

# Detecting Cooling Coil Fouling Automatically— Part 2: Results Using a Multilayer Perceptron

**Daniel A. Veronica, PhD, PE**

Member ASHRAE

*Received July 29, 2009; accepted June 3, 2010*

---

*A new concept to automatically detect cooling coil fouling is introduced in a companion paper (Veronica 2010). The concept maintains a data-driven dynamic model of the cooling coil on-line with coil instruments, so all steady and transient coil behavior is assimilated. The fouling surveillance is to periodically apply a specific transient sequence (the query) to this model off-line as a surrogate for applying such a test to the real coil. A query sum, calculated from the transient response of the model to a query, is used to detect changes in the real coil, to discern fouling from other faults such as instrument drift, and to further discern water-side from air-side fouling. A crucial element to viability of the concept is that the data-driven coil model respond to queries dynamically and recursively, with sufficient stability and accuracy so that fouling can be discerned. Yet, the model should be of reasonably simple architecture, permitting a plausible embedded implementation. Based on trials using simulator data, a relatively simple, three-node, multilayered perceptron demonstrates sufficient accuracy and stability so that the concept can be regarded as viable and as warranting further development. Also presented are the factors found to be critical for dynamically employing a data-driven model of any architecture to detect process faults.*

---

## INTRODUCTION

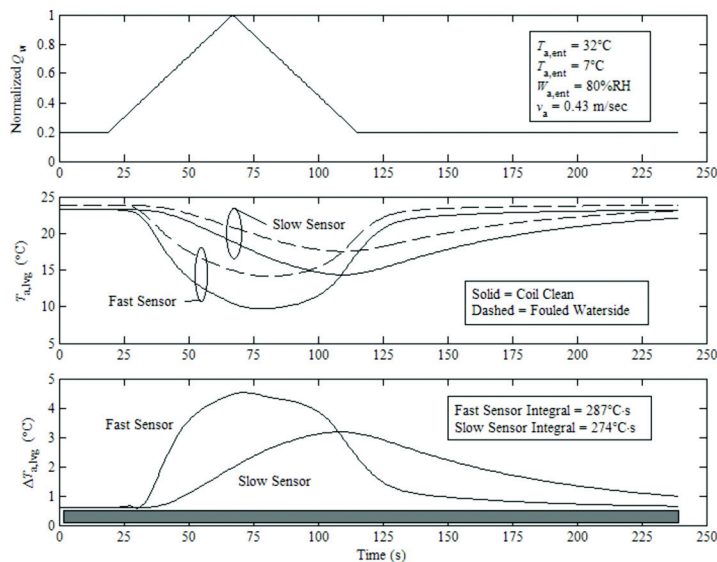
The companion paper (Veronica 2010) introduces a new concept to automatically detect cooling coil fouling by combining the strengths of data-driven and model-driven approaches to fault detection and diagnosis (FDD). The decision that classifies the coil at any particular time as being clean or fouled is made by an autonomous supervisory program called the *agent*, which internally employs (as needed): (1) a real-time, data-driven component; and (2) an embedded lookup table that is the product of many simulations previously run off-line in an independent, model-driven component. Fouling surveillance consists of the agent exercising periodic transients (queries) on a data-driven dynamic model that characterizes the coil thermal process as a surrogate for testing the real coil.

## Fouling Surveillance Using Dynamic Response of a Data-Driven Model

A fouling surveillance query passes a sequence of input vectors through the data-driven coil model as a time-series, off-line from the incoming stream of real-time instrument data. The coil water flow values in the sequence follow a triangular pulse shape as depicted in Figure 1, where the top plot panel shows the pulsed water flow input, and a text box lists the constant values of the remaining inputs, which fix the query to a particular point in the state space of coil inputs. The water flow pulse elicits a transient response in the model output (the temperature of air leaving the coil) as seen in the middle panel of Figure 1. This response is compared value-by-value

