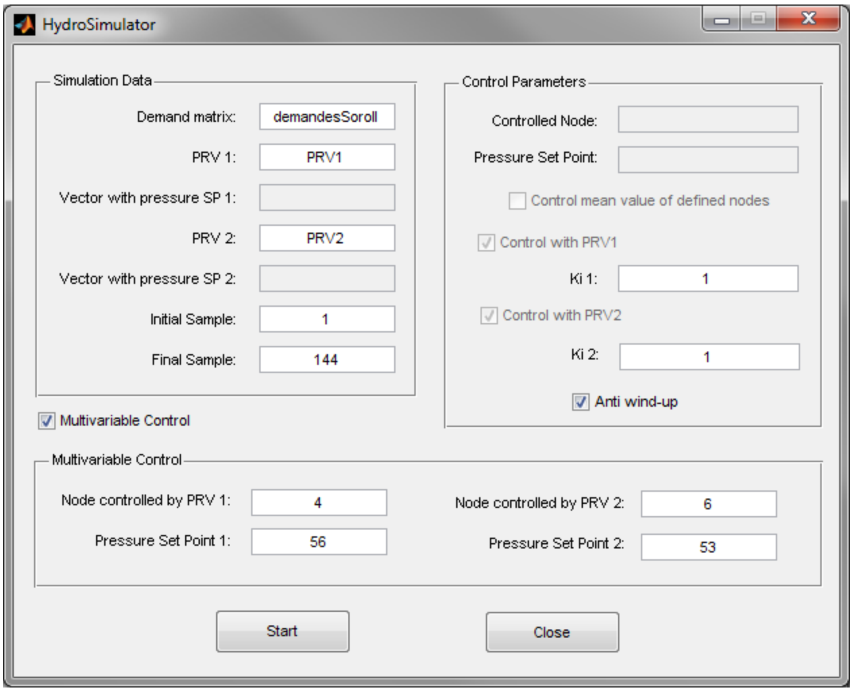


The resulting ControlledNodePressure is the mean of both nodes. In order to see the individual pressures of nodes ‘4’ and ‘6’, use the simulation module with sensors in nodes ‘4’ and ‘6’:



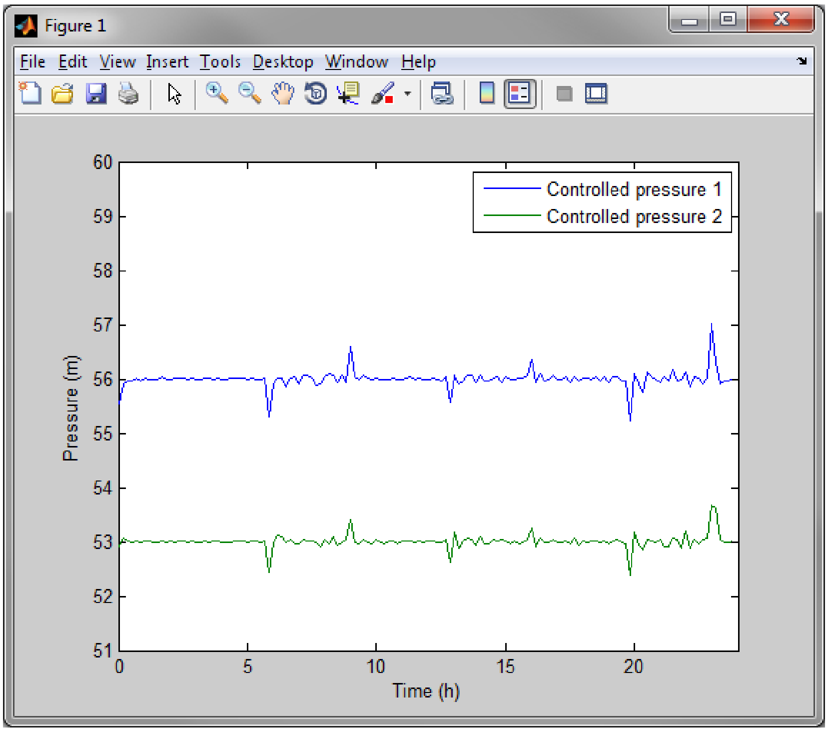

### 3.1.5. Multivariable control

When this box is checked, some modifications appear in the window. Set the parameters as shown next:



