EPA
www.epa.gov/research
technical BRIEF
BUILDING A SCIENTIFIC FOUNDATION FOR SOUND ENVIRONMENTAL DECISIONS

Enhancements to the EPANET-RTX (Real-Time Analytics) Software Libraries (Fiscal Year 2015)

Background

Water distribution modeling tools can be used by water utilities to help ensure reliable delivery of safe drinking water to the public. These modeling tools are useful for system planning, optimization of operations, contamination warning system design, contaminant detection, and disaster response. The ability to perform these activities depends, however, on accurate hydraulic and water quality network models and on suitable software modeling tools. The U.S. Environmental Protection Agency (EPA) developed EPANET (Rossman, 2000) as an easy-to-use software tool for water utilities and the research community to simulate water flow and contaminant transport within drinking water distribution systems.

EPANET, like most water distribution system modeling software programs, performs modeling *off-line*, which means the engineer uses a static, stand-alone description of the water distribution system (i.e., network infrastructure model) disconnected from any real-time data. EPANET lacks a mechanism to easily integrate real-time hydraulic and water quality sensor monitoring data with water distribution system infrastructure models. Although water utilities have invested heavily in data and information infrastructure systems (e.g., Supervisory Control and Data Acquisition [SCADA] systems) which capture important hydraulic data (e.g., pressure, flow, tank level, pump status) and water quality data, these time-series data are stored but seldom used. Because these data are often never accessed or used, they are not leveraged for making important operational decisions, such as infrastructure model calibration and verification, water usage optimization, leak detection, or contaminant event detection.

The lack of suitable methods, algorithms and software tools by which SCADA operational data streams can be easily connected with network infrastructure models has been a barrier to the usefulness of real-time SCADA data. The integration of network models with real-time data is critical for being able to continuously assess the accuracy of modeling and simulation results. Recognizing that new software tools are needed to support the real-time fusion of SCADA data and network infrastructure models, EPA's National Homeland Security Research Center initiated an open source project, Open Water Analytics (https://github.com/OpenWaterAnalytics), for the development of the EPANET Real-Time eXtension (EPANET-RTX) libraries. The goal of the EPANET-RTX project is to develop robust,

September 2015
EPA/600/S-14/441

state-of-the-art, real-time analytical software tools for water distribution system modeling. EPANET-RTX represents a new approach to distribution system modeling because it enables the creation of a persistent connection between the network model and a SCADA database with automated data transformation and synthesis. The development of real-time analytics for water systems promises to provide drinking water utilities with the necessary tools to improve system operations and management and to achieve long sought after goals, such as better water pressure management, improved leak detection and management, advanced energy management, and better water quality management.

EPANET-RTX utilizes the district metered area (DMA) concept to organize network elements and determine demand areas. DMAs represent hydraulically distinct areas and are based on common sets of water sources and sinks. The use of DMAs as a water demand management concept was introduced in the United Kingdom (UK) in the early 1980s. UK Report 26 (UK Water Authorities Association, 1980) defined a DMA as an area of a distribution system that is specifically defined (e.g., by the closure of valves), and for which the quantities of water entering and leaving the district are metered. DMAs are an essential component of demand management in the UK and elsewhere, historically because of the lack of domestic customer metering. Not only do DMAs allow the utility to understand the spatial and temporal pattern of demand, they are used to estimate and control leakage.

In EPANET-RTX, each DMA is described completely by its set of boundary pipes (limited to those with a valid flow measure) or a status set to “closed” (effectively, a measure of zero flow). EPANET-RTX automatically constructs the complete and unique set of DMAs for a network using an algorithmic process informed by the infrastructure topology, flow measure locations, and pipe statuses. Each DMA is constructed in a straightforward procedure that involves the software traversing the network in a methodical manner (e.g., depth-first or breadth-first graph search) and recording the junctions that have been visited, including storage tanks. The network search stops at all boundary pipes (measured flows, or closed statuses), and continues until all possible paths from DMA junctions have been explored. At the conclusion of this process, the junctions and storage tanks are known for each DMA, as are their closed and measured boundary pipes. This native capability of EPANET-RTX is used to create the basic demand data sets — aggregated to DMA regions.

Main features of the EPANET-RTX Open Source Project:

- A cloud-based, open-source software project that fosters collaboration between developers and encourages contributions from users.
- A collaboration site dedicated to providing the critical software components needed to easily develop flexible and robust real-time water distribution system modeling applications tailored to the end-user’s needs and objectives.
- A library of software classes and tools that extend the base EPANET hydraulic and water quality simulation functionality to include SCADA data analysis, transformation, and predictive forecasting for hydraulic and water quality parameters.

Objective for EPANET-RTX Libraries

The objective of the EPANET-RTX open source project is to develop the building blocks to enable accurate and reliable forecasts of water distribution system hydraulics and water quality. By merging SCADA operational data streams with geographical information system (GIS) based infrastructure data and by using the physical laws governing fluid and constituent transport in pipes, real-time analytics seek to forecast system behavior across the water distribution system. Using the fused data streams, real-time analytics provide estimates of pressures, flows, tank levels, and water quality variables everywhere in the system, including locations where there are no physical sensors.

Forecasting water distribution system behavior into the future is considered by many in the water community to be the Holy Grail for real-time water network simulation. That is, having the necessary tools to provide a reliable and accurate method of forecasting water distribution system behavior into the future is deemed critical. The capability to predict future water demands and, therefore, system hydraulics and water quality is a necessity for dealing effectively with rapidly evolving and unusual system disturbances, such as an intentional or accidental contamination events. Real-time modeling and predictive forecasting software, however, must be scalable, allowing the capability to be within reach of every utility that wants it.

The Las Vegas Valley Water District (LVVWD) has for several years performed a daily process of forecasting water demand and determining system operational variables that reflect management goals. These pieces of information are fed into their network simulation models, and the resulting forecasts are assessed. LVVWD has documented the value in their process (e.g., detecting anomalous infrastructure and data conditions) allowing them to make corrections early on before larger problems are created and costs are incurred (Boulos et al., 2014). However, the LVVWD forecasting processes are supported by a customized and internally developed software system that is not easily extensible to other water utilities, and would be difficult (and thus expensive) to recreate at another water utility. Similar near real-time model applications have been developed elsewhere, although again at high cost and considerable effort (Kara et al., 2015).

The EPANET-RTX software libraries provide access to different algorithms (e.g., time-series transformation routines) and statistical models (e.g., filtering, screening and performing statistical routines) that are foundational to connecting real-time data with infrastructure network models and performing real-time modeling. These technologies involve accessing a SCADA historian database, using filtering, smoothing, and other data transformation methods, and then running hydraulic and water quality simulations. The EPANET-RTX software libraries provide a software scaffolding that interfaces with these data transformation technologies to enable the smooth migration of data from the measurement (SCADA) domain into the modeling domain. The EPANET-RTX open source project contains a set of object libraries used for building real-time hydraulic modeling environments. More specifically, the EPANET-RTX libraries provide a set of building blocks (C++ classes and wrappers) that can be used and extended to create real-time data fusion applications, such as data acquisition and predictive forecasting.

The typical use of the EPANET-RTX libraries comprises building an application that connects a water utility's network model with sensor measurements that have been or are being recorded in a SCADA historian and then running an extended-period simulation driven by real measurements. The EPANET-RTX libraries make the complex task of network model and SCADA data fusion easier for programmers and engineers to use. Many processes that would typically be considered part of network model calibration are implemented automatically by an EPANET-RTX-based real-time model.

Figure 1 provides a conceptual depiction of the method in which EPANET-RTX classes are used to extract SCADA data in real-time, transform and clean the data, and fuse the data with a distribution system network model to allow for improved operational analyses. The solid arrows represent the data flow and the dashed lines denote EPANET-RTX library classes that are used to support the indicated functionalities.

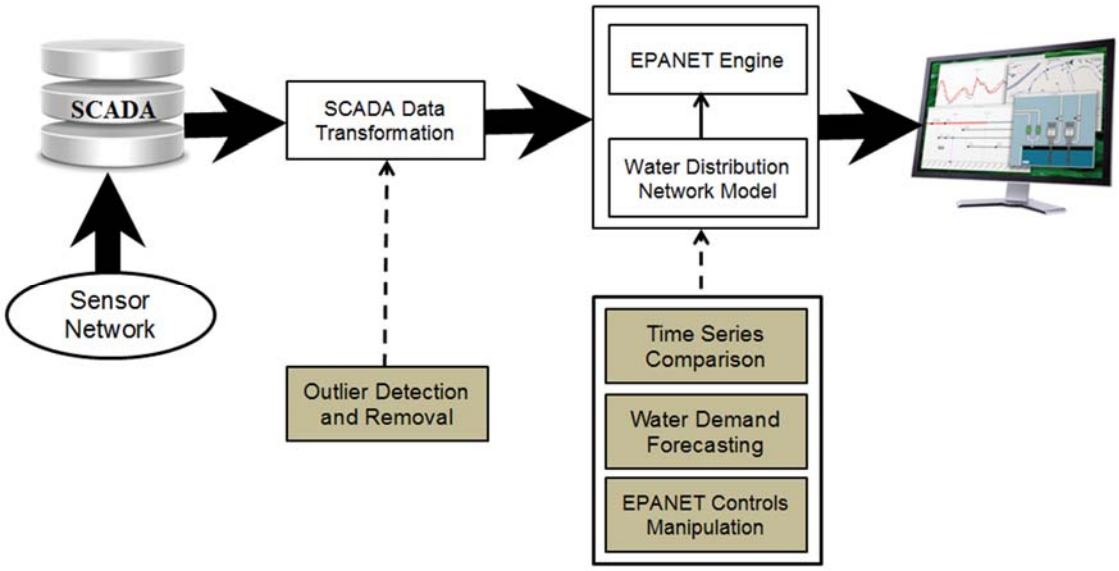

Figure 1. Illustration of EPANET-RTX-based Real-Time Analytics enabling SCADA Data fusion with a Water Distribution System Infrastructure Model supported by EPANET-RTX Library Classes (shaded).

FY2015 EPANET-RTX Enhancements

This technical brief summarizes advancements made to the EPANET-RTX software libraries in FY2015 and how these advancements are being applied to real water systems. The advancements to the EPANET-RTX libraries include tools for removing data outliers from time-series data and tools for performing quantitative analysis of time-series data (e.g., to compare SCADA measurements with model simulation outputs and assist in model calibration). Advancements also include the capability to manage EPANET controls for performing real-time simulations as well as forecasting. Finally, prototype tools (i.e., Python™, Python Software Foundation, Beaverton, OR) were developed for performing real-time water demand forecasting to support real-time hydraulic and water quality forecasting simulations.

Advancements to the EPANET-RTX software libraries (C++ classes) include new functionalities and modifications to existing functionalities to facilitate the new and advanced capabilities. The prototype water demand forecasting capabilities were developed using the Python/StatsModels module (http://www.statsmodels.org) along with the EPANET-RTX-based Python linkages. The EPANET-RTX linkages were developed to leverage the Python capabilities to provide demand forecasting and to provide hydraulic and water quality simulation forecasting of water distribution system behavior. Specifically, the Python/StatsModels module is now available through the EPANET-RTX libraries in order to perform data time-series model identification and parameter estimation as well as time-series predicting and forecasting. While this initial research has investigated and developed a prototype approach for water demand forecasting and predictive hydraulic and water quality simulation forecasting, it has also resulted in a development path for moving new, similarly complicated algorithms or methods into the EPANET-RTX libraries.

