Automating a Hydraulic Transient Model to Enable Efficient Sensitivity Analysis and Calibration

Hailiang Shen, Juraj Cunderlik and Chris Hunter

GHD, Waterloo, Ontario

Abstract

An in-house tool was developed to enable efficient sensitivity analysis and calibration study for Bentley's HAMMER steady state and transient models. The tool was created by coupling Bentley WaterObjects.NET, programming language C#, and genetic library in AForge.NET. The tool supports sensitivity analysis for the Hazen–Williams c coefficient for a steady state model, and for the inertia of pump plus motor and pipe wave speed for a transient model. The tool also enables calibration for field testing pressures. The tool was tested and applied in a transient study of a sewer system pumping station upgrade. The tool identified sensitive parameters of the case study model and achieved much better accuracy than the initial model setup after calibration.

1 Introduction

Hydraulic transient phenomena in closed pipeline systems occur when flow experiences sudden changes. These changes can be caused by pump stop, pump start (pipe filling), rapid inline valve closure, pipe burst, rapid demand change, and other factors. As a result, significant pressure fluctuations around the initial steady state can be observed. The consequences of transient phenomena could vary from vibration noise near the pumping station to pipeline breakage, damage to pumping stations, or backflow and the intrusion of dirty water (Boulos et al. 2005). Factors such as the elasticity of pipe boundaries, fluid compressibility, and pipe wall and minor frictions can dampen the generated transient pressures to a limited extent. Surge devices (e.g. surge valves or tanks, or air valves) are often required to mitigate destructive transient forces. For the design of a proposed feedermain in a water distribution system or a forcemain in a sewer system, it is therefore crucial to analyse the impact of transient pressures to ensure that the integrity of the pipeline is not compromised.

Hydraulic transients in a closed pipeline system are governed by one-dimensional continuity and momentum equations. Numeric solutions have been proposed, including finite difference, finite volume, finite element, the method of characteristics (MOC), and the wave plan method (WM) (Wood et al. 2005). Sabbagh-Yazdi et al. (2007) discussed the strength and weakness of each method.

Bentley's HAMMER software (Bentley 2015) can model almost all transient sources, surge mitigation devices, and their associated operations. HAMMER provides a graphical user interface (GUI) for model creation, data management, model setup and run, and post-processing. In addition to the built-in GUI, HAMMER also supports user customization with Bentley's WaterObjects.NET. WaterObjects.NET uses an application programming

interface (API) for model data input–output (IO), and model run control, thus enabling multiple runs of the numerical transient engine for various applications such as parameters sensitivity analysis, calibration, air valve sizing, optimal timing for valve closure, and pump scheduled stop or start.

Researchers have applied optimization in designing transient protection devices. Jung and Karney (2006) used genetic algorithm and particle swarm optimization to optimally design transient protection devices. Jung et al. (2011) formulated the design of a water distribution system as a two objective optimization problem, where the objectives are to minimize the cost and minimize the likelihood of transient events. Skulovich et al. (2015) applied piecewise mixed integer programming for optimal sizing of a closed surge tank in a water distribution systems.

The steady state model provides initial conditions (e.g. flow velocities and pressures) for the transient model to start with. Extensive research has been conducted to calibrate steady state models for pressurized systems (e.g. water distribution systems). Shen and McBean (2010) calibrated a small water distribution system with a genetic algorithm, and Qi et al. (2016) used particle swarm optimization for water distribution system optimization.

However few studies are found in automatic sensitivity analysis and calibration for HAMMER models. Sensitivity analysis can identify sensitive parameters which may deserve more data collection and further study, and calibration can provide more confidence that a model is behaving properly. Currently HAMMER does not support automatic sensitivity and calibration. In this study an in-house tool was developed to fill this gap.

