HydroSimulator

v4.2

Quick Guide

Developer: Gerard Sanz Estapé <u>gerard.sanz@upc.edu</u>

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	3
2. Use of HydroSimulator	4
2.1. Starting the application	4
3. Working with extra modules: Pressure control	9
3.1.1. Monovariable control with PRV1	10
3.1.2. Monovariable control with PRV2	12
3.1.3. Split-range control	13
3.1.4. Monovariable control of mean value	13
3.1.5. Multivariable control	14
4. Calibration Module	16
4.1. Generating an scenario	18
4.2. Running the calibration algorithm	21
5. Frequently Asked Questions	. 24

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:

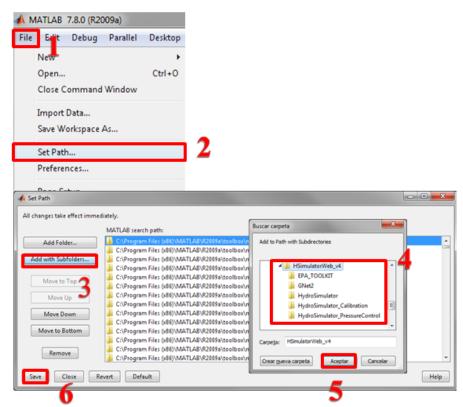
- HydroSimulator_PressureControl
- HydroSimulator_Calibration

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:

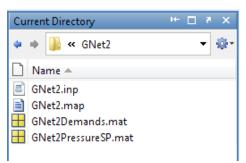


2. Use of HydroSimulator

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

2.1. Starting the application

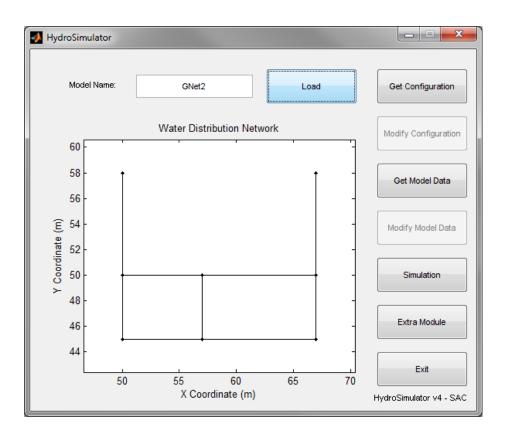
The working folder is always the one where the network data is located. In this case this folder is called: GNet2.



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

🛃 HydroSimulator		X
Model Name:	Load	Get Configuration
		Modify Configuration
		Get Model Data
		Modify Model Data
		Simulation
		Extra Module
		Exit
	 	HydroSimulator v4 - SAC

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



Some information about the network is stored in the workspace:

Workspace			→1 🗖	× s
1 🖬 🐿 🖬 🕷	▼ Stac <u>k</u> : Base ▼			
Name 🔺	Value	Min	Max	
🕂 configParam 🖸	[5,100,0.0010,0.0100,0 <11x2 cell>	0	86400	
🖸 labelsNodes	<10x2 cell>			
<u>ab</u> model <mark>⊞</mark> modelData	'GNet2' <1x12 int32>	0	11	

- 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:

				- Time variables-			
flow units:	LPS	Emitter Exponent:	0.5	Duration:	86400	ReportStep:	
	LPS		0.5	HydStep:	600	ReportStart:	
umber of trials:	100	Demand multiplier:	1	QualStep:		RuleStep:	
				PatternStep:	600	Statistic:	
ccurary:	0.001	Tolerance:	0.01	PatternStart:	0	Periods:	
parameter.		h parameter, just right click α					

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

				- Time variables-			
ow units:	LPS	Emitter Exponent:	0.5	Duration:	86400	ReportStep:	
	LPS		0.5	HydStep:	600	ReportStart:	
mber of trials:	100	Demand multiplier:	1	QualStep:		RuleStep:	
				PatternStep:	600	Statistic:	
ccurary:	0.001	Tolerance:	0.01	PatternStart:	0	Periods:	
that one parcel of	water is essenti 1 for all types of	ratter quality level below whi ally the same as another. quality analyses (chemical, ε ed in percent)).					