Forecasting into the future (i.e., forecasting in time to locations where sensors are unavailable) is more difficult than the current capabilities in the EPANET-RTX libraries (that is, now casting and hind-casting or historical simulations). The prototype methods for forecasting were developed by leveraging the correlations expressed in historical time-series data. Aggregate system water usage reflects the water usage by individual and commercial uses. Such aggregate usage is determined by a variety of factors including past and forecasted weather, system pressure, operations, and population-based water use. Water use within smaller regions of a distribution system, such as within DMAs, might be further affected by water customer dynamics that determine the movement of individuals within urban regions. The approach developed this year for forecasting aggregate demand was done using statistical time-series data models that are presumed to represent, through appropriate temporal autocorrelation structures, all of the integrated factors that affect water usage.

Table 1 lists the primary object classes within EPANET-RTX that were advanced this fiscal year along with a description of the functionality of each class. Further information about the EPANET-RTX classes is provided below.

Table 1. EPANET-RTX Software Enhancements Completed in FY2015

Object Classes	Description of the Functionality
Detection and Removal of SCADA Data Outliers	Classes designed to remove time-series data values that exceed static lower and upper bounds.
Quantitative Analysis of Time-Series Data	Classes designed with the general capacity to quantitatively measure the difference between two time-series objects.
Water Demand Forecasting	A subset of classes (and associated Python code) to provide water demand forecasting using statistical methods.
EPANET Controls Manipulation	Classes that can disable EPANET model controls running real-time simulations and enable such controls in forecasted simulations.

Detection and Removal of SCADA Data Outliers

One of the most important features of EPANET-RTX is its ability to use SCADA data to set real-time operation conditions for water distribution modeling. The EPANET-RTX libraries contain a set of SCADA data transformation tools to extend discrete SCADA data to a continuous time-series, such as with tank level and nodal pressure data. Noises in SCADA data, if not treated appropriately, could be propagated and result in error to important parameter estimates, such as DMA demand estimation values. Recognizing that data outliers needed to be removed from SCADA data, a set of EPANET-RTX classes were developed to support the functionality.

The EPANET-RTX class called “OutlierExclusionTimeSeries” was developed as a robust process for automatically identifying and eliminating outliers from time-series data during real-time data processing. The software class constructs an output time-series of data points from an input data time-series after excluding any data point that meets the definition of an outlier.

EPANET-RTX defines an outlier as a point value that lies outside the valid range [*q1 - as, q2 + as*], where *q1* and *q2* are exceedance statistics of the series over a user-specified time window, *w*; *s* is a statistic of the variability of the series defined over the same time window, and *a* is a user-defined scalar multiplier. The placement of the sampling window is either leading, lagging, or centered on, the current point. The statistic *s* is either the standard deviation or the inter-quartile range (IQR or difference between the 75th and 25th percentiles), and which mode is selected affects the valid range. If *s* is the standard deviation, then *q1* = *q2* which equals the mean over the window interval. When *s* is the IQR, then *q1* is the 25th percentile value, and *q2* is the 75th percentile. (This relates to a typical use in statistical process control, corresponding to the IQR with *a* = 1.5; this is also the common definition of outliers in a box and whiskers plot.) The OutlierExclusionTimeSeries is derived from the BaseStatsTimeSeries, which provides support for the basic statistical calculations defined over arbitrary time windows.

The OutlierExclusionTimeSeries class can be leveraged to develop a fairly general and robust process for automatically identifying and eliminating outliers from time-series data in real-time data processing. For outlier removal in particular, this new class will be combined with other native EPANET-RTX time-series processing capabilities.

Tank level time-series data contain data gaps and outliers that can lead to errors in the tank inflow rates that are automatically calculated by EPANET-RTX. These errors in turn would be propagated to the real-time DMA demands, and thus also to the simulation model results.

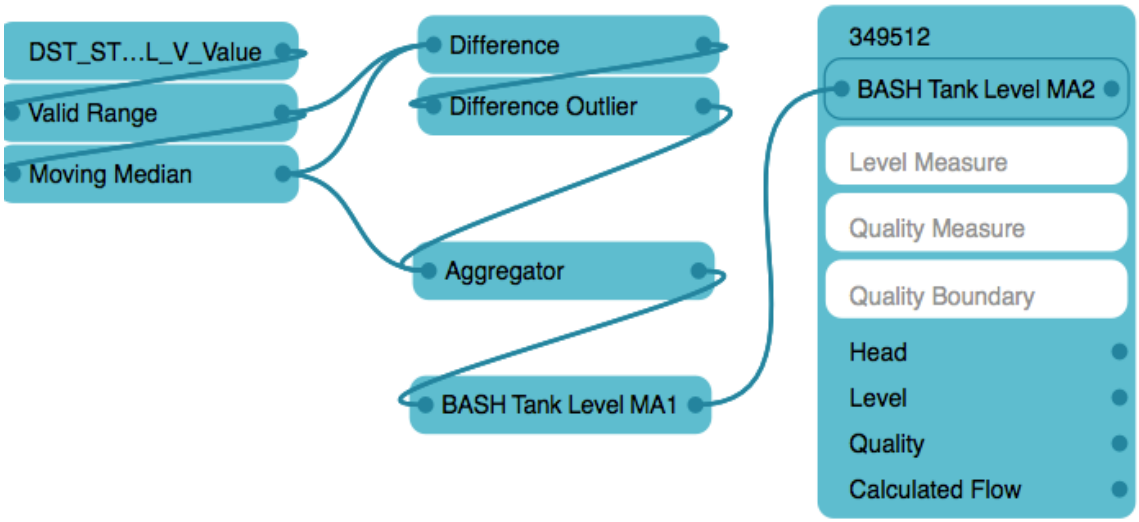

Figure 2. Typical use case for using outlier time-series removal class as applied to SCADA data source (upper left).

Figure 2 provides a schematic diagram illustrating the use of the outlier time-series data class using some example tank level data. Each block or rectangle in the figure represents an EPANET-RTX class that performs the indicated data transformation on its input time-series data stream. The rectangle labeled "DST_ST…L_V_Value" in the upper left corner of the figure is the SCADA data source. The larger rectangle on the right side of the figure represents the network model element's (Tank - "BASH" - parameter "level") measure/boundary anchor point to which the data stream is associated. This large rectangle is the terminal or endpoint after all the data transformations and flow into and out of the tank is determined. Each rectangle has (except the SCADA data source) both an input (left side of rectangle) and output (right side rectangle).

The general outlier removal approach illustrated above is based on filtering outliers from the set of residuals between an adequately smoothed model of the tank level data and the raw data stream. First, the raw data is ranged using the minimum and maximum tank levels ("Valid Range"). This process drops points that are physically unrealistic given the tank geometry. Next, the data are passed through a moving median filter ("Moving Median"). This filter serves as a simple model of the tank level data stream using a short, median time window, so that the resulting data stream is a reasonable representation of the level data and its variability. The median filter, however, is insensitive to outliers, so that the downstream difference of that data stream from the original (ranged) data ("Difference") results in a set of residuals where the outliers are exposed to a greater degree as statistical anomalies. In this process, the use of the median filter is just a simple example. If a better model of the process were available using other EPANET-RTX capabilities, it can be used and built into an application using the EPANET-RTX libraries. An example would be to replace the median filter with a short-term forecast of a statistical time-series model, fitted to the raw data stream.

These residuals are then passed through the EPANET-RTX outlier exclusion time-series filter ("Difference Outlier"), which removes any points that pass the definition of an outlier, based on a moving window calculation of the inter-quartile range. After the outliers are removed from the data stream, the resulting cleaned residuals are added back into the moving median filter ("Aggregator"), yielding the original (ranged) raw data, with the outliers removed. The remainder of the data processing are moving averages ("MA1" and "MA2") which are used to smooth the level data. This smoothing is required because the tank net inflow rate is computed from this tank level time-series data stream by numerical differentiation, which will amplify signal noise. A typical example of the outlier removal process is illustrated in Figure 3 below. This figure shows a short duration of raw tank level data (pink) along with the smoothed level data after outlier removal (blue). For comparison, the brown data stream is the smoothed data without the outlier detection. As can be seen, removal of the outliers prior to smoothing is effective in this case, as well as yielding a level data stream that more accurately represents the underlying level trace, and thus will more accurately represent the true tank net inflow. The outlier removal process is also important for enhancing the accuracy of automatic demand computations within DMAs. Documentation (doxygen code) for this new EPANET-RTX class is available on the [OpenWaterAnalytics website](https://github.com/OpenWaterAnalytics/epanet-rtx) (https://github.com/OpenWaterAnalytics/epanet-rtx).

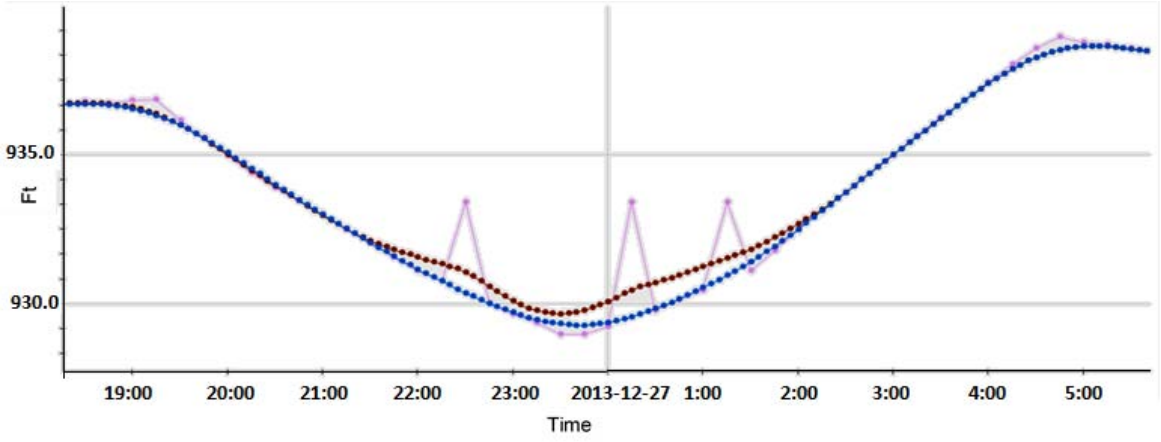

Figure 3. Example application of outlier time-series removal class on tank data.

Quantitative Analysis of Time-Series Data

The quantitative analysis of hydraulic and water quality simulation results can be insightful. For example, simple statistics such as mean, median, and maximum value, from a time-series data set can provide a useful, quantitative description of the data. The ability to perform a quantitative determination of the similarity of two data time-series is critical in many real-time model uses. One important example is the comparison of a water distribution system model simulation output (i.e., a data time-series of simulated pressure measurements) to its SCADA measurement counterpart. These types of quantitative analyses are critical in being able to perform water distribution system model calibration or evaluate model accuracy. For example, error statistics (e.g., mean square error) are important indicators to compare SCADA data with its corresponding simulated time-series.