2 Methodology

2.1 Governing Equations and WaterObjects.NET

Hydraulic transient phenomena in a closed pipe system are governed by two governing equations, as presented in Equations 1 and 2 (Wylie and Streeter 1978; Tullis 1989).

$$g \frac{\partial H}{\partial x} + \frac{dV}{dt} + \frac{fV|V|}{2D} = 0 \quad (1)$$

$$\frac{\partial H}{\partial t} + \frac{a^2}{g} \frac{\partial V}{\partial x} = 0 \quad (2)$$

where:

- H = piezometric head, (m)
- V = flow velocity, (m/s)
- f = friction factor in the Darcy–Weisbach equation,
- D = pipe diameter, (m)
- a = wave speed, (m/s)
- g = gravity acceleration, (m/s²)
- x = coordinate along pipe direction, (m) and
- t = time (s).

Using the method of characteristics, the two partial differential equations (1) and (2) can be replaced by two pairs of ordinary differential equations, identified as C+ and C– characteristics, as expressed in equations (3) and (4) (Wylie and Streeter 1978):

$$\text{C+ equation: } \frac{g}{a} \frac{dH}{dt} + \frac{dV}{dt} + \frac{fV|V|}{2D} = 0, \text{ for } \frac{dx}{dt} = +a \quad (3)$$

$$\text{C– equation: } \frac{g}{a} \frac{dH}{dt} - \frac{dV}{dt} - \frac{fV|V|}{2D} = 0, \text{ for } \frac{dx}{dt} = -a \quad (4)$$

HAMMER solves Equations 3 and 4 as well as various boundary conditions at surge tanks, surge valves, check valves, air valves and air chambers. It uses a discrete vapour cavity model (DVCM) to model the formation and collapse of vapour and air pockets, which can often cause secondary surges.

WaterObjects.NET is the application programming interface (API) for the Bentley Municipal Products Group. It consists of a family of .NET assemblies that host a set of interfaces and classes. It allows user to customize HAMMER in numerous ways, except for modifying the algorithm for steady state/extended period simulation (EPS) numerical engine and transient engine. A user can implement user defined logic, or use all HAMMER functionalities as a service (i.e. without opening the HAMMER GUI).

2.2 Sensitivity Analysis for Steady State and Transient Models

Sensitivity analysis is a process performed for the identification of the parameters that most affect a model's output. Sensitive parameters require more accurate estimation and, therefore,

more data collection in order to improve model reliability. The proposed process of sensitivity analysis for Bentley HAMMER is illustrated in Figure 1. In HAMMER the user can create multiple scenarios to study the sensitivity of a model to a particular parameter. However, as the resulting number of scenarios increases, the project may quickly become unmanageable. For example, a model with too many scenarios may consume hundreds of mega-byte disk volume, and slow down runs for specific scenarios.

Figure 1 Sensitivity analysis for HAMMER model.

A HAMMER model run consists of sequentially running the steady state and transient numerical engines. The steady state engine can be chosen as EPANET or WaterGEMS (Bentley 2015). The steady state engine outputs (e.g. pipe flows and node pressures) are used as the initial conditions for the transient numerical engine. The transient numerical engine then generates system transient responses (surge pressures, flow rates, and cavity open and close times) for model elements (e.g. junctions, pumps, pipes, air valves, surge valves). An automatic procedure is proposed for sensitivity analysis of three HAMMER input parameters: the Hazen–Williams c coefficient, inertia of pump plus motor, and pipe wave speed. The procedure does not generate new scenarios and directly updates parameters for the active scenario. Doing such avoids increasing model file size and model execution time.

Sensitivity analysis for the Hazen–Williams c coefficient can be performed for either the steady state results (e.g. steady state pressure) or the transient results (e.g. surge pressure), depending on the availability of field data. The impact of the Hazen–Williams c coefficient on the steady state results could be analysed if steady state data were collected. In the case that no steady state data were collected, the Hazen–Williams c coefficient could be analysed for surge results.

Sensitivity analysis is performed on a one-by-one basis for a baseline model. A baseline model includes the parameters set up by using the best estimate from literature and empirical equations. Then the sensitivity of a single parameter is analysed while other parameters are kept at their base values. This sensitivity analysis method was chosen over other methods (e.g. the Sobol sensitivity analysis discussed in Sobol 2001) because it is simple to perform and requires less setup time. However, the Sobol sensitivity analysis algorithm will be incorporated into future research.