---

**Daniel A. Veronica** is a mechanical engineer at the National Institute of Standards and Technology, Gaithersburg, MD.

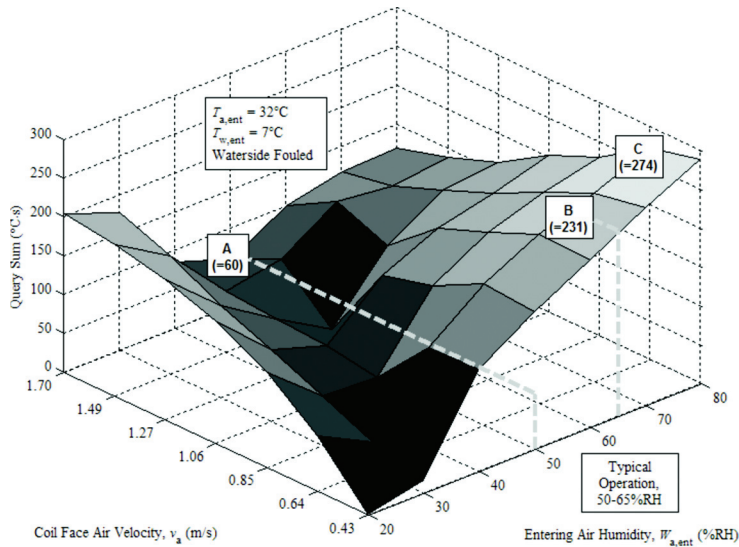


**Figure 1. Response of clean and water-side fouled cooling coil to pulse in water flow.**

to a baseline response to the same query, which was archived earlier when the coil was known to be clean. The value-by-value comparisons may yield a sequence of differences, such as those plotted in the bottom panel of Figure 1. The quantity used as the basis for fouling detection is the query sum, the summation of all differences over the length of the sequence. This approximates integrating the area under the difference values plotted in the bottom panel of Figure 1. A significantly nonzero query sum obtained from a surveillance query indicates the coil model has changed since the baseline query for the same point of input space. But, a nonzero sum from one surveillance query alone does not necessarily imply fouling in the real coil caused the change. Drift in the leaving air temperature measurement, for example, would also change the model.

The contribution of first-principles modeling to the concept is that query sums due to air-side and water-side fouling of the coil each exhibit distinct shapes when considered over the state space of coil inputs. Veronica (2010) shows this distinction is characterized adequately by an ensemble of three-dimensional surfaces for each mode (i.e., air-side or water-side) of fouling. Each surface exists over the state plane of humidity and flow rate of air entering the coil, as shown in Figure 2. Each ensemble requires at least four such surfaces, one for each permutation of the high and low values of entering air and water temperatures expected during coil operation. An ensemble can be embedded within the agent as a lookup table. Query sums for intermediate values of entering air or water temperature are obtained by interpolating linearly within the table.

Given the human expert information about fouling that the lookup table provides, the agent follows up a nonzero surveillance query sum with further complementary queries of the model at other points of input space where the table holds query sums significantly different from that predicted for the point of the surveillance query. If the ratios obtained between complementary query sums are similar to the ratios between values predicted in the table for the same points of input space, evidence now implicates fouling of the real coil as likely causing the data-driven coil model to change.



**Figure 2. A summation of surface for water-side fouling.**

## A Radical Departure from Prior Research

During queries the data-driven model is employed both dynamically and recursively. This is a radically different application of data-driven modeling than seen in prior FDD research by the HVAC community, which to date has generally employed data-driven predictions non-recursively to estimate steady states. Exercising brief test transients upon a real process to identify properties from its dynamic response is a well-understood practice in HVAC (e.g., step reaction curves). But, apparently no HVAC research has put forth a FDD concept applying such tests to data-driven models as a surrogate for test transients on the real process. None of the FDD work in the 41 papers of the International Energy Agency Annex 25 research of the mid-1990s (Hyvarinen 1996), or subsequent FDD work in *ASHRAE Transactions* or *HVAC&R Research*, explores the possibility.