Four new EPANET-RTX classes have been developed to support quantitative time-series data comparisons:

- ***BaseStatsTimeSeries*** is a class that supports basic statistical analysis of time-series data. This is the foundation class from which the OutlierExclusionTimeSeries and StatsTimeSeries classes are derived. The class includes basic functionality that defines the time window and position (leading, lagging, centered) over which statistics are computed and the computation of basic statistics (e.g., number of points, minimum, maximum, mean, and quartiles) of the time-series within that window. BaseStatsTimeSeries leverages summary statistics that were included in the base TimeSeries class.
- ***StatsTimeSeries*** is a class that constructs a time-series of a user-selectable statistic from the basic statistics support by the BaseStatsTimeSeries and using the window also defined by BaseStatsTimeSeries. The available statistics include standard deviation, variance, mean, median, 25% and 75% quartiles, interquartile range, maximum value, minimum value, point count, and root-mean-squared error.
- ***CorrelatorTimeSeries*** is a class derived from the Aggregator class and accepts two time-series inputs for correlation analysis. Points returned from this class are the dimensionless Pearson correlation coefficient [-1,+1] between the two series, defined over a time window that is either lagged, centered on, or leading the point. The correlation coefficient is defined for two series of equal length, and thus the number of points analyzed from each series must be identical. The points selected from each series depends on the clock assigned to the CorrelatorTimeSeries object. If the object is not assigned to a clock, then the point times are taken from the merged set of point times from both series, and point values are in general interpolated values at those times. If the object has a clock, however, then the point times are taken from that clock (whether it be regular or irregular) instead of from the respective clocks of the input series. This allows, for example, setting the clock equal to that of one of the input series, in which case the second series points are interpolated to the point times of first. (This use case matches a typical one for correlation analysis between modeled and measured time-series, where one would often expect the modeled values to be interpolated to the times of the measurements.)
- ***TimeOffsetTimeSeries*** translates a time-series of data along the time axis by a specified positive or negative lag value in seconds with the point values untransformed. For example, a time lag of -3600 seconds would result in a time-series with the same values but delayed for one hour. A main use case for this class is to construct correlations between two time-series at a particular lag (using the CorrelatorTimeSeries), and thereby identify the value of the lag that results in the maximum correlation magnitude, with both the value of the lag and the associated correlation being reported as error descriptors.

Water Demand Forecasting

Forecasted simulation of water distribution networks is important in predicting system hydraulic and water quality behavior and supporting near-term operational decision making for utility operators. An accurate forecasted simulation of hydraulic and water quality behavior in a water distribution system depends on a reliable water forecasting scheme that captures water use patterns in the system and uses sophisticated statistical methods to predict water demand. Statistical models are used to explain variations in the time-series data over time in order to be able to support forecasts for the time-series data into the future. The approach to building a general forecasting capability into EPANET-RTX leverages these statistical time-series models, along with other methods, to forecast water usage within DMAs, while making use of real-time SCADA data such as operational statuses, settings of pumps and valves, system boundary head, and water quality.

EPANET-RTX libraries already had the capability for accessing and transforming data streams to automatically construct historical DMA demands. These same time-series data streams were used for developing and testing statistical models to forecast future demand and, thus, drive a real-time, predictive water distribution system simulation model. The accurate forecasting ability for complete system state (e.g., predicting flows and pressures) depends, at least in part, on an accurate forecast of DMA water demand. The work completed in FY2015 focused on the identification of formal methods for statistical model structure identification and parameter estimation in order to develop an understanding about how to develop good DMA-based demand forecasting models and how to describe their accuracy. Recognizing the importance demand forecasting, several EPANET-RTX classes were developed to support robust and effective water demand forecasting.

All statistical analysis and time-series data models for the DMA demand data were developed and investigated using the Python StatsModels module (www.statsmodels.org). StatsModels is a Python module that provides classes and functions for the estimations of many different statistical models, including, for example, conducting statistical tests and statistical data exploration. Because the code is open source and released under the Modified BSD (3-clause) license, the entire code base is open, free, and distributable with other software projects like EPANET-RTX, either in source code or binary form, under certain requirements.

For prototyping purposes, the forecasting and statistical methods researched were developed in a Python integrated development environment. Because of the EPANET-RTX-Python linkages that were developed, EPANET-RTX is now capable of leveraging Python code for new EPANET-RTX classes and methods, including the Python StatsModels methods. These StatsModels methods were used for time-series model identification and parameter estimation as well as time-series prediction and forecasting. Although the methods investigated here were produced in prototype Python form, this work established a clear development path for moving successful prototypes into EPANET-RTX.

The StatsModels project (official release at the time of writing is Version 0.6.1) contains many sub-modules under active development by various authors. The EPANET-RTX work described here for demand forecasting used the StatsModels' *StateSpace* project which contains classes and functions that are useful for time-series analysis using StateSpace methods. For these types of time-series models, evaluating the likelihood function is a byproduct of running a Kalman filter whose parameters are theoretically related to the time-series model parameters. One important application of a fully-fledged time-series model using the StateSpace backend is the SARIMAX class, which implements general Auto-Regressive, Integrated, Moving-Average time-series models with additive and multiplicative seasonal effects, as well as exogenous variables.[1] The StateSpace/SARIMAX time-series models are a recent addition which include a seasonal modeling capability[2]. The StateSpace/SARIMAX project is still under active development and scheduled for its first official release in StatsModels v0.7, the EPANET-RTX project team used the pre-release SARIMAX code to support the work on demand modeling and forecasting. As StateSpace/SARIMAX matures it should provide a stable platform to build upon for general forecasting within EPANET-RTX, and the underlying maximum likelihood and Kalman filtering modules might also be leveraged for other tasks (e.g., real-time hydraulic and water quality parameter estimation).

A classical approach was adopted for statistical model identification, parameter estimation, and model accuracy evaluation, as supported by the Python/StatsModels/StateSpace module. Model identification refers to the process whereby a time-series forecasting model structure is specified and justified. *Structure* refers to the number of autoregressive and moving average terms, and the degree of differencing, both for small lags and seasonal period lags, and including the specification of the seasonal period. *Parameter estimation* refers to the algorithmic process of estimating the best values of the model autoregressive and moving average parameters, given a particular DMA demand data set used to compare with the forecasting model predictions. In general, model identification is a process that invariably relies on human interpretation of various statistics that relate to the goodness of fit of the model predictions to the data, the number of parameters to be estimated, the estimated precision of the parameter estimates, and the statistical characteristics of residual differences between the model predictions and the data. In contrast, parameter estimation is an algorithmic process that either converges successfully, or not, and in the case of non-convergence, the reason is usually assumed to be a model structure that is too heavily parameterized relative to the information contained in the data. While parameter estimation is an algorithmic process, there are meaningful connections between parameter estimation and model identification that ties these two processes together. For example, tests on model residuals that are important for approving a particular model form can only be done once parameter estimates are obtained for that particular model. However once the model form is determined through the non-algorithmic

[1] Exogenous variable is an independent variable that affects a model without being affected by the model. An example important here would be rainfall which can affect water demand.

[2] Seasonal capability in the context of time series models means an ability to incorporate autoregressive or moving average terms with non-consecutive lags. This is critical for DMA demand time series because they will be expressed on an hourly or sub hourly frequency, yet have seasonal components on the order of days or weeks. Without a seasonal modeling capability, such terms could only be incorporated through the use of consecutive lags of very high order (e.g., 24 or 168 in the case of hourly data), which would be practically impossible to estimate due to the number of parameters.

process, that forecasting model can then be incorporated directly into EPANET-RTX through the above mentioned procedures, including both the estimation of parameters and predictions and forecasts. Prediction means estimating the behavior of a parameter between data points. Forecast means extending those parameter estimates to situations where there are no physical sensors or information.

The overall process of model identification and parameter estimation can be applied in a practical manner. The Python/StatsModels module can be utilized outside of EPANET-RTX for forecasting model structure identification, with the forecasting model structure then becoming part of a standard EPANET-RTX real-time water distribution system model configuration. EPANET-RTX already includes efficient processes for exporting raw and processed data streams (e.g., DMA demands) to standard data base formats such as SQLite (www.sqlite.org). Moreover, as part of this task, a Python module was developed to allow for easy querying of EPANET-RTX SQLite databases, so that EPANET-RTX processed datasets can be efficiently generated and then imported into the Python/StatsModels module used for the forecasting model identification.

The Python EPANET-RTX modules are included in the EPANET-RTX code distribution so that they can continue to be maintained and extended, and users of EPANET-RTX who wish to implement statistical models can have a Python tool chain that is available for accessing data and performing forecasting model identification and parameter estimation. The results described in the section titled “Illustrative Utility Case Study” provide an illustration of this approach using the Python/EPANET-RTX tool chain.

EPANET and EPANET-RTX Controls Manipulation

The EPANET engine (Rossman, 2000) has two different methods of evaluating and executing simulated control scenarios and actions: Controls and Rules.

Controls are in many ways the simpler of the two methods. The constraint imposed on a control description is that it must be contingent on either a Node grade or an elapsed simulation time. Further, the control can only affect the setting or status of a single Link element when activated. The syntax for declaring a control is simple (see Figure 4). In contrast to the "rules" method, the controls method, which depends on a Tank level, may cause the hydraulic simulation to use a smaller time step in order to capture the intended state change described by the control.

Rules, on the other hand, are capable of arbitrary complexity and length. Through using Boolean logic and conditional clauses in the syntax (see Figure 5), a user is able to associate a desired set of one or more actions with a particular set of one or more conditionals. The conditionals can reference almost any attribute of the dynamic hydraulic simulation. However,

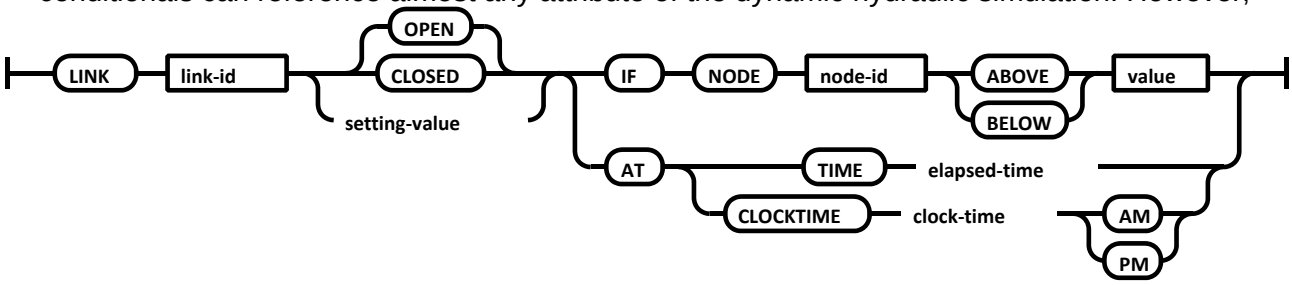

Figure 4. EPANET Controls Syntax.

Rules are evaluated on a rigid time step, and can only interrupt a hydraulic time step to within the resolution of the finer "rule time step" (by default 1/10th of the hydraulic time step length).

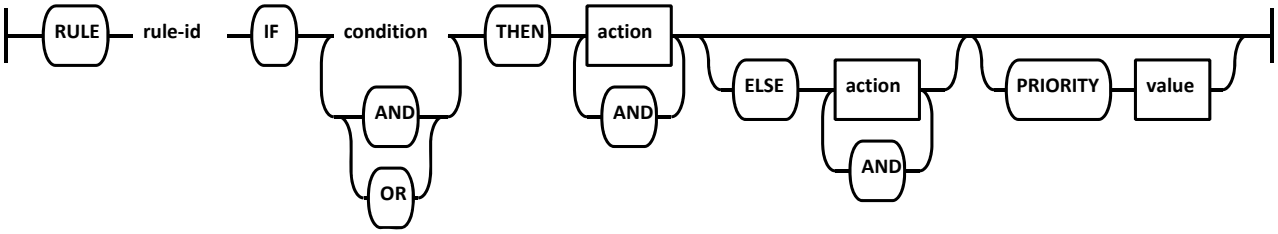

Figure 5. EPANET Rules Syntax.