Automatic sensitivity analysis for the HAMMER model requires input parameter updates and nodal pressure retrieval. The biggest challenge in programming WaterObjects.NET is identifying the correct interfaces to work with for each task. The commands for parameter updates are listed in Table 1, and the

C# code snippet for retrieving nodal surge pressure time series is provided in the Appendix to this paper.

Table 1 WaterObjects.NET commands for parameter updates.

Task	Command for parameter update
Open current HAMMER project	ParentFormModel.CurrentProject
Get domain dataset	CurrentProject. IdahoDomainDataSet
Get pipe element manager	IdahoDomainDataSet. IdahoPipeElementManager
Get the Hazen–Williams c coefficient	IdahoPipeElementManager. Physical_HazenWilliamsCField
Set the Hazen–Williams c coefficient	(Physical_HazenWilliamsCField as IEditField).SetValue()
Get inertia of pump plus motor	IdahoPumpDefinitionElementManager. SupportElementField (StandardFieldName. IdahoPumpDefinition_PumpAndMotorInertia)
Set inertia of pump plus motor	IdahoPumpDefinitionElementManager. SupportElementField (StandardFieldName. IdahoPumpDefinition_PumpAndMotorInertia) as IEditField).SetValue()
Get wave speed	IdahoPipeElementManager. DomainElementField(StandardFieldName. HAMMER_WaveSpeed, StandardAlternativeName.HAMMER)
Set wave speed	(SupportElementField (StandardFieldName. IdahoPumpDefinition_PumpAndMotorInertia) as IEditField).SetValue()

2.3 Calibration Study for HAMMER Model

Calibration is to match model responses with field testing data, by updating model parameters until the model matches calibration data within pre-specified margins (especially sensitive ones). The HAMMER model calibration problem was formulated as an optimization problem. The objective function is defined as the summation of the squared error defined in Equation (5).

$$F = \sum (P_{M_i} - P_{O_i})^2 \quad (5)$$

where:

F = objective function, to be minimized,

P_{M_i} = pressure at target node at HAMMER reporting time t ,

P_{O_i} = field testing pressure at target node at reporting time t (if no field data are available at time t , P_{O_i} is linearly interpolated),

For steady state calibration, the Hazen–Williams c coefficient is subjected to $C \in (C_{min}, C_{max})$, and for transient calibration inertia and wave speed are subjected to $I \in (I_{min}, I_{max})$ and $a \in (a_{min}, a_{max})$. The subscripts min and max are respectively the minimum and maximum of the associated parameter.

The process of calibration study is illustrated in Figure 2. The process is similar to Figure 1 except for the additional step *Calculate optimization objective*. Although an optimization algorithm based on genetic algorithms (GA) is supported in other Bentley products (e.g. WaterGEMS and WaterCAD), it is not designed to calibrate the HAMMER transient model. Therefore an external optimization library is needed for transient model calibration.

Figure 2 HAMMER model calibration process.

GA has been applied to solve model based optimization problems (e.g. Shen and McBean 2010), and was selected to solve this optimization problem as formulated in Equation 5. GA was selected for it is easy to implement and interpret. However other optimization algorithms (e.g. dynamic programming, gradient based methods, simulated annealing, particle swarm) might also perform equally well, and will be explored in solving this optimization problem in future research. GA is based on the concept of the evolution of generations. Each generation includes a user defined number of chromosomes (or individuals). Each chromosome has its fitness defined for specific problem. Three basic operations, *selection*, *mutate* and *crossover*, are performed on these individuals to generate better child chromosomes for the next generation. The evolution continues until the number of generations reaches a user predefined value or the best chromosome fitness does not improve between the current and previous generations.