One challenge is that during the query the model is necessarily off-line, and thus, cut off from any exogenous (i.e., error correcting) data sampled from real instruments (or, in this paper, sampled from a first-principles simulation). The situation is roughly analogous to deduced (ded or dead) reckoning of ship position in the era of celestial navigation. Each position predicted during a period of extensive cloud cover was an estimate propagated from past estimates since no sun or star fix, which would bring in a datum exogenous to the estimation method itself, was possible. Here, the query to detect coil fouling must occur under a data cloud cover. Upon each of 35 steps in response to a query, the model recurs its output (i.e., air temperature leaving the coil,  $T_{a,lvg}$ ) prediction back to its input to enable the next prediction. The potential for error accumulation and instability is obvious. When on-line with the sampled data, the model has the comparatively easier task of being a process estimator, which is to predict a single  $T_{a,lvg}$  value only one time step ahead (step  $n + 1$ ) of its input at any time step  $n$ . The analogy then is to get a sun or star fix at every position predicted under clear skies, with the residuals used to update the estimation parameters (sea current, etc.) for when clouds do intervene. Similarly, here coil process estimation by the data-driven model keeps it

updated with developments in the coil (e.g., fouling). However, faults are not deduced directly from estimates of steady-state parameters as in past HVAC research on FDD (e.g., Lee et al. 1996a and 1996b; Peitsman and Bakker 1996; Li et al. 1996; Diaz et al. 1999), but are instead deduced from dynamic response of the coil model to queries. Merits of this new FDD concept are presented in the companion paper (Veronica 2010). A relatively simple data-driven model meeting this challenge is addressed in the sections that follow.

## MODELING COOLING COIL DYNAMICS USING A MULTILAYER PERCEPTRON

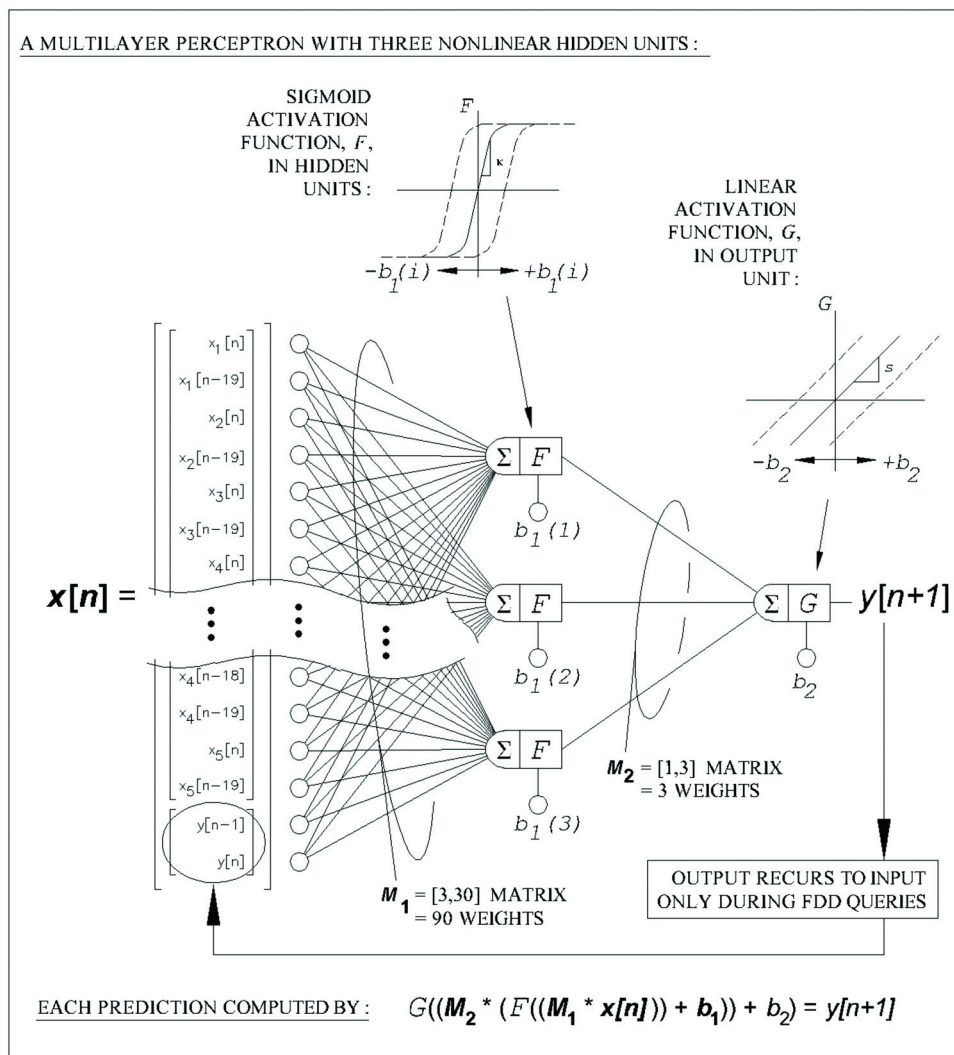
The concept requires a data-driven coil model with architecture able to represent the dynamic response of a real coil to a pulse in water flow. Veronica (2002) describes success at regulating air temperature leaving a real, 4720 l/s (10,000 cfm) air handling unit (AHU) by using a multilayer perceptron (MLP) as a data-driven dynamic model of the unit's cooling coil. The artificial neural network control method introduced by that work, called *emulator differencing*, places recursive demands on the model that are similar to those required here. That suggested that the same MLP architecture, depicted in Figure 3, be used for preliminary trials to prove the new FDD concept is fundamentally viable. Though relatively simple, this configuration predicted air temperatures one 20 s time step ahead of current data from a real, full-sized (2.1 m<sup>2</sup> [22.6 ft<sup>2</sup>] face) cooling coil typically to within 0.25°C (0.45°F).