These differing approaches mean that controls cannot always be represented as rules, and vice versa. However, there exist enough commonalities that perhaps future open-source development can merge the two into a unified rule-based control scheme.

To support real-time predictive modeling, certain assumptions about system control logic, as described by a dynamic control scheme, were made in the EPANET-RTX development. A pragmatic approach was used that involves the extension of the existing EPANET-based technologies to attain the goal of forecasting system behavior. The current functionality (for real-time simulation) in EPANET-RTX disables all model controls and dynamic system adjustments

are interpreted from the historical record. Two new techniques were developed in addition to the current real-time (or historical-operational) modeling method:

- *Forecasting Simulation* - Model controls are re-enabled, and EPANET-RTX controls are disabled, from the start-of-forecast. Existing EPANET logic automatically adjusts system parameters in response to the forecasted simulation, which is driven by demand forecasts.
- *Comparing Control Actions* - When both the forecast of a particular time range and its real-time "retrospective" simulation have been performed, EPANET-RTX must provide a set of outputs describing the difference between the "model assumed" controls actions taken in the forecast with the "actual" historical record of what control actions were taken in the real system.

In order to gain oversight and a deeper inspection of model-based controls, the EPANET API (application programmer's interface) was extended to provide a more expressive set of accessors related to its control logic. The canonical API for the EPANET toolkit (Rossman, 2000) only allows very basic access to the underlying program structures that are of interest, and then only for Controls (not Rules). Further, the EPANET-RTX simulation library was also extended to access the required deeper EPANET-level information and expose it in an object-oriented fashion to support the use cases required above. The extension enabled:

a. Selectively enabling / disabling controls and rules
b. Enumerating the elements that are affected by a control or rule
c. Exposing both functionalities (a) and (b) in the EPANET-RTX class library simply and intuitively.

To this end, the requisite modifications to the EPANET and EPANET-RTX libraries, respectively, are as follows:

EPANET Libraries

OW_controlEnabled / ***OW_setControlEnabled ::*** inspect and alter whether a control is enabled for hydraulic simulation.

OW_ruleEnabled / ***OW_setRuleEnabled ::*** inspect and alter whether a rule is enabled for hydraulic simulation.

OW_getRuleAffectedLinks :: get a list of links affected by a particular rule.

Taken together, these new EPANET API functions enable the client code to describe in a greater level of detail the number and extent of both controls and rules, and to inspect the controlled Link elements that are affected by each.

EPANET-RTX Libraries

RTX::Model::Control :: This is a new, simple class describing the controls and rules (combined). The class retrieves the list of pipe elements affected by a control along with which parameter is affected (status or setting).

RTX::Model::runForecast :: This is a new method that prepares the EPANET engine to run a forecasted simulation and performs an extended-period analysis for a time frame specified. The forecasted simulation is similar to a real-time simulation, but reverts the hydraulic engine to use preprogrammed control logic rather than real-time data feeds from the SCADA system.

The extensions to EPANET-RTX libraries are fairly simple, but provide important information related to the control of the forecasted simulation and enables the running of an extended period simulation in the forecasting mode. Applications developed from the EPANET-RTX libraries may use the control information to enumerate the affected link elements and retrospectively determine the historical accuracy of previously-forecasted control actions. The new class addition only surfaces the information that is pertinent to the use case described here, but could be extended further to provide a more detailed view of the model control logic.

The following is a sample of EPANET-RTX C++ code demonstrating the execution of a forecasted simulation and an extended period analysis:

```
time_t, t1, t2, t3; /* = [...] */
PointRecord::_sp rtSimOutputDb /* = [...] */;
PointRecord::_sp forecastOutputDb /* = [...] */;
PointRecord::_sp forecastControlsDb /* = [...] */;  Model::_sp model /* = [...] */;

model->setStorage(rtSimOutputDb);
model->runExtendedPeriod(t1, t2); // run historical simulation

model->setStorage(forecastOutputDb);
model->setForecastControlsStorage(forecastControlsDb);   model->runForecast(t2, t3); // run a
forecast

// now perform the historical simulation for the time frame previously forecasted.
model->setStorage(rtSimOutputDb);   model->runExtendedPeriod(t2, t3);

// get the forecasted controls for a pump
TimeSeries::_sp forecastedPumpControl = model->forecastedSettingWithLinkName("pump_1");
// and get the actual controls for the same pump.
TimeSeries::_sp actualPumpControl = model->pumpWithName("pump_1")->settingParameter();
```

Real-Time Modeling Path Forward

The primary goal for advancing the EPANET-RTX libraries this year was to develop a reliable way to predict complete water distribution system hydraulic state (e.g., pressures, flows, and tank levels). To achieve this goal has required defining what is meant by “real-time simulation”, and to refine the tools available to probe this question. With the completion of the work outlined here, the EPANET-RTX libraries are now better positioned to achieve the needed predictive abilities. The added enhancements from the “EPANET and EPANET-RTX Controls Manipulation” and the “Water Demand Forecasting” will enable EPANET-RTX users to build an application that can easily run forecasted simulations of hydraulics and water quality. The outputs of such forecasts take the form of prediction time-series, which can be stored in a variety of database formats. The practical utility of these forecasts will be determined with the development and testing of suitable numerical approaches to enable and perform the necessary error analyses.

To illustrate this point, consider the use case of a forecasted control sequence (on-off status values) generated from the combination of control rules and forecasted hydraulic behavior that is to be compared with an historical record of on-off status values from a process SCADA historian. The meaningfulness of comparing two Boolean series is not well-defined in the general sense, nor is the approach self-evident. For instance, is a simple comparison of residuals adequate to quantify any divergence? Or should an event-based statistical analysis be considered? How much divergence is significant for operational purposes and goals? These questions are not yet answered. However, with the enhancements described here, these questions and others could now be answered using the EPANET-RTX libraries.

Illustrative Utility Case Study

Two different DMA demand datasets (Figures 6a and 6b) were used to support the development of the two prototype DMA demand forecasting models previously discussed. Both DMAs are from the distribution system of a mid-western U.S. water utility, with average demands of approximately 10 and 30 million gallons per day (MGD) for DMAs 1 and 2, respectively. Figures 6a and 6b show DMA demand time-series plots calculated by EPANET-RTX from the raw flow and tank level data series for these two DMAs over a three week time frame (the demand is shown in the top panel of the pair of graphs shown for a DMA). These three weeks of data were used for the forecasting model identification and parameter estimation; a larger augmented data set was then used for calculating the forecast accuracy of the fitted models. Figure 7 shows the autocorrelation and partial autocorrelation functions for the two raw DMA demand time-series shown in Figures 6a and 6b. These figures illustrate the important temporal correlation characteristics of the demand time-series data. The temporal correlation characteristics of the demand time-series data determine the form of the statistical time-series models that should be used. In particular, the temporal characteristics determine the number of autoregressive and moving average terms, both for short lags and seasonal periods. The autocorrelation functions for both data sets suggest a mixed form for short lags, with both moving average and autoregressive terms, because of the slowly decaying autocorrelations at short lags combined with the "shoulder" exhibited at lags of 1 or 2 (i.e., the decay of the autocorrelation does not appear to be strictly exponential[3]). The autocorrelations also show expected significant relationships at multiples of a 24-hour seasonal period and decay to both sides of those lags, which indicates that a seasonal period of at least 24 hours will be necessary in the forecasting

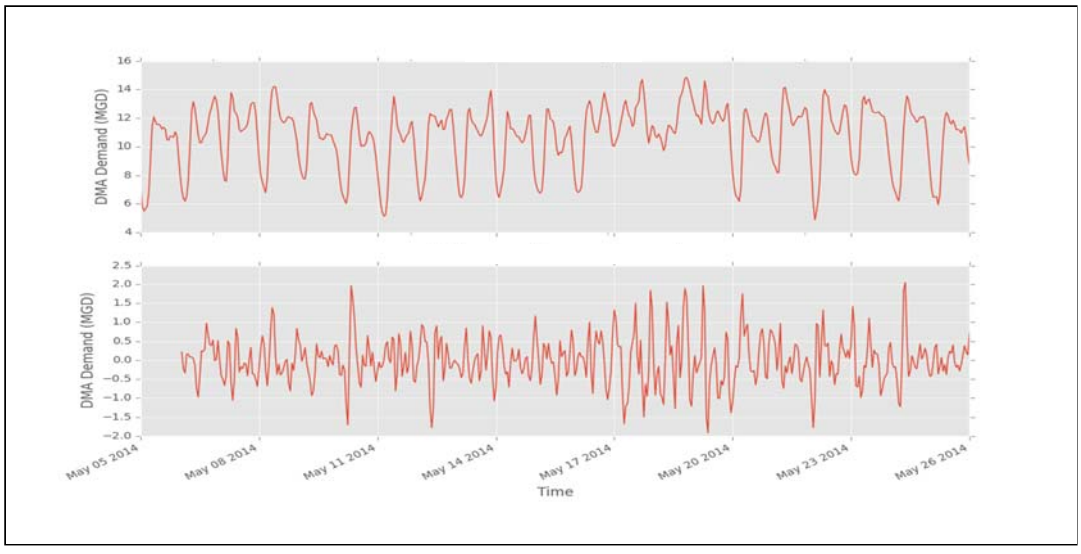

Figure 6a. Raw and differenced demand time-series data for DMA 1.

[3] The theoretical autocorrelation structures of pure autoregressive and pure moving average model forms are well known. If a pure autoregressive model, then the autocorrelation structure will be exponential decay, whereas if pure moving average the autocorrelation structure will be spikes at the lags associated with the moving average terms.