Since WaterObjects.NET is written purely in .NET, a GA library written in .NET has advantages over one written in other languages (e.g. C/C++, FORTRAN) in avoiding data and functions interoperability with other native code. The AForge.NET (Kirillov 2013) Genetic library was selected. It is written purely in C#, which

enables seamless interaction with WaterObjects.NET. Also it is open source, and can be enhanced or incorporated into a user's program. The current implementation of Genetic library does not include an online archive to record the evaluated chromosomes. Duplicate chromosome evaluation can be very computationally expensive (e.g. one HAMMER run could take minutes or even hours depending on model size, time step, simulation duration, and the number and frequency of reporting variables). Accordingly, Genetic was improved by archiving the evaluated individuals within and across generations.

Herein, GA fitness function was defined as the reciprocal of F in Equation 5. An individual in a GA population is better than another if its GA fitness is larger. C was the single parameter for steady state model calibration, and was represented as BinaryChromosome, which is the only chromosome type in the Genetic library supporting one-dimensional optimization. Pump and motor inertia and wave speed multipliers were calibrated for the transient model. Each new inertia and wave speed value was generated by multiplying its base value with some multiplier. The multipliers were generated between parameter low and high bounds with a step change, and were represented as ShortArray-Chromosome in the Genetic library.

2.4 Sensitivity Analysis and Calibration Tool

A tool was developed to implement the utilities for HAMMER sensitivity analysis and calibration for both steady state and transient models. The in-house tool development environment was Visual C# 2013 Premium, .NET4.0, and HAMMER version 08.11.06.58 (64-bit, the latest version at the time of this study).

The tool works with a baseline HAMMER model. The baseline model contains the base values for all parameters to be analysed. For example the Hazen–Williams c for combinations of pipe material and age could be obtained from literature. The wave speed of water is a function of pipe material, pipe wall thickness and pipe joint type, and could be calculated with equation 6, which is valid for thin walled pipelines ($D/e > 40$) (Haested Methods 2003).

$$a = \sqrt{\frac{\frac{E_v}{\rho}}{1 + \frac{DE_v}{eE}\Psi}} \quad (6)$$

where:

- a = characteristic wave speed of water (m/s),
- E_v = bulk modulus of elasticity for water (Pa),
- ρ = liquid density (kg/m³),
- D = diameter (mm),
- e = wall thickness (mm),
- E = Young's modulus for pipe material (Pa), and
- Ψ = pipeline support factor.

The inertia of pump and motor could be estimated with Equation 7:

$$I = 0.03768 \times \left(\frac{P}{N^3}\right)^{0.9556} + 0.0043 \times \left(\frac{P}{N}\right)^{1.48} \quad (7)$$

where:

- P = shaft power (kW), and
- N = rotational speed (rpm ÷ 1 000).

The first and second terms calculate the inertia for pump and motor respectively. All coefficients were calculated by regressing pump and motor data from different manufacturers. The two pump coefficients 0.037 68 and 0.955 6 were calculated from 284 data points, and they are changed to 0.034 07 and 0.844 when fitting 28 data points from a manufacturer.

However, the three estimated parameter values have uncertainty. For example, the Hazen–Williams c coefficient for concrete pipe was given as 100 to 150 by Connell (2001). The inertia could be 50% to 200% of the calculated value from Equation 7, and the wave speed for concrete pipe could vary between 350 m/s and 1 165 m/s (Thorley 2004).

The baseline model should not contain any runtime errors. After opening a base model, an information window is displayed (as shown in Figure 3) which summarizes the numbers of various elements for the active scenario in the base model. The element naming and its numbers help to understand WaterObjects.NET (e.g. the terminology of each element manager), and validate that the HAMMER model is loaded correctly.

Element	Number
BaseLinkElementManager	60
IdahoPipeElementManager	60
BaseIdahoNodeElementManager	54
IdahoDemandNodeElementManager	47
IdahoFireFlowNodeManager	47
IdahoJunctionElementManager	47
BaseHammerNode	6
AirValveElementManager	3
BaseDirectedNodeElementManager	2
BasePump	2
StandardPumpElementManager	2
SavSrvElementManager	2
IdahoReservoirElementManager	1
BaseOnifice	1
DischargeToAtmosphere	1

Figure 3 Baseline model information window.