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

HydroSimulator v1.0 - SAC Quantity of nodes: 10	Gu	ntity of links: 11	_	Get more information about	
Node types		Link types		Information about nodes	
Quantity of tanks:	0	Quantity of pipes with check valve:	0	Information about links	
Quantity of reservoirs:	2	Quantity of pipes:	9	Information about patterns	
Quantity of junctions:	8	Quantity of pumps:	0	Generate	
		Quantity of valves:	2		
Quantity of patterns: 1	Quantity of	curves: 0 Quantity of simples	s controls: 0	Close	

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

Workspace		→ I 🗖	× 5
1 🖻 🔁 🖏 🗛 🔤 -	Stac <u>k</u> : Base ▼		
Name 🔺	Value	Min	Max
Η configParam	[5,100,0.0010,0.0100,0	0	86400
🖸 dadesLinks	<12x9 cell>		
🖸 dadesNodes	<11x8 cell>		
🚺 dadesPatterns	<2x4 cell>		
🖸 labelsLinks	<11x2 cell>		
🖸 labelsNodes	<10x2 cell>		
ab model	'GNet2'		
Η modelData	<1x12 int32>	0	11
ab radioButton	'Nodes'		

- 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:

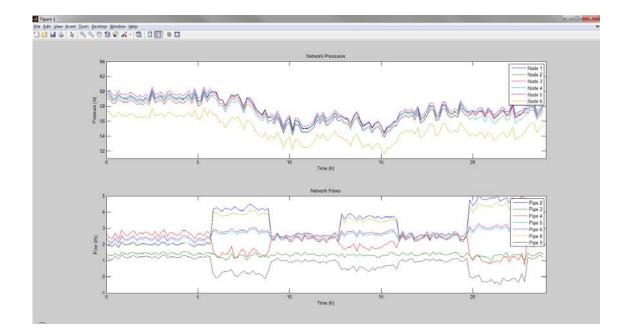
```
>> nodes=labelsNodes(1:6,1)
>> pipes=labelsLinks(1:7,1)
```

Define the simulation parameters as shown in the next figure:

📣 HydroSin	HydroSimulator						
Input Dat	a						
PRV1:	PRV1	SP1: [SP1				
PRV2:	PRV2	SP2:	SP2				
	Demands: de	mandesS	oroll				
Output D	Output Data						
Pres	sure Sensors Labels:		nodes				
F	flow Sensors Labels:		pipes				
∣ Time Pan	Time Parameters						
Initial Sa	mple: 1	Run Simulation					
Final Sa	mple: 144	Cancel					

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:

📣 Select File to (Open		×
Bus <u>c</u> ar en:	Hydro Simulator_PressureControl	⇐ 🗈 💣 💷 ◄	
(Has)	Nombre	Fecha de modifica	Tipo
Sitios recientes	🖺 HSimControlador.m	15/04/2013 10:34	MATLAB
Sitios recientes	HSimControladorMultivariable.m	17/01/2013 13:34	MATLAB
	HSimControlPressio.fig	15/04/2013 11:32	MATLAB
Escritorio	HSimControlPressio.m	15/04/2013 11:28	MATLAB
	🖺 HSimObtenirCota.m	18/10/2012 11:49	MATLAB
633	🖺 HSimObtenirNodesContPRV.m	17/01/2013 12:50	MATLAB
Bibliotecas			
Equipo			
Red			
	•		Þ
	Nombre: HSimControlPressio.m	•	<u>A</u> brir
	Tipo: MATLAB files	•	Cancelar

The following module is shown:

MydroSimulator	
- Simulation Data	Control Parameters
Demand matrix:	Controlled Node:
PRV 1:	Pressure Set Point:
Vector with pressure SP 1:	Control mean value of defined nodes
PRV 2:	Control with PRV1
Vector with pressure SP 2:	KP 1:
Initial Sample:	Control with PRV2
Final Sample:	KP 2:
	Anti wind-up
Multivariable Control	
Start	Close

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

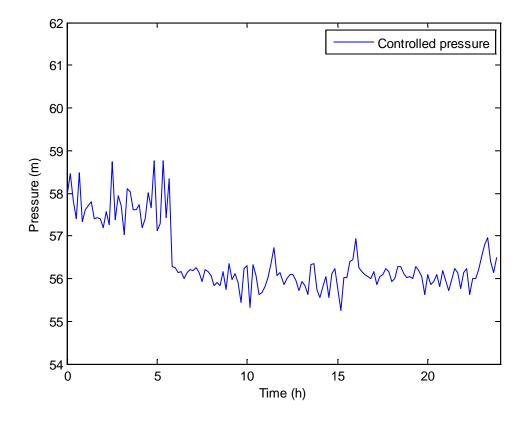
Set parameters as shown in next figure and Start the pressure control algorithm:

📣 HydroSimulator			
Simulation Data		Control Parameters	
Demand matrix:	demandesSoroll	Controlled Node:	5
PRV 1:	PRV1	Pressure Set Point:	56
Vector with pressure SP 1:		Control mean v	value of defined nodes
PRV 2:	PRV2	Control with PRV1	
Vector with pressure SP 2:	SP2		1
Initial Sample:	1	Control with PRV2	
Final Sample:	144	Ki 2:	
		🗸 Anti	wind-up
Multivariable Control			
	Start	Close	
Į	j		

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:

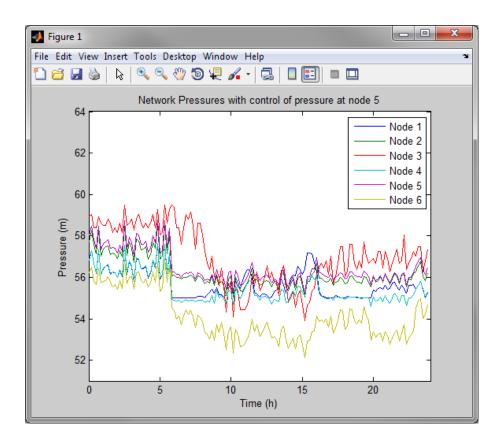


It can be seen that most time pressure is controlled at the desired set point. SP1Control can be used in the simulation module as Set Point for PRV1 in order to observe the resulting pressures in the rest of the nodes. Close the Extra Module window. Open the simulation module. Set parameters as shown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

🛃 HydroSimulator						
Input Dat	a					
PRV1:	PRV1	SP1: [SP1Control			
PRV2:	PRV2	SP2:	SP2			
	Demands: de	mandesS	Soroll			
Pres	Output Data Pressure Sensors Labels: nodes Flow Sensors Labels: []					
Time Para	Time Parameters					
Final Sa			Cancel			