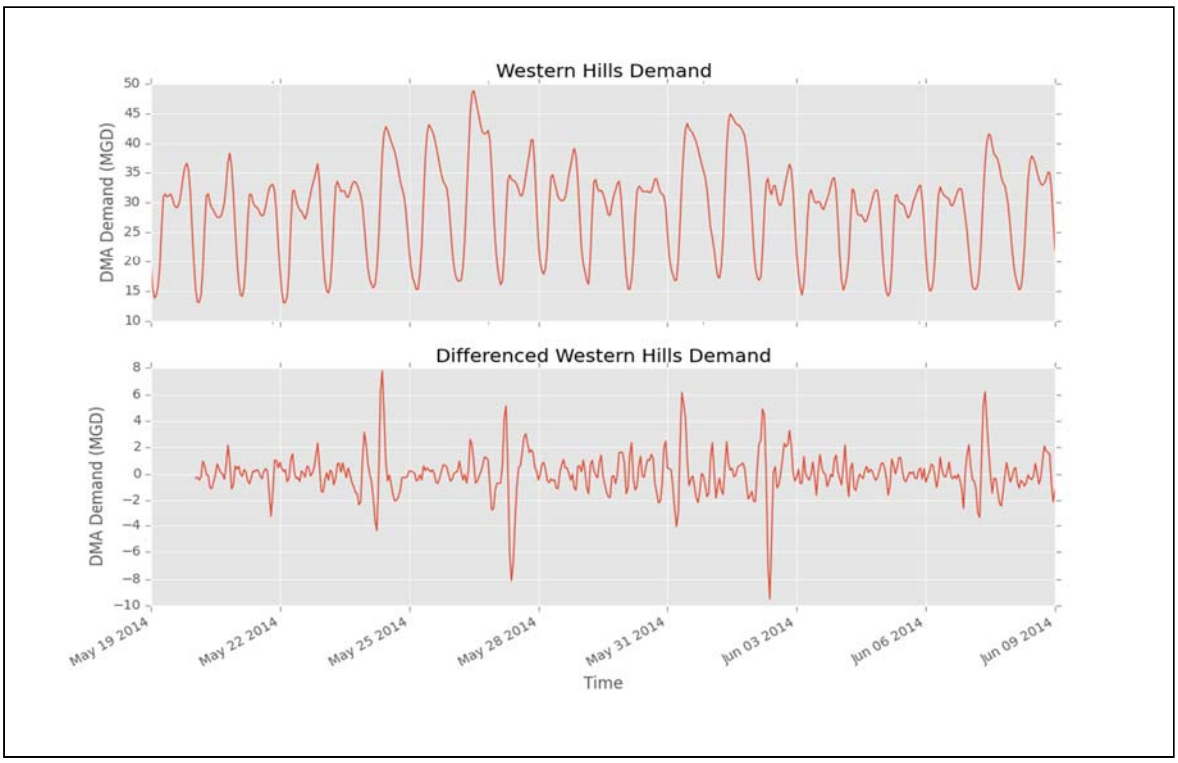

Figure 6b. Raw and differenced demand time-series data for DMA 2.

model. This also indicates that a seasonal difference of 24 hours is useful to try as a strategy to remove the non-stationarity introduced by the seasonal correlation.

Paired with the raw demand time-series data in Figures 6a and 6b (bottom panel in each figure) are the series after one level of differencing at lag 1 and a seasonal period of 24. The lag 1 differencing is the common strategy for removing non-stationarity of mean — a requirement of applying the sort of time-series models proposed here. The raw demand series do not show obvious signs of non-stationarity, even though there is some variation apparent in the mean of the series over time. Thus, this lag 1 differencing is regarded as being possibly not important, but also probably not harmful for modeling purposes. The seasonal differencing at a lag of 24 hours is intended to remove seasonal non-stationarity as was exhibited in the Figure 7 autocorrelation functions. Resulting autocorrelation and partial autocorrelation functions for these two differenced series are shown in Figure 8. In contrast to Figure 7, the autocorrelation functions in Figure 8 do not exhibit obvious non-stationarity at seasonal periods, although there remain possible significant correlations at seasonal periods (for DMA 1), and thus alternative forecasting models should consider seasonal lag autoregressive or moving average terms. The autocorrelation and partial autocorrelation plots in Figure 8 indicate overall that a mixed model of relatively low order should be considered. This is because the partial autocorrelation coefficients show a slow decay (indicative of a moving average model behavior). In addition, the autocorrelation function shows significant values at longer lags, and so autoregressive

behavior cannot be confidently ruled out. It is also possible to interpret these results as a pure moving average form of order 1, if one regards the autocorrelation function as having a shoulder at lag 1 and none of the significant parameters are slowly decaying.

Overall, the analysis suggests that seasonal differencing should be strongly considered at least at a lag of 24 hours (and possibly at other lags, though that evidence is less clear). It also suggests that in addition to seasonal differencing, forecasting models of a mixed type with low orders at both short and seasonal period lags should be considered. As a result of this analytic and manual inspection process, a variety of different forecasting model structures were developed using the Python/StatsModels tool chain, and each of these forecasting models were fit to the raw demand data. For each forecasting model that was fit, a variety of summary statistics that help to interpret the quality of the forecasting model and the estimates as well as statistical tests on model residuals were obtained. The following paragraphs provide more discussion and illustration of the approach.

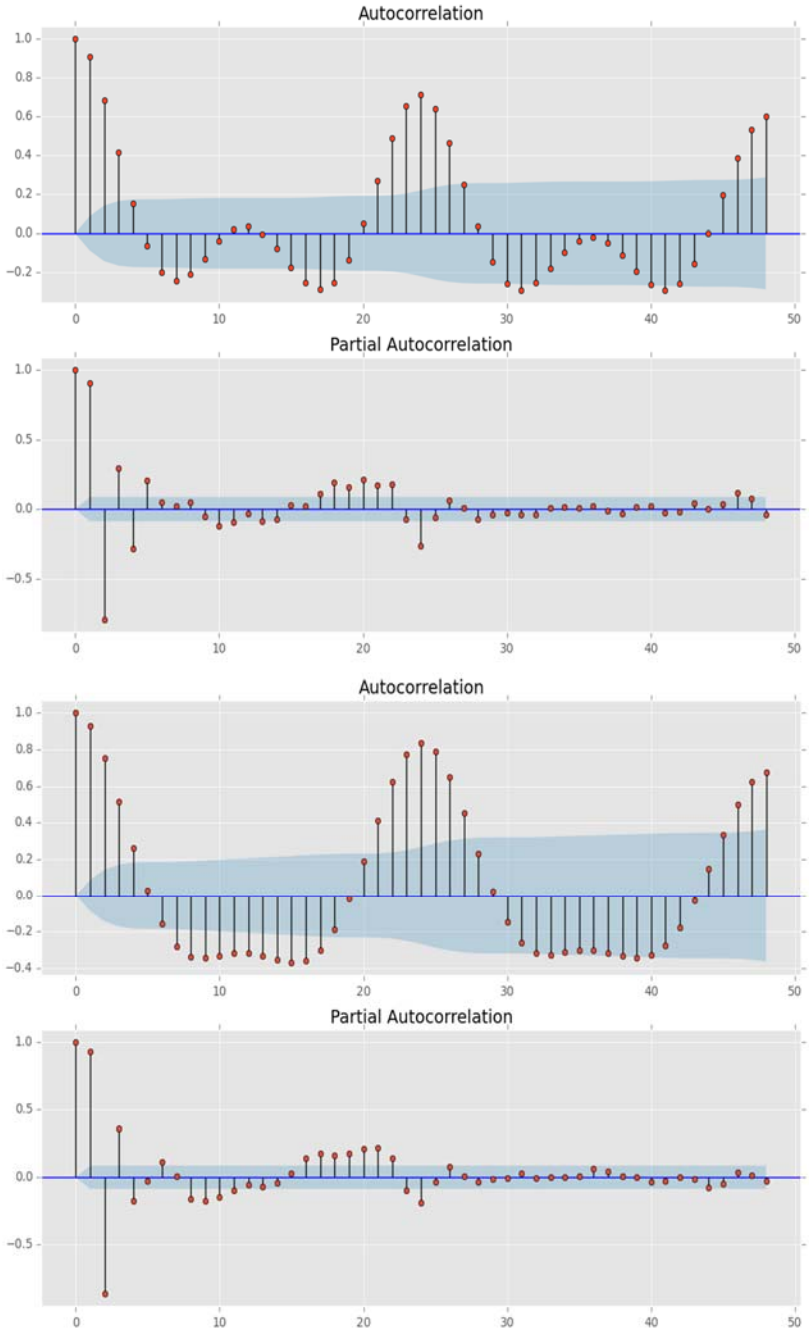

Figure 7. Raw demand data autocorrelation and partial autocorrelation functions for DMA 1 (top) and DMA 2

Figure 9 shows the standard output from the Python/StatsModels SARIMAX method for maximum likelihood parameter estimation. The summary information indicates that the form of the time-series forecasting model is a SARIMAX (2,1,2)x(2,1,1,24). The first term in parentheses describes the short lag model terms, indicating that it is an ARIMA model with 2 orders of autoregressive terms, 1 level of differencing, and 2 orders of moving average terms. The second term describes the seasonal ARIMA portion of the model, indicating that it includes 2

orders of seasonal autoregressive terms, 1 level of seasonal differencing, and 1 level of seasonal moving average terms. The final term specifies the season period to be 24 (hours).

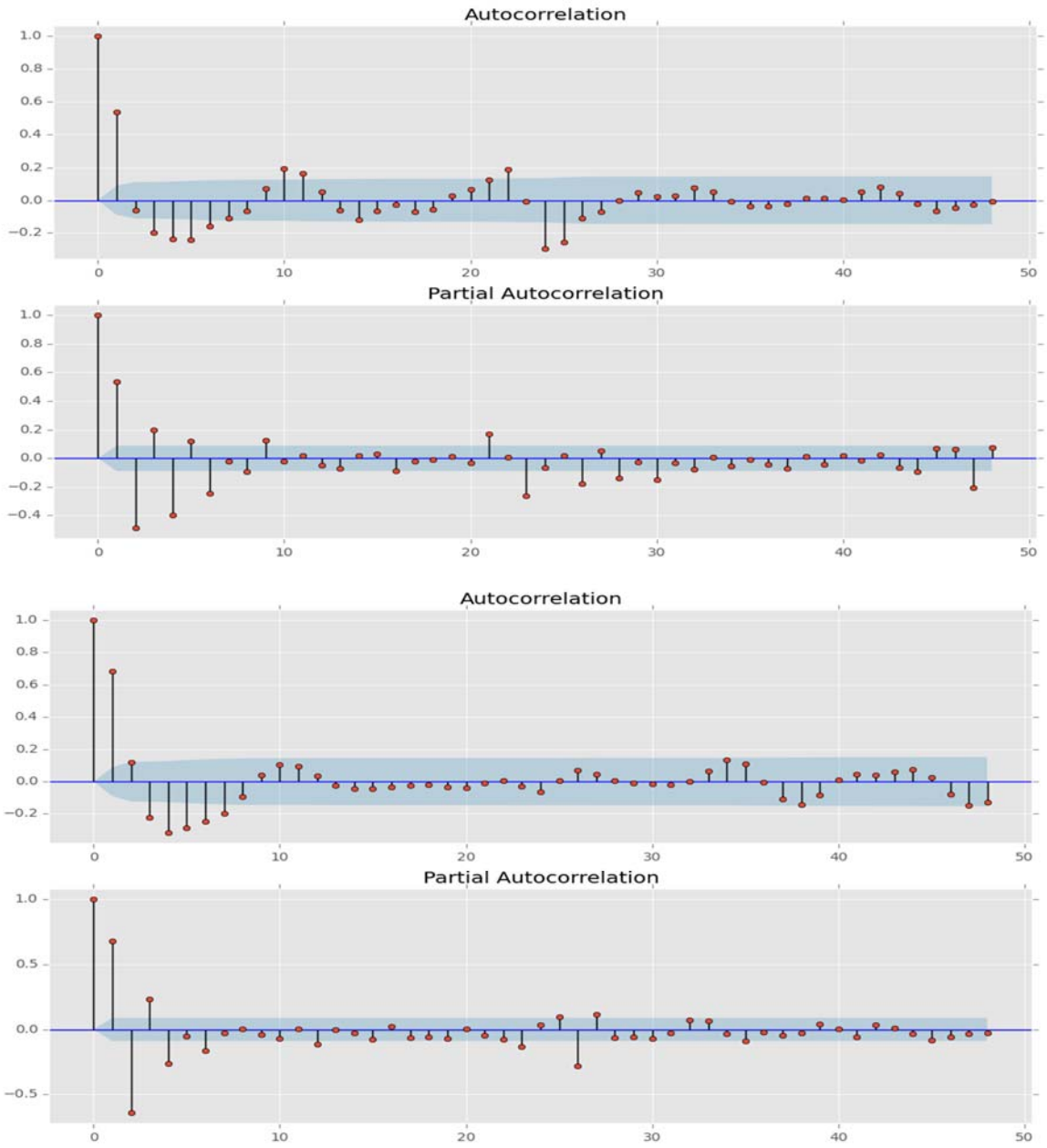

Figure 8. Differenced demand data autocorrelation and partial autocorrelation functions for DMA 1 (top) and DMA 2 (bottom).