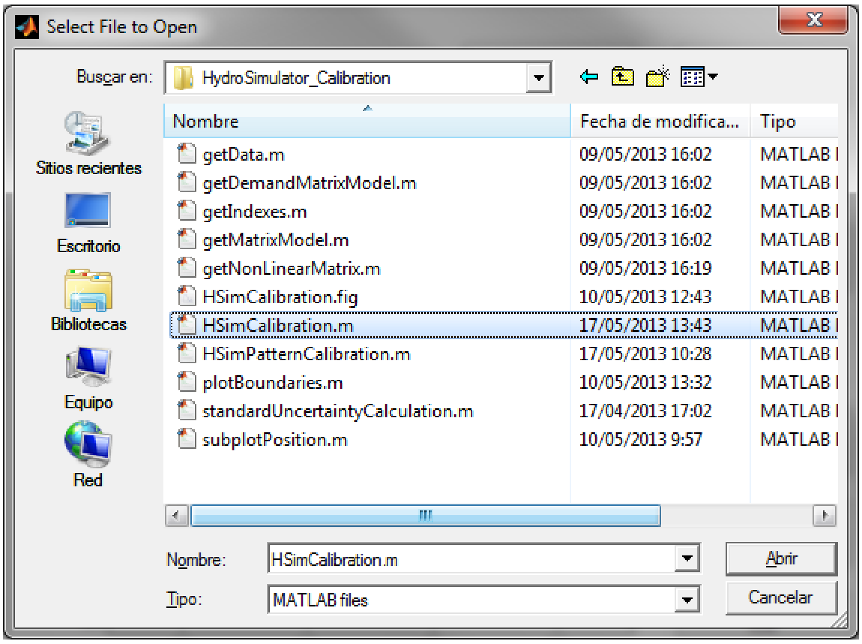
When the simulations finish, some information is stored in the workspace:

- Row 1 of ControlledNodePressure: Pressure at the controlled node 1.
- Row 2 of ControlledNodePressure: Pressure at the controlled node 2.
- SP1Control: Set point in the controlled valve 1.
- SP2Control: Set point in the controlled valve 2.

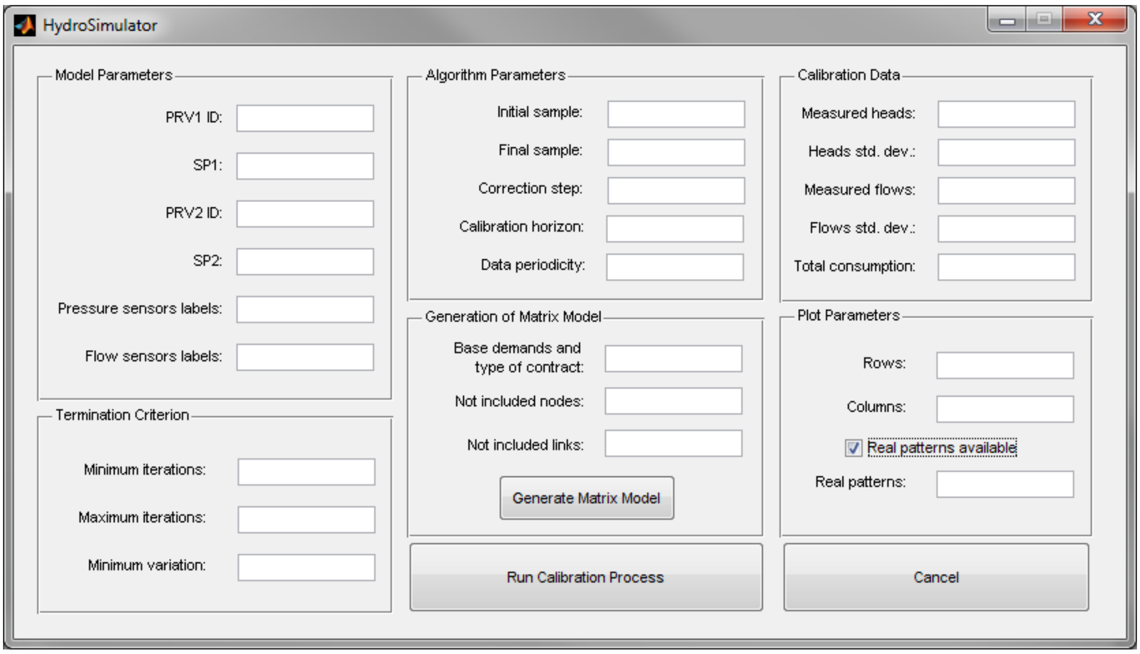


# 4. Calibration Module

An extra module for performing demand pattern calibration is also available. Remember to set the path of the new folder, and to work from the folder with the network *.inp* file. This time, the *GNet3.inp* network is used. Open the calibration module as indicated next:




The following window appears:



**Generation of Matrix Model**

The matrix model must be generated before running the calibration process. This function stores in the Matlab workspace the matrices:

- B: Connection between nodes and pipes
- BDM: Base Demand Matrix, with the average consumption of each node.
- PMM: Pattern Matching Matrix, with the relation between each node and pattern.
- R: Conductivity parameter of each pipe.
- indexRowB, ,indexColB: EPANET index of the nodes (rows) and pipes (columns) of matrix B.

The following data is required:

- Base demands and type of contract: Cell array with columns:
    - $1^{st}$ column: Label of the demand nodes.
    - $2^{nd}$ column: Base demand of the demand nodes.
    - $3^{rd}$ column: Type of contract of the demand nodes.
- Not included nodes and links: Cell with the labels of the nodes and links not included in the matrix model of the network.

**Model Parameters**

Boundary conditions and sensors definition:

- PRV1 and PRV2 ID: Label of each pressure reduction valve.
- SP1 and SP2: Set points for the pressure reduction valves.
- Pressure sensors labels: Labels of the pressure sensors.
- Flow sensors labels: Labels of the flow sensors.

If no pressure or flow sensors are available, define them as an empty array: *[]*.

**Termination Criterion**

Definition of the termination criterion that halts the methodology:

- Minimum and maximum iterations: Number of minimum and maximum iterations of the methodology for each sample being calibrated.
- Minimum variation: Minimum variation between error at sample *k* and error at sample *k-1* which cause the current sample calibration to halt.

**Algorithm Parameters**

Definition of the time-related parameters, as well as the step size:

- Initial and final sample: Initial and final hour of the calibrated demand patterns.
- Correction step: Parameter that controls the step size for the parameter correction.
- Calibration horizon: Number of samples associated to the same hour used.
- Data periodicity: Number of samples between samples of the same hour (generally, 24).

**Calibration data**

Measurements used for the calibration methodology:

- Measured heads and flows: Matrix of the measured heads and flows, with number of rows equal to the number of pressure or flow sensors.
- Heads and flows standard deviation: Standard deviation associated to the sensor uncertainty. One unique value for each type of sensor.

**Plot parameters**

Configuration of the graphical results:

- Rows and columns: Number of rows and columns for plotting the calibrated parameters. The product rows x columns must be equal or greater than the number of calibrated parameters (different type of contracts).
- Real patterns: Matrix with the real patterns (the rows correspond to the patterns, and the columns correspond to the time samples).

## 4.1. Generating an scenario

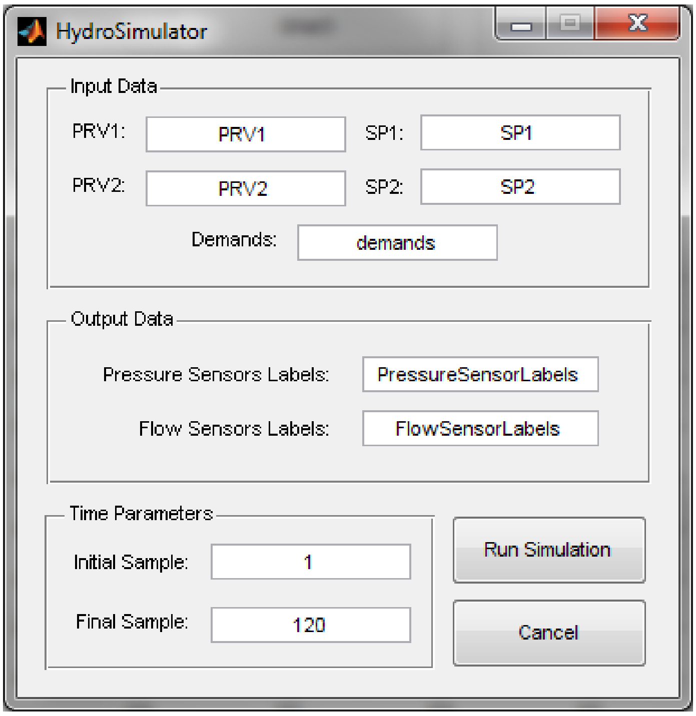
Use the provided demands (*GNet3_Geographic.mat)* to generate the demand matrix for the simulation module:

```
>> load GNet3_Geographic.mat
>> for i=1:10;dem=syntheticDemands{i,2};dTable(i,:)=dem;end
>> demands=[(1:10)' dTable];
```