The tool enables sensitivity analysis with the utility shown in Figure 4 and Figure 5 for the steady state model and the transient model respectively. Target Node Id defines the model node(s)

that will be targeted for sensitivity analysis. Currently the utility implements the sensitivity analysis for three parameters: the Hazen-Williams c coefficient for the steady state model, inertia of pump plus motor and wave speed for the transient model, but can later be extended to any other WaterObjects.NET supported parameters (e.g. pump rotational speed, air valve inlet and outlet sizes, valve closure timing, pipe diameter, and many others).

Figure 4 Utility for steady state parameter sensitivity analysis.

Figure 5 Utility for transient parameter sensitivity analysis.

The description of each input parameter is listed in Table 2 below. The utilities evaluate sensitivity for each parameter one at a time, and list the steady state pressure, and minimum, maximum and average surge pressures at the Target Node Id on the grid in Figure 4 and Figure 5 respectively.

The tool enables model calibration with the utilities shown in Figure 6 and Figure 7. The optimal multipliers and the corresponding fitness of current population are displayed in real time in the tables in Figure 6 and Figure 7. As evolution continues, the fitness of the best chromosome is expected to increase.

Table 2 Supported parameters for sensitivity analysis.

Input	Description
***High	The high end of the multiplier for parameter ***
***Low	The low end of the multiplier for parameter ***
***Step	The step change of the multiplier for parameter ***

Note: *** is the Hazen-Williams c coefficient, inertia of pump and motor, or wave speed

Figure 6 Utility for steady state model calibration.

Figure 7 Utility for transient model calibration.

Inputs for the three parameters have the same meanings as in Table 2; other GA and file related inputs are given in Table 3.

Table 3 Parameters for calibration study.

Input	Description
CalibrationElementId	The field testing data are collected for these nodes
CalibrationFileName	The field testing data are stored in this file
LogFileName	Log generated is stored in this file
PopulationSize	The number of GA populations in a generation
GenerationSize	The number of generations
CrossoverRate	GA crossover rate. The default value of the Genetic library is 0.75
MutationRate	GA mutation rate. The default value of the Genetic library is 0.1

3 Case Study

The sensitivity analysis and calibration tool was applied to a pumping station upgrade project. A HAMMER model was created from as-built drawings for the forcemain profile and pumping stations, specifications for pumps, surge valves and air valves. A baseline model sketch of the pumping station is depicted in Figure 8. Wastewater is pumped from a wet well and is delivered through a forcemain to a manhole. The forcemain is approximately 4 km, and the number of model elements are listed in Table 4. The forcemain profile is shown on Figure 9.

Figure 8 HAMMER model layout around the pumping station.

Table 4 System dimension and parameter values.

Item	Value
# of pipes	60
# of junctions	47
# of surge valves	2
# of air valves	3
# of reservoirs	1
# of pumps	2 (only PMP-1 was active in the case study)
# of discharge to atmosphere	1 (representing the downstream discharge manhole)
Pipe length	5 – 354 m
Pipe diameter	400 – 450 mm
Pipe material	Concrete
Air valve sizes	50 mm, 75 mm, 100 mm
Surge valve size	150 mm

Figure 9 Forcemain system profile.

The base values and their associated uncertainty range for the Hazen–Williams c coefficient, wave speed, and inertia of pump and motor are listed in Table 5. For wave speed calculation, pipe thickness and joint type were not available, so Equation 6 could not be used. Instead the middle value of the uncertainty range of wave speed for concrete pipe was taken as the base value. For the inertia calculation, the shaft power is 186.425 kW, and rotational speed is 1 720 rpm. The calculated inertia was 5.6 kg.m², and was rounded up to 6 kg.m² to account for the coupling effect of motor and pump. The multiplier values of a parameter were calculated as the range end value divided by base value, e.g. multipliers for the Hazen–Williams c coefficients 100 and 150 are 100/110 (i.e. 0.91) and 150/110 (i.e. 1.36).

Table 5 Base values and associated uncertainty ranges for the three parameters.