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')
```

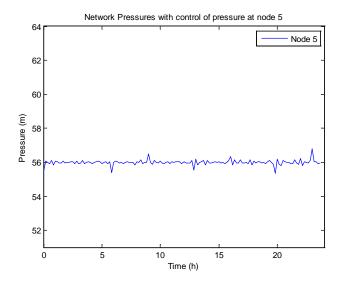


3.1.2. Monovariable control with PRV2

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:



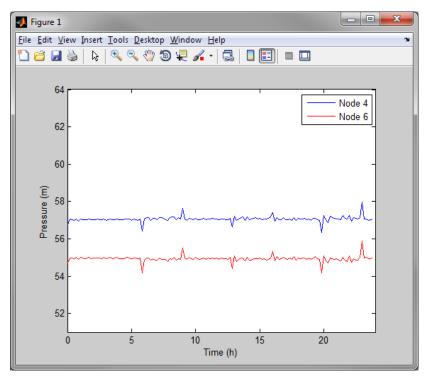
3.1.4. Monovariable control of mean value

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'	esC={'4' '6'}
--------------------	---------------

Simulation Data		Control Parameters	
Demand matrix:	demandesSoroll	Controlled Node:	nodesC
PRV 1:	PRV1	Pressure Set Point:	56
Vector with pressure SP 1:		🔽 Control mean valu	e of defined nodes
PRV 2:	PRV2	☑ Control with PRV1	
Vector with pressure SP 2:	SP2	Ki 1:	1
Initial Sample:	1	Control with PRV2	
Final Sample:	144	Ki 2:	1
Multivariable Control		Z Anti wir	id-up

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':



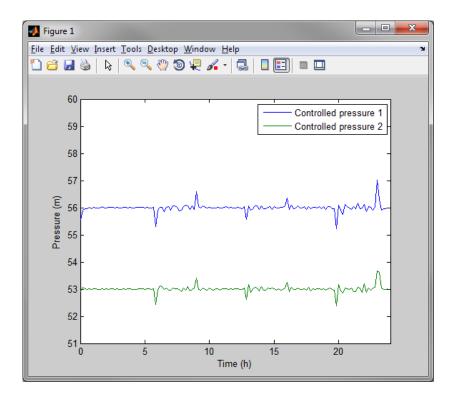
3.1.5. Multivariable control

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

🙏 HydroSimulator						X
Simulation Data		īI	— Control Parameters ———			
Demand matrix: de	mandesSoroll		Controlled Node:			
PRV 1:	PRV1		Pressure Set Point:			
Vector with pressure SP 1:			Control mean v	alue of defi	ned nodes	
PRV 2:	PRV2		✓ Control with PRV1			
Vector with pressure SP 2:			Ki 1:		1	
Initial Sample:	1		\checkmark Control with PRV2			
Final Sample:	144		Ki 2:		1	
Multivariable Control			🔽 Anti y	wind-up		
Muttivariable Control						
Node controlled by PRV 1:	4		Node controlled by PRV 2:		6	
Pressure Set Point 1:	56		Pressure Set Point 2:		53	
Sta	rt		Close			

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:

Select File to C	Open		×
Bus <u>c</u> ar en:	HydroSimulator_Calibration	← 🗈 📸 🖛	
œ.	Nombre	Fecha de modifica	Тіро
Sitios recientes	🖺 getData.m	09/05/2013 16:02	MATLAB I
	📋 getDemandMatrixModel.m	09/05/2013 16:02	MATLAB I
	📋 getIndexes.m	09/05/2013 16:02	MATLABI
Escritorio	📋 getMatrixModel.m	09/05/2013 16:02	MATLABI
	📋 getNonLinearMatrix.m	09/05/2013 16:19	MATLABI
	HSimCalibration.fig	10/05/2013 12:43	MATLABI
Bibliotecas	HSimCalibration.m	17/05/2013 13:43	MATLABI
	HSimPatternCalibration.m	17/05/2013 10:28	MATLABI
Equipo	District plotBoundaries.m	10/05/2013 13:32	MATLABI
Equipo	standardUncertaintyCalculation.m	17/04/2013 17:02	MATLABI
	SubplotPosition.m	10/05/2013 9:57	MATLABI
Red			
			۱.
	Nombre: HSimCalibration.m	-	<u>A</u> brir
	Tipo: MATLAB files	•	Cancelar

The following window appears:

Algorithm Parameters	Calibration Data
Initial sample:	Measured heads:
Final sample:	Heads std. dev.:
Correction step:	Measured flows:
Calibration horizon:	Flows std. dev.:
Data periodicity:	Total consumption:
Generation of Matrix Model	Plot Parameters
Base demands and type of contract:	Rows:
Not included nodes:	Columns:
Not included links:	Real patterns available
Generate Matrix Model	Real patterns:
Run Calibration Process	Cancel
	Initial sample: Final sample: Correction step: Calibration horizon: Data periodicity: Generation of Matrix Model Base demands and type of contract: Not included nodes: Not included links:

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:
 - \circ 1st column: Label of the demand nodes.
 - \circ 2nd column: Base demand of the demand nodes.
 - \circ 3rd 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):

🚺 HydroSim	ulator		X
- Input Dat	a		
PRV1:	PRV1	SP1:	SP1
PRV2:	PRV2	SP2:	SP2
	Demands:	demands	\$
	ata sure Sensors Labels: 'low Sensors Labels:		ureSensorLabels vSensorLabels
Time Para	ameters		
Initial Sa	mple: 1		Run Simulation
Final Sa	mple: 120		Cancel

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:

HydroSimulator		Contractor Television	