The log likelihood is a quantitative descriptor of how well the forecasting model fits the time-series demand data, and, in fact, is the quantity that is maximized when fitting the forecasting model to the data. This quantity can be used to compare different forecasting models for the same data set, in terms of how well the forecasting model represents the data. However there are other important considerations for forecasting model selection; one is to make sure there is not an excess of parameters, because this usually implies that the parameters cannot be

estimated with a high degree of precision. The Akaike information criterion (AIC) is one measure of the relative quality of a statistical model that takes into account these factors in addition to the fit of the forecasting model to the data. The AIC measures the amount of information lost in representing the data with an approximate forecasting model. It is intended to be used, but not without consideration, as a way to quantitatively compare different forecasting models and thus as a means for forecasting model selection (i.e., model identification). In principal, many different forecasting model forms could be selected that minimize the AIC, but in practice the AIC is used more as a method of identifying and excluding obviously poorer forecasting models, as opposed to a means to discriminate uniquely and precisely between different forecasting model forms. For the DMA demand data sets used, the forecasting models summarized in Figure 9 both had very good likelihood and AIC, compared to the other forecasting model that was considered.

The main tables in Figure 9 show the model parameters, their estimated values, and the uncertainty in those estimated values. The parameters are labelled according to a standard convention of [ar|ma].[S.].Li where ar/ma indicates autoregressive or moving average term, S

```
Optimization terminated successfully.
         Current function value: 0.277210
         Iterations: 17
         Function evaluations: 1945
                                Statespace Model Results
==============================================================================================
Dep. Variable:                          Brecon Demand   No. Observations:                  505
Model:             SARIMAX(2, 1, 2)x(2, 1, 1, 24)   Log Likelihood                -139.991
Date:                            Thu, 23 Jul 2015   AIC                            297.982
Time:                                    17:36:26   BIC                            336.003
Sample:                                05-05-2014   HQIC                           312.895
                                     - 05-26-2014
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept     -0.0002      0.000     -0.689      0.491      -0.001       0.000
ar.L1          1.2054      0.035     34.854      0.000       1.138       1.273
ar.L2         -0.4333      0.033    -13.272      0.000      -0.497      -0.369
ma.L1         -0.0816      0.018     -4.474      0.000      -0.117      -0.046
ma.L2         -0.8212      0.029    -28.513      0.000      -0.878      -0.765
ar.S.L24       0.1474      0.031      4.762      0.000       0.087       0.208
ar.S.L48      -0.0123      0.024     -0.508      0.611      -0.060       0.035
ma.S.L24      -0.9864      0.132     -7.451      0.000      -1.246      -0.727
sigma2         0.0914      0.012      7.596      0.000       0.068       0.115
==============================================================================
Ljung-Box (Q):                       53.97   Jarque-Bera (JB):                 20.99
Prob(Q):                              0.07   Prob(JB):                          0.00
Heteroskedasticity (H):               1.15   Skew:                             -0.21
Prob(H) (two-sided):                  0.37   Kurtosis:                          3.94
==============================================================================
Optimization terminated successfully.
         Current function value: 1.153425
         Iterations: 5
         Function evaluations: 582
                                Statespace Model Results
==============================================================================================
Dep. Variable:                  Western Hills Demand   No. Observations:                  673
Model:             SARIMAX(2, 1, 2)x(2, 1, 1, 24)   Log Likelihood                -776.255
Date:                            Fri, 17 Jul 2015   AIC                            1570.510
Time:                                    12:08:12   BIC                            1611.115
Sample:                                05-21-2014   HQIC                           1586.235
                                     - 06-18-2014
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
intercept   4.293e-05      0.000      0.151      0.880      -0.001       0.001
ar.L1          1.3622      0.026     51.768      0.000       1.311       1.414
ar.L2         -0.5563      0.030    -18.495      0.000      -0.615      -0.497
ma.L1         -0.2647      0.030     -8.800      0.000      -0.324      -0.206
ma.L2         -0.6687      0.035    -19.286      0.000      -0.737      -0.601
ar.S.L24       0.2581      0.026      9.880      0.000       0.207       0.309
ar.S.L48      -0.1202      0.025     -4.735      0.000      -0.170      -0.070
ma.S.L24      -0.9987      0.163     -6.140      0.000      -1.318      -0.680
sigma2         0.5702      0.090      6.360      0.000       0.394       0.746
==============================================================================
Ljung-Box (Q):                       60.94   Jarque-Bera (JB):              1292.77
Prob(Q):                              0.02   Prob(JB):                         0.00
Heteroskedasticity (H):               1.56   Skew:                             0.61
Prob(H) (two-sided):                  0.00   Kurtosis:                         9.81
==============================================================================
```

Figure 9. Summary of fitted SARIMA model parameter values and statistics for DMA 1 (top) and DMA 2 (bottom).

indicates a seasonal lag period coefficient, and i indicates the lag. For the forecasting models selected, there are nine parameters being estimated, including the intercept (bias), the white noise variance (sigma2), and 7 autoregressive or moving average parameters: order 2 small-lag autoregressive and moving average terms (ar.L1/L2 and ma.L1/L2); order 2 seasonal autoregressive terms with seasonal period 24 (ar.S.L24/L48); and order 1 seasonal moving average term with the same seasonal period (ma.S.L24). The final parameter listed (sigma2) is the estimated variance of the white noise disturbance that is a fundamental component of all

statistical time-series models. Final estimated parameter values, as well as their standard errors (i.e., standard deviations), are also shown. The standard errors are important for assessing the degree of uncertainty, or lack of precision, in the parameter estimates. Essentially, these standard errors should be small relative to the estimated values, otherwise the parameters would not be statistically significantly different from zero, and should be eliminated. For the two forecasting models summarized, all of the parameters are statistically significantly different from zero, except for the intercept (the estimate of the mean value of the differenced series, in this case), and the parameter ar.S.L48 for DMA 1. It is reasonable that the intercept should be close to zero for a differenced series that is stationary. In addition for DMA 1, the seasonal autoregressive terms could be reduced to order 1 without harming the forecasting model. In subsequent analyses, the forecasting models indicated in Figure 9 were unchanged, so that for both demand series the forecasting model format is the same, in case it is useful ultimately to rely on relatively stable forecasting model forms for DMA demand forecasts.

The residuals tests summarized in Figure 10 are used to understand the degree to which model residuals have the character of white noise (i.e., no temporal correlations and normally distributed). The basic idea is to begin with a real data series exhibiting a complex autocorrelation structure, and if it is modeled correctly (i.e., the model would have the same autocorrelation structure as in the data), then the difference between the model and the data — the residuals — should be uncorrelated. If significant temporal correlations are left in the residuals, or if the residuals were significantly biased, then additional forecasting model forms that could represent those correlations or biases are needed.

For a visual overview of model residuals' characteristics, they are plotted in the upper left pane

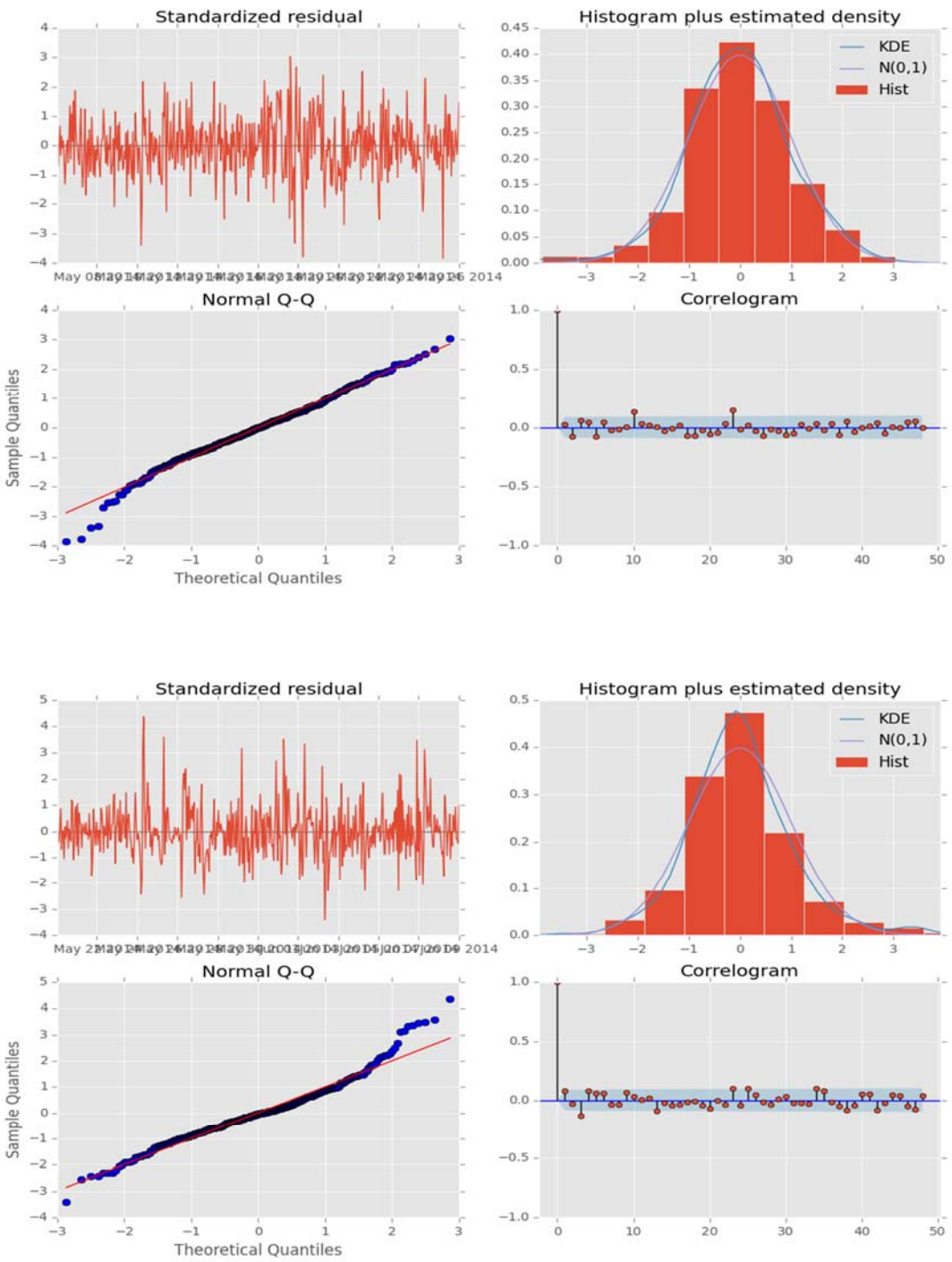

Figure 10. Model residual test statistics for DMA 1 (top) and DMA 2 (bottom).

in Figure 10. The upper right pane compares the histogram of the normalized residual values to an ideal normal distribution, and the lower left pane shows a Q-Q plot where, if the residuals are

normally distributed, all of the quantiles should line up on the red line. Finally, the lower right pane is the autocorrelation function of the residuals, which should ideally show that the residuals are statistically uncorrelated at all lags; in these cases there are a small number of lags that show autocorrelations of mild statistical significance, as they extend slightly beyond the 95% confidence region (indicated in blue). For the models shown for the two DMAs, the comparison of residuals to normality are reasonable except that there are some deviations in the tails, as shown by the samples deviating from the 45 degree line on the normal quantiles plot at large deviations from the mean. The autocorrelations show that most of the autocorrelation structure in the data has been captured, yet there appears to be statistically significant residual correlations that remain at lags of roughly one day, and perhaps at shorter or longer lags too. This indicates that the best statistical model form might not have been found yet, and perhaps additional seasonal periods should be built into the forecasting model.