Parameter	Range	Base value	Multipliers
Hazen–Williams c coefficient	100 to 150 (Stephenson 1989)	110 (HAMMER default value for concrete pipe)	0.91 to 1.36
Wave speed	350 m/s to 1165 m/s (Thorley 2004)	750 m/s (middle value of the range)	0.47 to 1.55
Inertia of pump and motor	3 kg.m ² to 12 kg.m ² (50% to 200% of the calculated value, Thorley 2004)	6 kg.m ² (calculated with Equation 7)	0.50 to 2.00

The transient model time step controls the model result accuracy and run time. A model with large time step can lead to inaccurate model outputs, although it requires a shorter run time. A model with a small time step can significantly increase the model run time (e.g. from seconds to minutes), but will generate accurate results. For sensitivity analysis and calibration study, the transient model is required to be run many times with different sets of input parameters, so the runtime for a single model is crucial. Four different time steps were tested, and the profiles for the maximum and minimum pressures are shown in Figure 10. Three time steps (0.001 s, 0.002 s and 0.003 s generated very similar results, while time step 0.004 5 s differed from the other three for the maximum pressures between distances 0 m and 2 500 m and

for the minimum pressures around 2 000 m. Therefore time step 0.003 s was used in the baseline model. Cavitation was observed at the three air valves (Figure 9 above) for all four tested time steps.

Figure 10 Maximum and minimum transient pressures for different time steps.

High frequency pressure data were collected from pump PMP-1 (Figure 8 above) shutdown testing, which was performed as part of the pump performance test. Pressures were collected at node 0+000 with a pressure transducer (IFM: PI1693, measurement range –14.4 psi to 362.4 psi, –100 kPa to 2 500 kPa), which samples pressures data ~10 times/s.

Figure 11 below shows the sensitivity results for the three parameters: the Hazen–Williams c coefficient for the steady state model, and inertia and wave speed for the transient model. Parameters were analysed for their associated uncertainty ranges (Table 5). For the steady state model, increasing the Hazen–Williams c coefficient value from 100 to 150 decreased the pressure at node 0+000 from 59.9 psi to 54.9 psi (413 kPa to 378.5 kPa). For the transient model, increasing wave speed from 350 m/s to 1 165 m/s increased the maximum surge pressure at node 0+000 from 75.2 psi to 101.9 psi (518.5 kPa to 702.6 kPa), and decreased the minimum pressure from –0.9 psi to –6.2 psi (–6.2 kPa to –42.7 kPa); increasing inertia from 3 kg.m² to 12 kg.m² would decrease the maximum pressure from 98.7 psi to 76.3 psi (680.5 kPa to 526 kPa), and increase the minimum pressure from –6.5 psi to –1.2 psi (–44.8 kPa to –8.3 kPa).

Increasing wave speed in general enlarged the surge envelope (i.e. increased the maximum pressure and decreased the minimum pressure). Increasing inertia narrowed the surge envelope, which was expected since larger inertia makes pump spin-down slower and therefore mitigates the transient.

Figure 11 Sensitivity analysis for three parameters.

From field monitoring, the pressure at node 0+000 when the pump was running in steady state was 56 psi (386.1 kPa), which was used for steady state model calibration. The crossover and mutation rates were kept at the default values in Table 3 (above), and population and generation numbers were set to 20 and 10 respectively. Five GA runs were performed and the optimal Hazen–Williams c value and GA fitness value are listed in Table 6. The optimal value of C is 132 (corresponding multiplier 1.2), and the corresponding GA fitness is 0.99. The pipe C values were updated to 132 in the baseline model for the following transient model calibration.

Table 6 Optimal parameter values and GA fitness.

	GA run	1	2	3	4	5
Steady state	Hazen–Williams C	132	132	132	134	132
	GA fitness	0.99	0.99	0.99	0.87	0.99
Transient	Inertia	11.7	11.1	11.7	11.1	10.8
	Wave speed	615	615	615	577.5	577.5
	GA fitness	1.15E ⁻⁶	1.06E ⁻⁶	1.15E ⁻⁶	1.08E ⁻⁶	1.07E ⁻⁶

Figure 12 shows the transient calibration results. The crossover and mutation rates were kept at their default values as shown in Table 3 (above), and population and generation numbers were set to 20 and 10 respectively. The optimal inertia and wave speed were 11.7 kg.m² and 615 m/s respectively, and the corresponding fitness was 1.15E⁻⁶. Calibration significantly reduced the discrepancy between model output and field testing pressures. The discrepancy in pressure magnitude at node 0+000 was reduced.