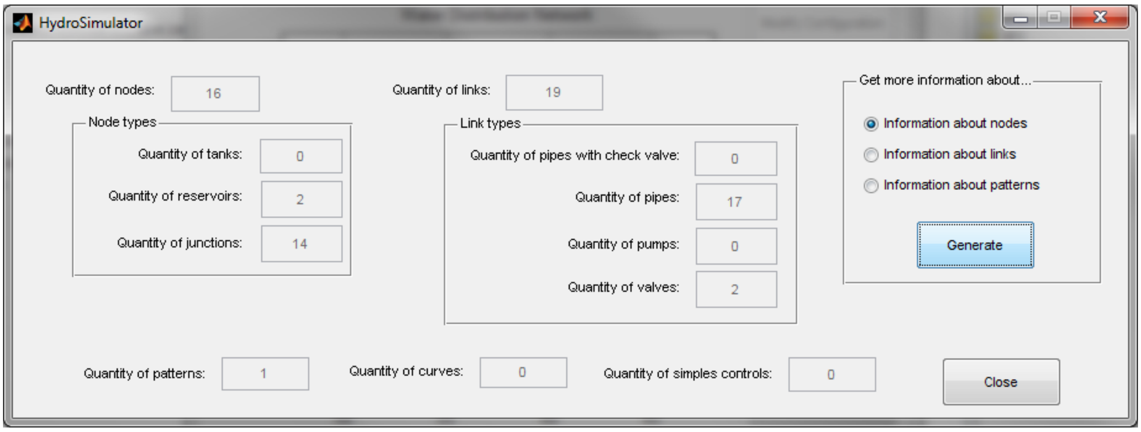
Define the used sensors and load the boundary set points:

```
>> PressureSensorLabels={'2' '7' '8' '10'}
>> load pressureInputs.mat
>> FlowSensorLabels={'PRV1' 'PRV2'}
```

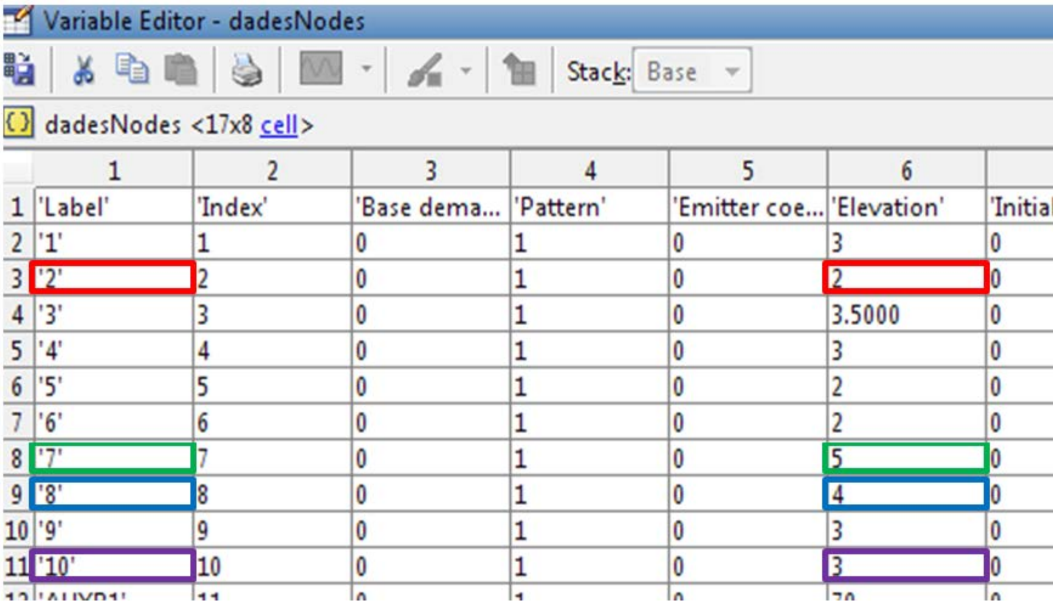
Simulate 5 days of data (120 hours):