At this point it can be concluded that two reasonable forecasting models have been determined for both DMA demands, sufficient for prototype testing of forecast accuracy. It should not be concluded, however, that these represent the best statistical model that can be determined. As previously indicated, additional seasonal periods should be considered, or alternative ways to represent multiple seasonal periods in the most efficient manner. In addition, not every different possible combination of forecasting model orders were considered during this prototype evaluation. Most significantly, though, these forecasting models have not considered exogenous regressors, even though the SARIMAX model form allows for them.[4] Exogenous variables should be considered as a way to improve the forecasting model accuracy, in light of the many studies that have correlated water demand with various factors, notably weather variables such as temperature, humidity, and precipitation. Weather variables are interesting also for their ability to be leveraged into demand forecasts, because their forecasted values are available in real-time from independent sources. Thus, if it can be established that weather variables can usefully explain a portion of demand variability, then they can presumably, and practically, be used to improve forecasts as well. Exploration of this inclusion is, however, left to future work, but the ground work for such exploration is supported by the work described here.

The forecasting models can be used efficiently to forecast demand for any distance into the future, even though logic says that the forecast accuracy could suffer the farther into the future one looks[5]. There are two main types of forecasting, *in-sample* and *out-of-sample*. In-sample forecasting deals with predicting demand values (in this case) within the range of the data. Out-of-sample forecasting involves using the forecasting models to predict outside of the data range. In addition, there is what is termed "*dynamic*" prediction. In dynamic prediction, as the forecasting model forecasts into the future, it is relying on its own predictions as opposed to actual data. Thus, out-of-sample forecasting is always dynamic; there is no data available so

[4] An exogenous regressor is a variable that is uncorrelated with, or has zero covariance with, the random error term.

[5] Interestingly, the method of forecasting used here is the underlying Kalman filter, of course with the coefficients that have been estimated using the StatsModels/SARIMAX module. The existing Python StateSpace project toolchain is leveraged efficiently for this prototyping analysis, and can be leveraged just as easily within EPANET-RTX using the Python linkages that were built.

dynamic forecasting is all that would be possible. However dynamic forecasting can also be conducted in-sample, which is useful for comparing the results of longer term forecasts with data.

Figure 11 shows illustrative results for non-dynamic (one-step-ahead) and dynamic (out-of-sample) forecasting. The dots are the actual demand data, and the blue line represents in-sample non-dynamic forecasting. This is labelled as “one-step-ahead” forecasting because, being non-dynamic, it moves one step at a time, using the new data points that become available to it instead of its own historical predictions. The red line starting after approximately 4 days represents a one day period of dynamic forecasting, thus illustrating the accuracy of the forecasting model for predicting 24 hours into the future. But, the dynamic forecast is done in-sample, so that it can be shown along with the resulting data. The dashed grey lines show the 95% confidence intervals for the forecasts, which are a direct result of using the statistical modeling framework that yields not only the estimated parameter values, but also their estimated standard errors. The confidence intervals are a way of translating difficult to interpret information about the uncertainty in parameter values into more practical and easy to interpret information about the uncertainty in the demand forecast. Such confidence intervals can ultimately be propagated further through the network model predictions to yield similar confidence intervals in the hydraulic and water quality states that depend on the uncertain forecasted demands. In summary, these results give a first and preliminary impression about the level of accuracy that could be expected for 24 hour aggregate demand forecasts, and provide proof via illustration that that such estimates are feasible to use for driving real-time network simulations.

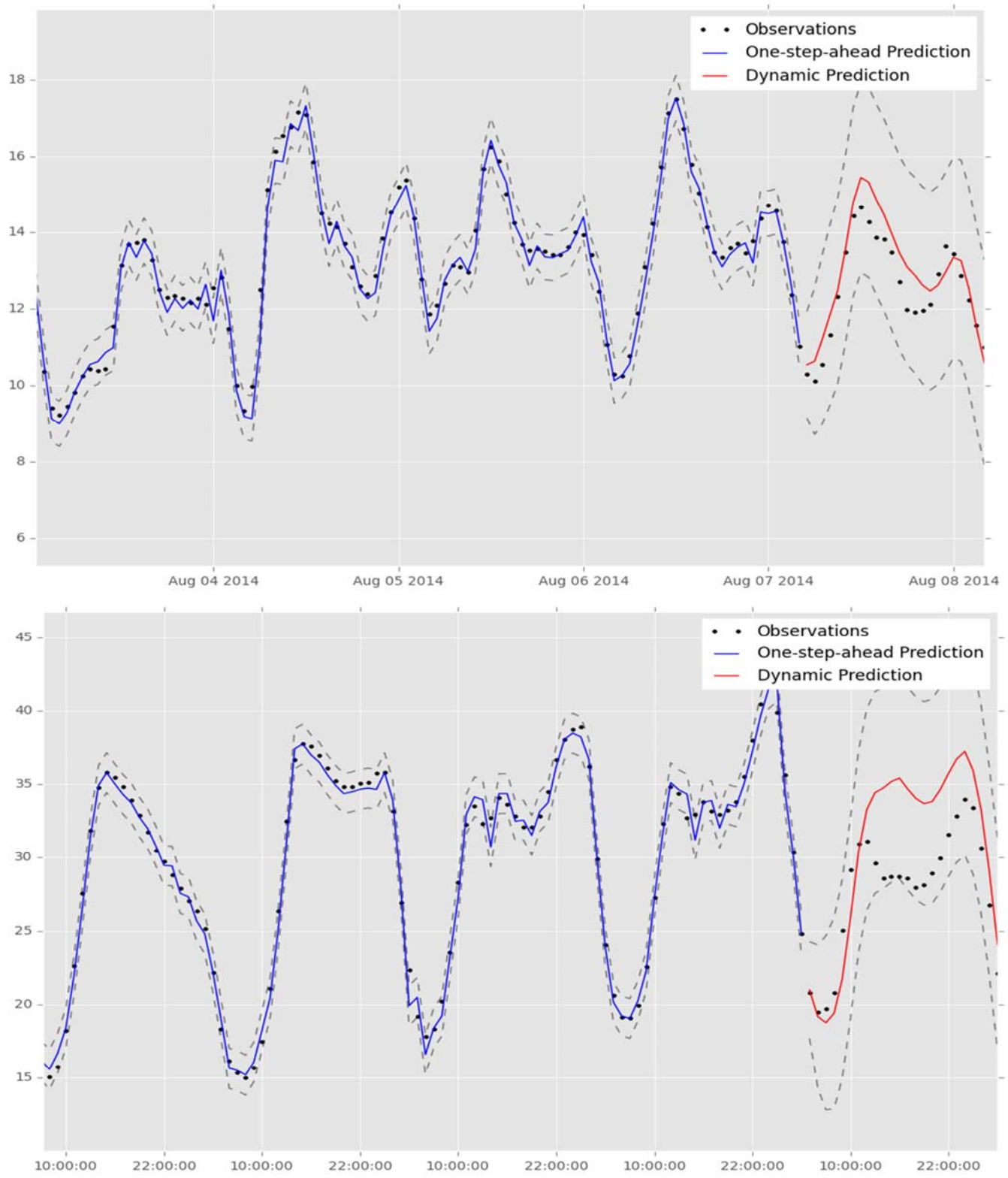

Figure 11. Illustrative dynamic (out-of-sample) demand forecasts for DMA 1 (top) and DMA 2 (bottom). The blue line shows in-sample one-step-ahead forecasts for the first four days, followed by a 24 hour dynamic forecast shown by the red line. The dynamic forecast does not use data after it begins, and data points are shown only for comparison to the forecast. Forecast confidence intervals (95%) are shown by the grey dashed lines.

A preliminary assessment of demand forecast accuracy was performed, using the same demand forecasting models, but with an additional 74 and 45 days of demand data for DMA 1 and DMA 2, respectively. This mimics a practical real-time scenario where the time-series model parameter values would have been estimated using 3 weeks of prior demand data and then used — without modification to the parameter values — to forecast demand for the ensuing additional data period. For simplicity of analysis and explanation, 24 hour forecasts were conducted every midnight during the 74 or 45 day evaluation period, and those forecasts were compared with the data to quantify errors. These errors are then summarized as a percentage mean absolute error (MAE), as a function of time-of-day (TOD) (or, forecast distance from midnight); the percentage is calculated relative to the TOD mean from the actual demand data set. Figure 12 shows the results for both DMA 1 and DMA 2. The upper pane is the percentage MAE forecast error as a function of distance from midnight for the evaluation period, and the lower pane is the TOD mean calculated from the data for the same period. Errors were reported ranging from 2 to 16 percent depending on forecast distance. The forecasting models did not show a uniform tendency for percentage forecast error to degrade with forecast distance. (With improvements in percentage error being associated with time periods of higher demand, it could be that the error magnitude, and not the percentage error, might degrade uniformly with forecast distance.).