Figure 12 HAMMER model calibration results.

4 Conclusions and Recommendations

A sensitivity analysis and calibration tool was built to implement efficient sensitivity and calibration for three parameters (the Hazen–Williams c coefficient for the steady state model, inertia of pump plus motor, and pipe wave speed for the transient model), with C#, WaterObjects.NET, and the AForge.NET Genetic library. The tool was used in a case study, which demonstrated its ability to identify sensitive parameters and so achieve better model accuracy.

The tool can be extended for optimal design including air valve sizing, optimizing locations and properties of surge devices, identifying optimal timing for valve closure, and optimizing operation for scheduled pump stops and starts.

The following conclusions and recommendations can be drawn from this study:

1. The developed in-house tool has been developed for generic use, and it can be applied to other projects for sensitivity analysis and calibration study with zero or little change to the code base;
2. The calibration data only included the pressure time history at one node outside the pumping station. The developed calibration algorithm will be further validated once more data were collected at other locations (e.g. air valves and surge valves); and
3. Multiple time steps should be run for the baseline model. The time step that is large yet generates similar results to smaller time steps could be used for transient calibration.

Calibration is needed to obtain an accurate transient model. As seen from the case study herein, calibration can greatly improve the accuracy of the model comparing with an initial model setup. Future calibration objectives could be to minimize model outputs with field testing data for pump start instead of pump shutdown, as used in this paper, to cover more system initial conditions.

Future research directions include:

1. Herein the genetic algorithm was utilized to solve the formulated optimization problem for calibra-

tion. Other optimization methods and optimization objectives will be evaluated in the future research and projects;

2. The Genetic library in AForge.NET will be further enhanced by introducing adaptive crossover and mutation rates, which may identify better individuals in shorter time; and
3. A simple sensitivity algorithm was applied to analyze the sensitivity of three parameters (the Hazen–Williams c coefficient, wave speed, and inertia). Other sensitivity analysis algorithms will be evaluated in future research and projects.

Acknowledgment

The authors would like to thank the two anonymous reviewers. Their constructive comments significantly improved the quality of this paper and the in-house built tool, and helped in shaping future research directions.

References

- Bentley. 2015. Bentley HAMMER: V8i User Guide. Exton, PA: Bentley Systems, Incorporated.
- Boulos, P. F., B. W. Karney, D. J. Wood and S. Lingireddy. 2005. "Hydraulic Transient Guidelines for Protecting Water Distribution Systems." *Journal—American Water Works Association* 97 (5): 111–24.
- Connell, D. 2001. *Hazen-Williams C-factor Assessment in an Operational Irrigation Pipeline*. Montreal: McGill University. Doctoral Thesis.
- Jung, B., P. F. Boulos and T. Altman. 2011. "Optimal Transient Network Design: A Multi-Objective Approach." *Journal—American Water Works Association* 103 (4): 118–27.
- Jung, B. and B. W. Karney. 2006. "Hydraulic Optimization of Transient Protection Devices using GA and PSO Approaches." *Journal of Water Resources Planning and Management* 132 (1): 44–52.
doi: 10.1061/(ASCE)0733-9496(2006)132:1(44).
- Kirillov, A. 2013. AForge.NET Framework 2.2.5 Documentation. <http://www.aforgenet.com/framework/documentation.html>.
- Haested Methods. 2003. *Advanced Water Distribution Modeling and Management*. Waterbury, CT: Haested Press.
- Qi, X., K. Li and W. D. Potter. 2016. "Estimation of Distribution Algorithm Enhanced Particle Swarm Optimization for Water Distribution Network Optimization." *Frontiers of Environmental Science & Engineering* 10 (2): 341–51.
doi: 10.1007/s11783-015-0776-z.
- Sabbagh-Yazdi, S. R., N. E. Mastrokakis and A. Abbasi. 2007. "Water Hammer Modeling by Godunov Type Finite Volume Method." *International Journal of Mathematics and Computers in Simulation* 4 (1): 350–5.