The calibration module requires the head measurements (pressure + node elevation). In order to generate the measured heads matrix, it is necessary to get the elevation of each node. Open the *Get Model Data* module and get information about nodes:



Now, find the elevations of each measured node:




| | 1 | 2 | 3 | 4 | 5 | 6 | |
|---|---|---|---|---|---|---|---|
| 1 | 'Label' | 'Index' | 'Base dema... | 'Pattern' | 'Emitter coe... | 'Elevation' | 'Initial |
| 2 | '1' | 1 | 0 | 1 | 0 | 3 | 0 |
| 3 | '2' | 2 | 0 | 1 | 0 | 2 | 0 |
| 4 | '3' | 3 | 0 | 1 | 0 | 3.5000 | 0 |
| 5 | '4' | 4 | 0 | 1 | 0 | 3 | 0 |
| 6 | '5' | 5 | 0 | 1 | 0 | 2 | 0 |
| 7 | '6' | 6 | 0 | 1 | 0 | 2 | 0 |
| 8 | '7' | 7 | 0 | 1 | 0 | 5 | 0 |
| 9 | '8' | 8 | 0 | 1 | 0 | 4 | 0 |
| 10 | '9' | 9 | 0 | 1 | 0 | 3 | 0 |
| 11 | '10' | 10 | 0 | 1 | 0 | 3 | 0 |

And calculate the measured heads matrix:

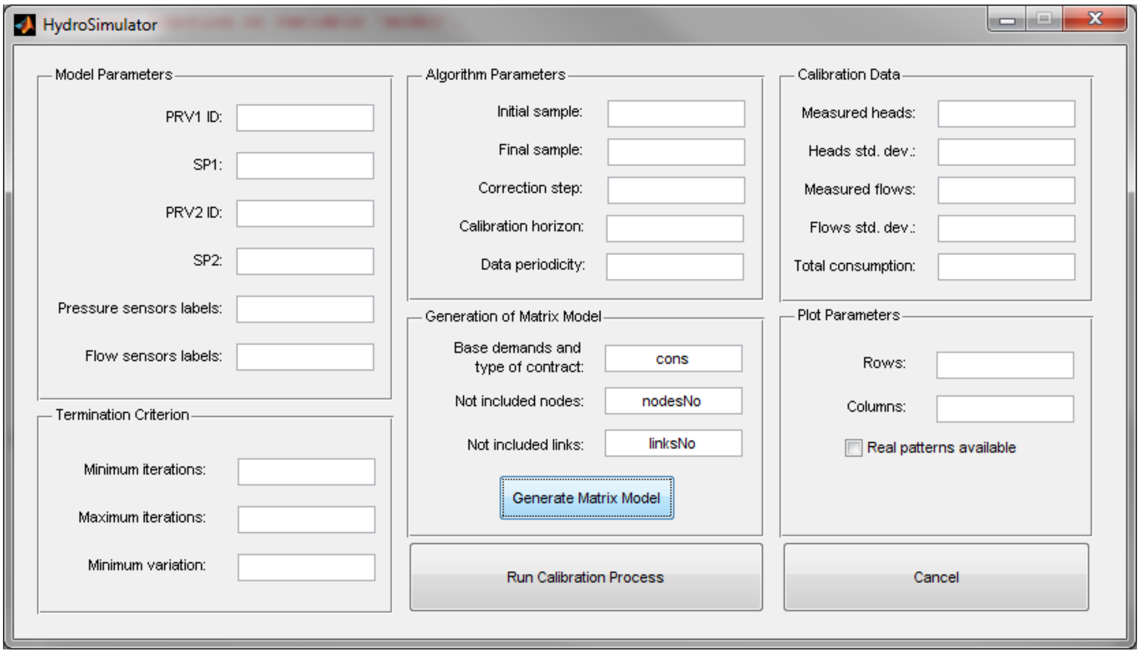
```
>> elevations=[2 5 4 3];
>> for i=1:4;Heads(i,:)=Pressures(i,:)+elevations(i);end
```

Calculate the total consumed demand:

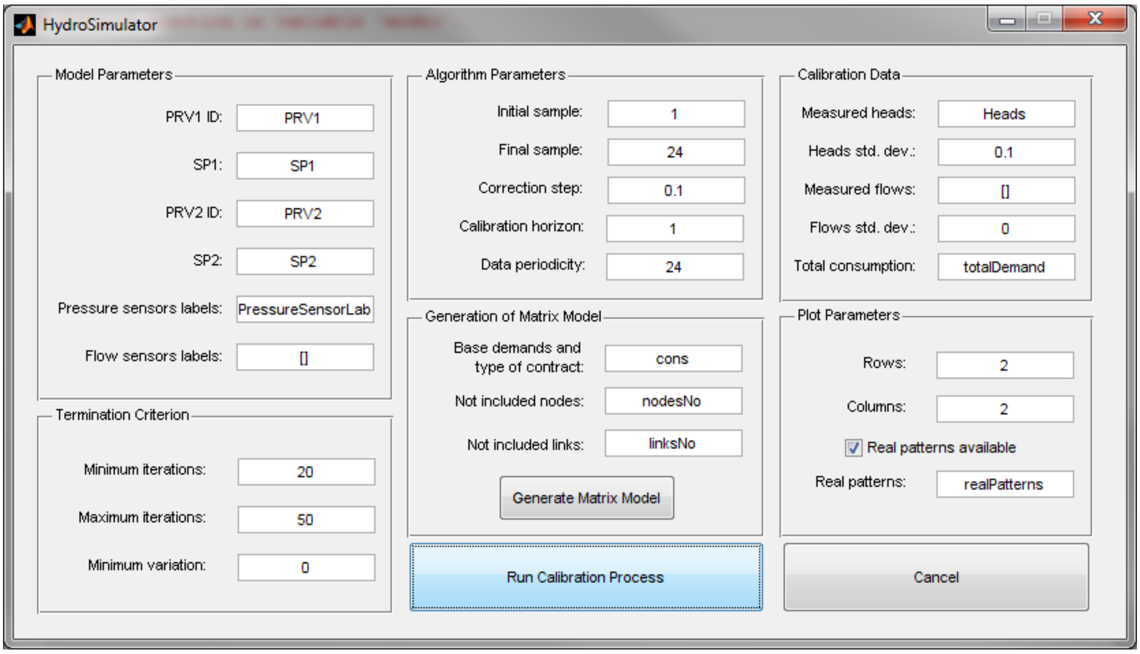
```
>> totalDemand=sum(Flows);
```

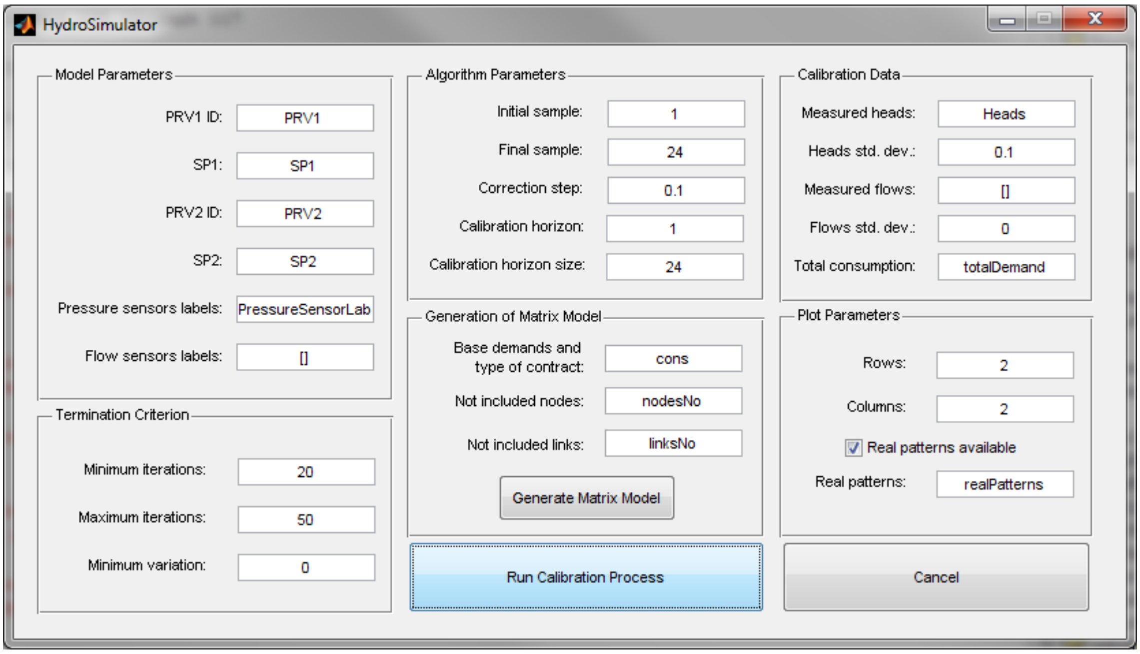
## 4.2. Running the calibration algorithm