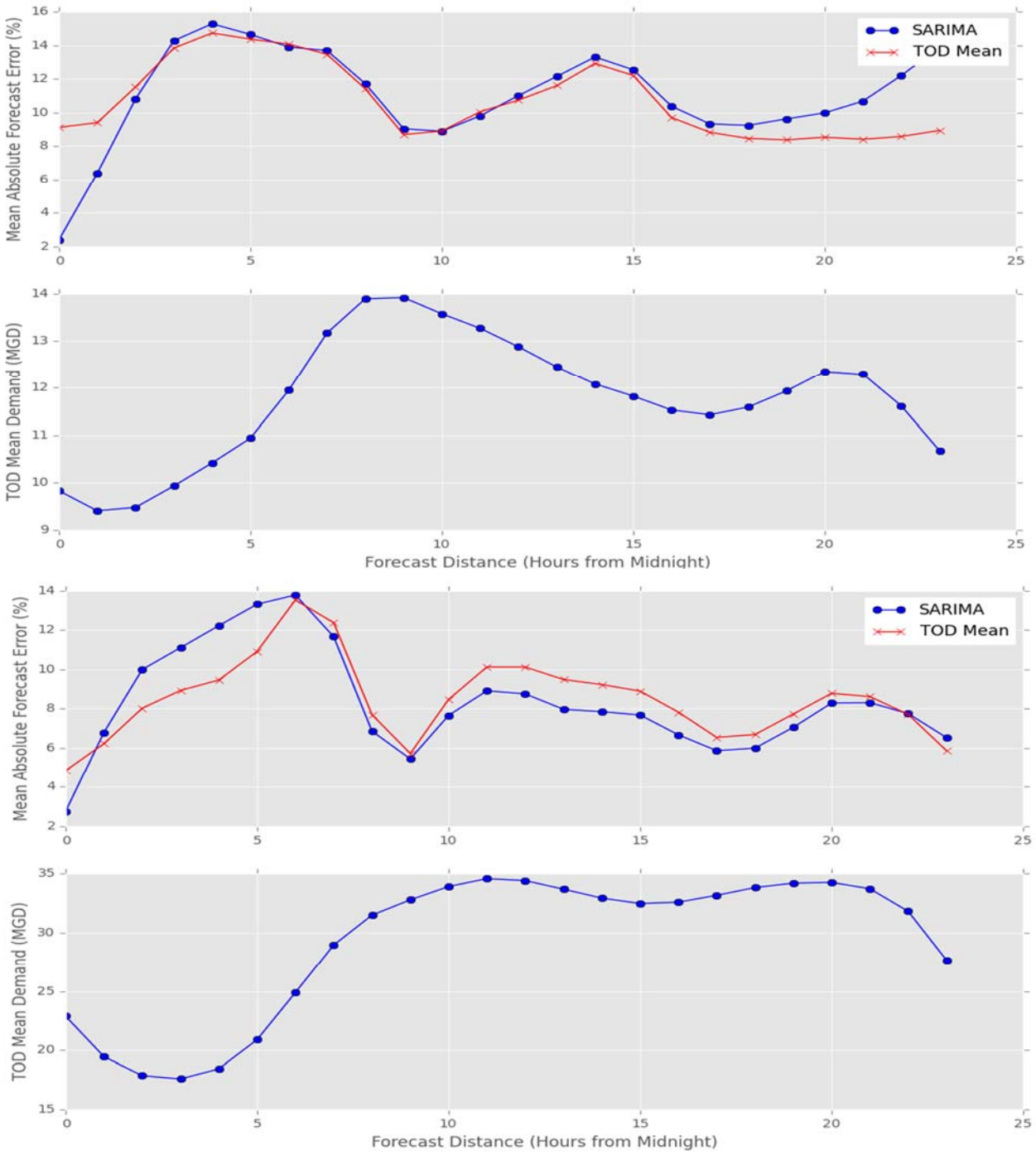

Figure 12 — Forecast errors and time-of-day (TOD) mean demand as a function of forecast distance for DMA 1 (top) and DMA 2 (bottom). Dynamic forecasts of up to 24 hours were constructed at midnight for a sample period of 74 days (DMA 1) and 45 days (DMA 2). Mean absolute forecast errors are quantified as a percentage of the time-of-day (TOD) mean demand at each forecast distance. SARIMA model forecast errors are compared to forecasts using a simple TOD mean model, showing that for the particular SARIMA model used here (without exogenous variables) the simple TOD mean model is competitive.

Also shown in Figure 12 is a comparison of the SARIMA statistical model forecast with a conceptually and mathematically simple forecasting approach — simply using the TOD mean demand as a forecast for any future value. It was somewhat surprising that this simple forecasting model compared very well with the fitted SARIMA model. It is too early to draw conclusions from this prototype analysis about the need for more sophisticated forecasting models, or the ability to use very simple ones. It is, however, useful to learn that one can and should use such simple forecasting models as a yardstick for measuring future improvements in forecasting performance, such as those assumed to be possible through the judicious selection and use of exogenous weather variables. Finally, even if simple forecasting models can be used in this case for accurate forecasting, they are not easily integrated with exogenous variables, and can be assumed to be limited to situations where strong periodic forcing is present in the data signal. Periodic forcing refers to the underlying processes that would create periodicity in the demand, which is expected for water demands.

Small-Scale Field Application of EPANET-RTX

The ultimate goal of the open source project of EPANET-RTX is to provide the water community with modeling tools that enable optimized water usage based on use of real-time hydraulic (e.g., pressure, flow, and tank level data) and water quality data. During the development processes, EPA has engaged large and small utilities to test and improve the functionalities of EPANET-RTX. In 2012 and 2013, we conducted our first field scale evaluation of an EPANET-RTX-based real-time hydraulic and water quality model. The evaluation was conducted on the Northern Kentucky Water District (NKWD) water system (Uber et al., 2014).

In September of 2014, the EPANET-RTX project team began working with the City of Milford, Ohio and their water system representatives. Milford's water system is a small system serving approximately 7,000 customers. We adapted the City of Milford's water utility network model to a SCADA data driven, real-time model using the EPANET-RTX libraries (See Figure 13). The

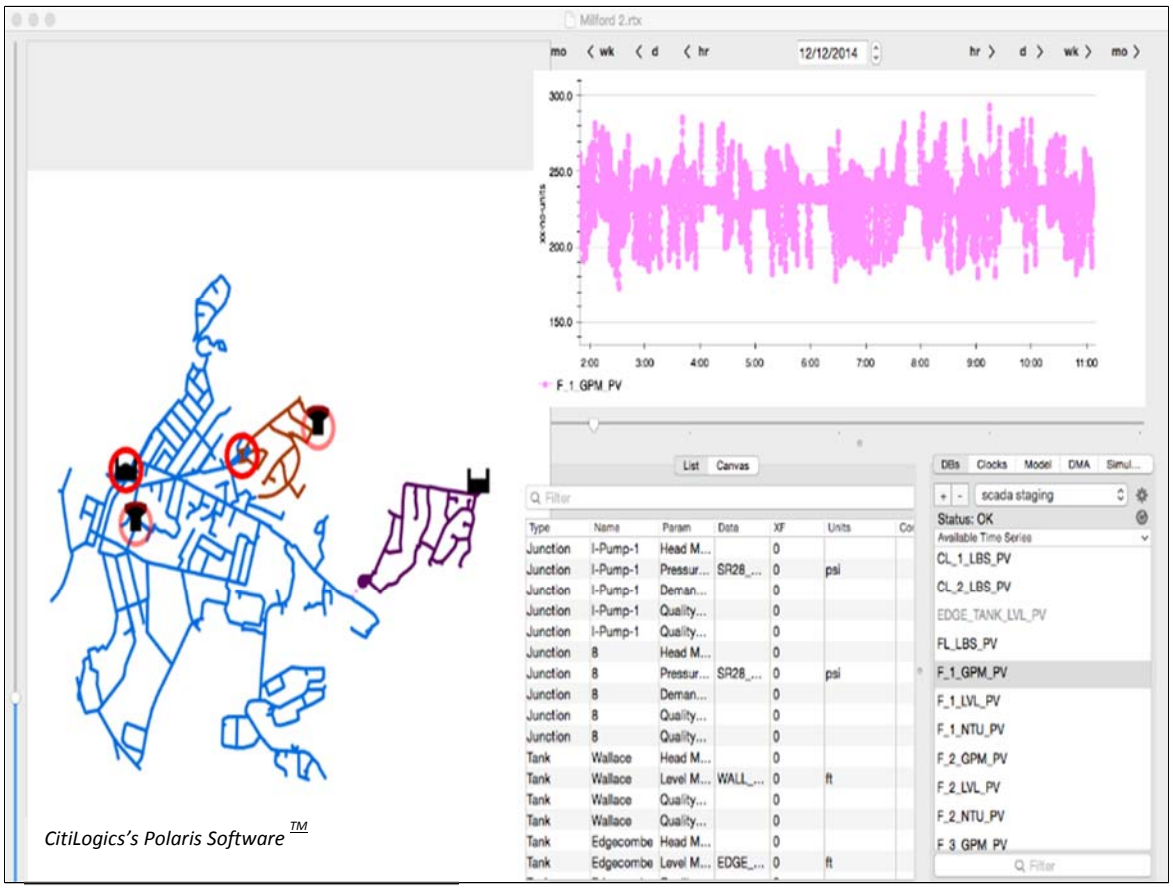

Figure 13. City of Milford's EPANET-RTX-based Real-Time Model

newly configured real-time model is being used to investigate possible improvements to the utility's water system operations. The study is planned to be completed in 2016. In 2015, we demonstrated to the City of Milford how EPANET-RTX-based analytics were used to fuse the city's infrastructure model with their real-time, sensor data. The infrastructure model and data fusion process identified a number of issues and, as a result, made significant improvements to

the existing model's accuracy. We highlighted to the City of Milford the potential benefits that could be achieved through a continuous understanding of system operations using real-time analytics, e.g., improved water quality and identification of water losses. Field sampling and real-time measurements began in 2015. These sampling and measurement results will be used to help identify and investigate possible additional improvements to the infrastructure model and possibly identify opportunities to improve operations.

For More Information

EPANET-RTX is an open-source software development project intended for programmers interested in water distribution system simulation and water distribution system engineers interested in programming. There are various ways to get involved in the project, including connecting to the code repository, reviewing coding conventions, and using the issues tracker to make a feature request and communicate with the developers. To learn more, visit the OpenWaterAnalytics website (http://openwateranalytics.github.com/epanet-rtx/)

Contact Information

For more information, visit the EPA Web site (http://www.epa.gov/nhsrc/)

Technical Contacts: Robert Janke (janke.robert@epa.gov)
Michael Tryby (tryby.michael@epa.gov)

General Feedback/Questions: Kathy Nickel (nickel.kathy@epa.gov)

If you have difficulty accessing this PDF document, please contact Kathy Nickel (Nickel.Kathy@epa.gov) or Amelia McCall (McCall.Amelia@epa.gov) for assistance.

U.S. EPA's Homeland Security Research Program (HSRP) develops products based on scientific research and technology evaluations. Our products and expertise are widely used in preventing, preparing for, and recovering from public health and environmental emergencies that arise from terrorist attacks or natural disasters. Our research and products address biological, radiological, or chemical contaminants that could affect indoor areas, outdoor areas, or water infrastructure. HSRP provides these products, technical assistance, and expertise to support EPA's roles and responsibilities under the National Response Framework, statutory requirements, and Homeland Security Presidential Directives.

Disclaimer

The U.S. Environmental Protection Agency through its Office of Research and Development funded, managed, and collaborated in the research described here under EPA contract EP-C-10-060. This technical brief has been subjected to the Agency's review and has been approved for publication. Note that approval does not signify that the contents necessarily reflect the views of the Agency. Mention of trade names, products, or services does not convey official EPA approval, endorsement, or recommendation.

References

Boulos, P., Jacobsen, L.B., Heath, J.E., and Kamojjala, S. 2014. “Real-time modeling of water distribution systems: A case study,” *J. Amer. Water Works Assoc.* 106 (9):E391-E401.

Kara, S., Karadirek, E., Muhammetoglu, A., and Muhammetoglu, H. 2015. “Real-time monitoring and control in water distribution systems for improving operational efficiency,” Desalination, and Water Treatment, DOI: 10.1080/199443994.2015.1069224.

Rossman L.A. 2000.EPANET 2 Users Manual. Cincinnati, Ohio: U.S. Environmental Protection Agency, Office of Research and Development, National Risk Management Research Laboratory. EPA/600/R00/057

U.K. Water Authorities Association. 1985. “Report 26 Leakage Control Policy and Practice,” Technical Report, 198. Reprint of original report: Technical Working Group on Waste of Water. 1980. NWC/DoE Standing Technical Committee report number 26. London: National Water Council.

Uber, J., Hatchett, S., Hooper, St., Boccelli, D., Woo, H., and Janke, R. 2014. Water Utility Case Study of Real-Time Network Hydraulic and Water Quality Modeling Using EPANET-RTX Libraries. Cincinnati, Ohio: U.S. Environmental Protection Agency. EPA 600-R-14-350.