Quantity of nodes: 16 Node types Quantity of tanks: Quantity of reservoirs: Quantity of junctions:	Quantity 0 2 14	of links: 19 Link types Guantity of pipes with check valve: Quantity of pipes: 1 Quantity of pumps: 0 Quantity of valves: 2	7 O Information about patterns
Quantity of patterns:	1 Quantity of cur	ves: 0 Quantity of simples co	ntrols: 0 Ciose

Now, find the elevations of each measured node:

-1	Variable Ed	litor - dadesNo	des				
	1 🔏 🖻	🛍 🗳 🛛	- 4	Stack:	Base 🔻		
()	dadesNode	es <17x8 <u>cell</u> >					
	1	2	3	4	5	6	
1	'Label'	'Index'	'Base dema	'Pattern'	'Emitter coe	'Elevation'	'Initia
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
10	ALIVD11	11	0	-	0	70	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:

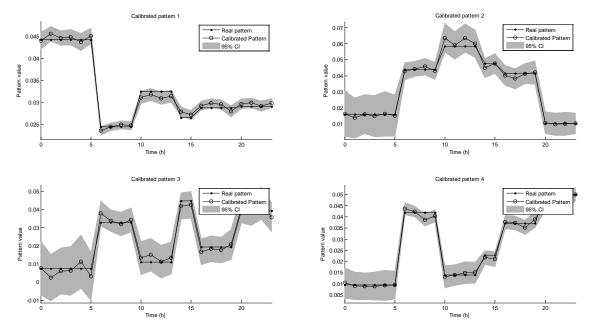
HydroSimulator	and the second se	
- Model Parameters	Algorithm Parameters	Calibration Data
PRV1 ID:	Initial sample:	Measured heads:
SP1:	Final sample:	Heads std. dev.:
	Correction step:	Measured flows:
PRV2 ID:	Calibration horizon:	Flows std. dev.:
SP2:	Data periodicity:	Total consumption:
Pressure sensors labels:	Generation of Matrix Model	Plot Parameters
Flow sensors labels:	Base demands and type of contract:	Rows:
- Termination Criterion	Not included nodes: nodesNo	Columns:
	Not included links: linksNo	Real patterns available
Minimum iterations:	Generate Matrix Model	
Maximum iterations:		
Minimum variation:	Run Calibration Process	Cancel

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

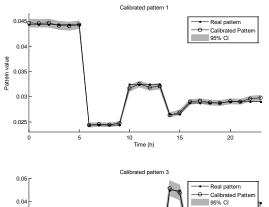
Model Parameters ———		Algorithm Parameters		Calibration Data ——	
PRV1 ID:	PRV1	Initial sample:	1	Measured heads:	Heads
SP1:	SP1	Final sample:	24	Heads std. dev.:	0.1
		Correction step:	0.1	Measured flows:	0
PRV2 ID:	PRV2	Calibration horizon:	1	Flows std. dev.:	0
SP2:	SP2	Data periodicity:	24	Total consumption:	totalDemand
Pressure sensors labels:	PressureSensorLab	Generation of Matrix Model		Plot Parameters	
Flow sensors labels:		Base demands and type of contract:	cons	Rows:	2
Termination Criterion		Not included nodes:	nodesNo	Columns:	2
		Not included links:	linksNo	🔽 Real patter	ns available
Minimum iterations:	20	Generate Matr	iv Model	Real patterns:	realPatterns
Maximum iterations:	50	Generate Matr	IX MODEL		
Minimum variation:	0	Run Calibration	Process	Car	ncel

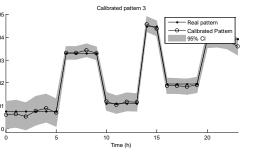
Model Parameters		Algorithm Parameters —		Calibration Data ——	
PRV1 ID:	PRV1	Initial sample:	1	Measured heads:	Heads
SP1:	SP1	Final sample:	24	Heads std. dev.:	0.1
		Correction step:	0.1	Measured flows:	0
PRV2 ID:	PRV2	Calibration horizon:	1	Flows std. dev.:	0
SP2:	SP2	Calibration horizon size:	24	Total consumption:	totalDemand
Pressure sensors labels:	PressureSensorLab	Generation of Matrix Model		Plot Parameters	
Flow sensors labels:		Base demands and type of contract:	cons	Rows:	2
Termination Criterion		Not included nodes:	nodesNo	Columns:	2
		Not included links:	linksNo	📝 Real patter	ns available
Minimum iterations:	20	Generate Mat	iv Madal	Real patterns:	realPatterns
Maximum iterations:	50	Generate Mati	IX WOULD		
Minimum variation:	0	Run Calibration	Process	Car	

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:

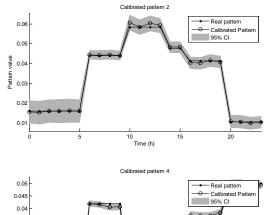


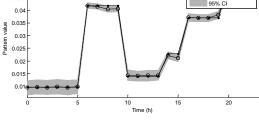


- 0.03 - Aaltem Aaltem Juice Herminian Aaltem

0.01

0 0





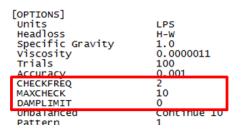
5. Frequently Asked Questions

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 ==> 8(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:



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 ==> <u>HydroSimulator>ExtraModule_Callback at 237</u>
    eval(['run ' PathName FileName]);
Error in ==> <u>gui_mainfcn at 96</u>
    feval(varargin(:));
Error in ==> <u>HydroSimulator at 48</u>
    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.