Now, open the calibration module. Load *NodesLinksNO.mat* and define the parameters for generating the matrix model:

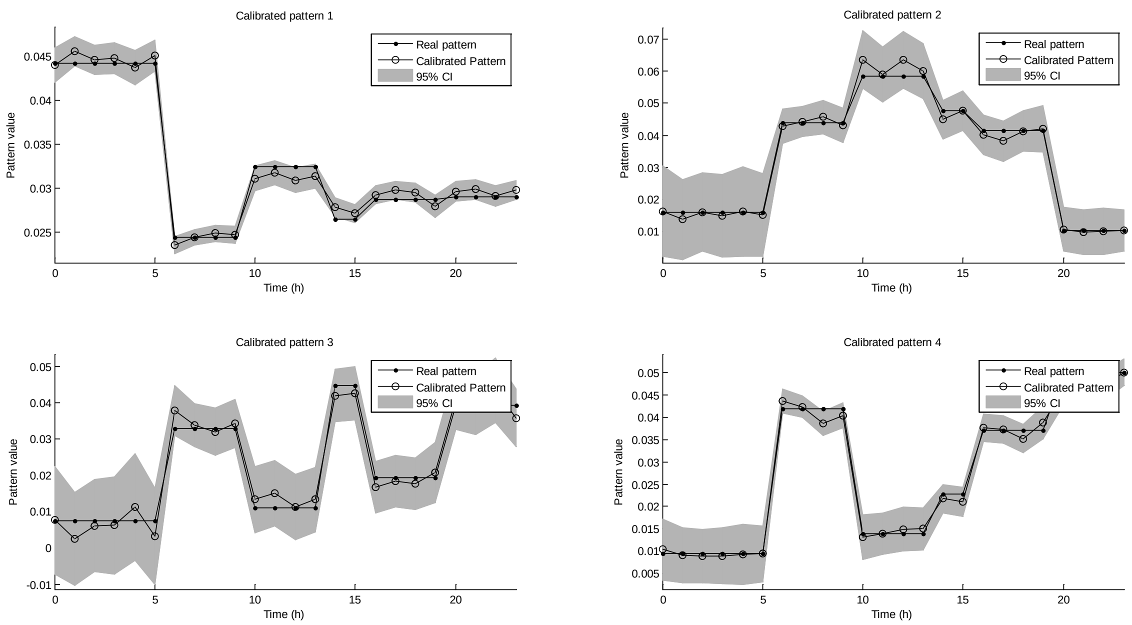



Define the rest of the parameters and perform the calibration process:

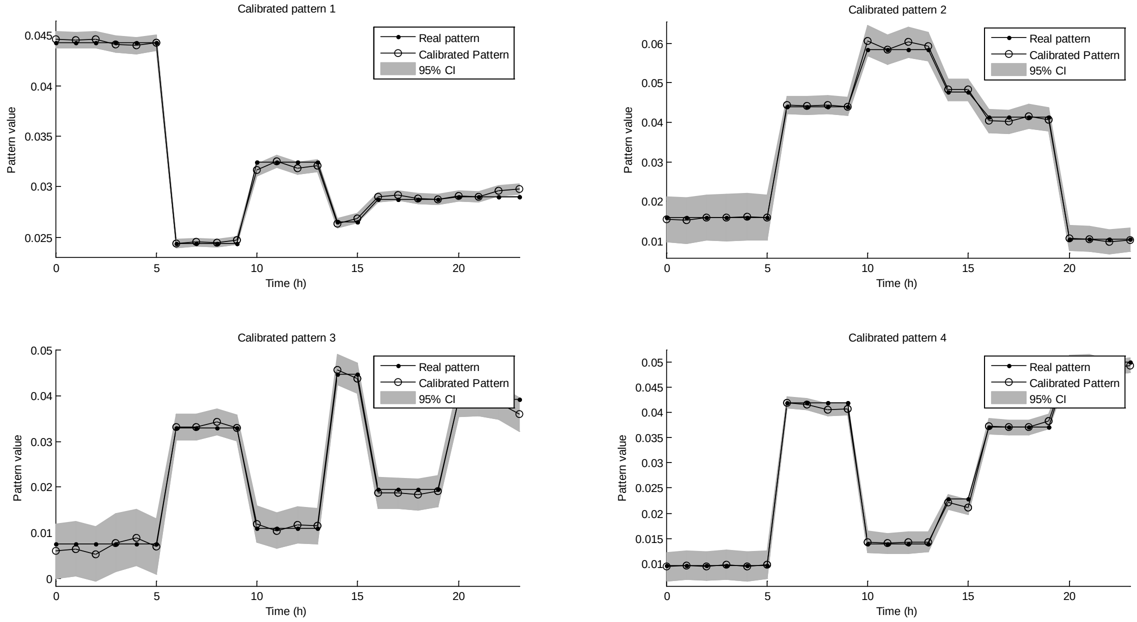





Although no noise is present in the measured pressures, we define that the standard deviation of the measurements is 0.1 m. The results are:




If the calibration horizon is increased (define it as 5 in the calibration module), the uncertainty in the calibrated patterns is reduced:


Calibrated pattern 1
Real pattern
Calibrated Pattern
95% CI
Pattern value
0.045
0.04
0.035
0.03
0.025
0
5
10
15
20
Time (h)
Calibrated pattern 2
Real pattern
Calibrated Pattern
95% CI
Pattern value
0.06
0.05
0.04
0.03
0.02
0.01
0
5
10
15
20
Time (h)
Calibrated pattern 3
Real pattern
Calibrated Pattern
95% CI
Pattern value
0.05
0.04
0.03
0.02
0.01
0
0
5
10
15
20
Time (h)
Calibrated pattern 4
Real pattern
Calibrated Pattern
95% CI
Pattern value
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0
5
10
15
20
Time (h)

# 5. Frequently Asked Questions

**Question:** *When I load a network in the main interface, I get the following error:*

```
>> HydroSimulator
??? Error using ==> matlabENgetflowunits at 6
Error Epanet -> System Error 102: no network data available.

Error in ==> HSimCarregarParametres at 18
    paramConfig(1)=double(matlabENgetflowunits());

Error in ==> HydroSimulator>CarregarModel_Callback at 121
configParam=HSimCarregarParametres(model);

Error in ==> gui_mainfcn at 96
        feval(varargin{:});

Error in ==> HydroSimulator at 48
    gui_mainfcn(gui_State, varargin{:});

Error in ==> @(hObject,eventdata)HydroSimulator('CarregarModel_Callback',hObject,eventdata,guidata(hObject))


??? Error while evaluating uicontrol Callback
```

**Answer:** *Probably you are using an alternative network not included in the website, and you have generated the .inp file from EPANET2. This newest version of EPANET includes in the .inp file three lines in capital letters in the [OPTIONS] section:*

```
[OPTIONS]
 Units                LPS
 Headloss             H-W
 Specific Gravity     1.0
 Viscosity            0.0000011
 Trials               100
 Accuracy             0.001
 CHECKFREQ            2
 MAXCHECK             10
 DAMPLIMIT            0
 Unbalanced           Continue 10
 Pattern              1
```

*Open the .inp file with the notepad and delete these three lines in order to solve the problem.*

**Question:** *When opening an extra module, I get the following error:*

```
??? Error using ==> run
Too many input arguments.

Error in ==> HydroSimulator>ExtraModule_Callback at 237
    eval(['run ' PathName FileName]);

Error in ==> gui_mainfcn at 96
        feval(varargin{:});

Error in ==> HydroSimulator at 48
    gui_mainfcn(gui_State, varargin{:});

Error in ==>
@(hObject,eventdata)HydroSimulator('ExtraModule_Callback',hObject,eventdata,guidata(hObject))


??? Error while evaluating uicontrol Callback
```

**Answer:** *Check that the path to the extra module .m file does not contain any empty space.*