Shen, H. and E. McBean. 2010. "Hydraulic Calibration for a Small Water Distribution Network." In *Water Distribution Systems Analysis: Proceedings of the 12th International Conference*, edited by K. E. Lansey, C. Y. Choi, A. Ostfeld and I. L. Pepper, 436–46. Reston, VA: American Society of Civil Engineers. doi: 10.1061/41203(425)41.

Skulovich, O., R. Bent, D. Judi, L. S. Perelman and A. Ostfeld. 2015. "Piecewise Mixed Integer Programming for Optimal Sizing of Surge Control Devices in Water Distribution Systems." *Water Resources Research* 51:4391–408. doi: 10.1002/2014WR016256.

Sobol, I. M. 2001. "Global Sensitivity Indices for Nonlinear Mathematical Models and their Monte Carlo Estimates." *Mathematics and Computers in Simulation* 55 (1–3): 271–80. doi: 10.1016/S0378-4754(00)00270-6.

Stephenson, D. 1989. *Pipeline Design for Water Engineers* (3rd ed). New York, NY: Elsevier Science Publishers B.V.

Thorley, A. R. D. 2004. *Fluid Transients in Pipeline Systems*, 2nd ed. New York: American Society of Professional Engineers.

Tullis, J. P. 1989. *Hydraulics of Pipelines: Pumps, Valves, Cavitation, Transients*. New York: Wiley.

Wood, D. J., S. Lingireddy, P. F. Boulos, B. W. Karney and D. L. McPherson. 2005. "Numerical Methods for Modeling Transient Flow in Distribution Systems." *Journal—American Water Works Association* 97 (7): 104–15.

Wylie, E. B. and V. L. Streeter. 1978. *Fluid Transients*. New York: McGraw-Hill.

Appendix

Code Snippet to Retrieve Nodal Pressure

```
DataTable IHammer.GetTransientData(int nodeId)
{
    // Thanks to the Bentley Technical Support for reviewing
    this code snippet.
    TransientResultsProjectContext projectContext = new
    TransientResultsProjectContext(m_Project.ProjectManager,
    m_Project.TransientResultsViewerProfileElementManager,
    m_Project.TransientResultsViewerTimeHistoryElementManager,
    m_Project.TransientResultsViewerNodeHistoryElementManager);
    DataTable dt = new DataTable();
    dt.Columns.Add("Time");
    dt.Columns.Add("Pressure");

    try
    {
        int pressureId = (int)Haestad.HAMMER.TransientResults-
        Viewer.Domain.TimeHistoryType.Pressure;

        TransientResultsViewerTimeHistoryElement element =
        m_Project.TransientResultsViewerTimeHistoryElementManager.
        GetElement(nodeId, pressureId);

        using (TransientResultsGraphFormModel graphModel
        = new TransientResultsGraphFormModel(m_ApplicationModel,
        projectContext, element))
        {
            TrvManagerBase resManager = graphModel.Transien-
            tResultsGraphManager;
```

```
            IField timeField = resManager.Field(string.Format("trv-
            Time{0}", (this as IDataIO).ActiveScenarioId));

            Dictionary<int, double> times = (Dictionary<int, dou-
            ble>)timeField.GetValues();

            IField pressureField = resManager.Field(string.For-
            mat("trvPressure{0}", (this as IDataIO).ActiveScenarioId));

            Dictionary<int, double> pressures = (Dictionary<int,
            double>)pressureField.GetValues();

            for (int i = 0; i < pressures.Count; i++)
            {
                dt.Rows.Add(times[i], pressures[i]);
            }
        }
    }
    finally
    {
        projectContext.ResetHOFResultsDictionary(true);
        projectContext.ResetHMRResultsDictionary(true);
    }

    return dt;
}
```