### Configuring a MLP for the Tasks Required

Multilayer perceptrons are well addressed by many foundational texts, such as Bishop (2006). Figure 3 is a diagram of the MLP employed here. The cooling coil thermal process is computationally regarded as mapping input sensor (instrument) data to output sensor data. At each discrete time step  $n$ , a multivariate vector  $X$  is composed of 30 elements (for clarity, fewer are depicted). The elements include current and specified past values of coil process input quantities  $x_1, x_2$ , etc. (e.g., temperature of entering air and water, humidity of entering air, airflow rate, and water control valve position), and current and past values of the coil output (temperature of air leaving the coil). Each quantity is normalized to offer a wide dynamic range under typical AHU operation, while ensuring no value goes above 1.0 or below a small, positive nonzero amount. The MLP maps vector  $X$  to a scalar output  $y$ , which predicts the coil output at step  $n + 1$ . That is, the mapping has a time horizon of one sample period. This feature allows the MLP to assimilate sensor data sampled when the coil is not in thermal equilibrium. Since the mapping bridges a time differential like that of underlying finite-difference equations, the mapping can model both steady-state and dynamic coil behavior.

Two quantities from the output of the coil, the temperature of air leaving the coil at time step  $n$  and  $n - 1$ , are also elements of the input vector. The MLP is thus tasked to map the coil process ahead not only by differential time, but also by differential output, as would finite-difference equations representing the coil. That is, it is known from first-principles that a dynamic input-output mapping of the coil is possible using the information given to the MLP.

The MLP principally performs two nonconcurrent tasks. One task is on-line process estimation, which requires assimilating coil input and output values as real-time instrument samples taken during AHU operation. The purpose is to keep the MLP temporized to the current condition of the real coil and ready to answer queries validly. An optimization routine uses sampled data to continually test the input-output mapping within the MLP. The MLP parameters are adapted (trained) so the mapping represents (learns) an optimal generalization of the thermal process, evident from the more recently sampled data. Optimization occurs by backpropagation of error, a method described well by Mitchell (1997). The metric applied to MLP training is root mean squared (RMS) error taken over the validation data, a partition of all the data acquired that



**Figure 3. A multilayer perceptron modeling a process.**

is used only to test generalization ability. While on-line, the MLP receives the two air temperatures' output values in its input vector from the leaving air temperature sensor. That is, all 30 values input to the MLP during process estimation come from instruments (real or simulated), so they never recur from past output.

The second task is performed off-line from the stream of sampled data, and it is to respond to queries from the agent. During this task the MLP input vector never contains sampled data. Upon each step of a query the vector is populated with 28 values specified to locate and define the query in coil input space, plus two values of the MLP output (leaving air temperature) recurred from the two preceding steps of the query. As with any dynamic model tasked to propagate temporally, these output values must be initialized to begin recursions. Initialization comes from historic data acquired when the MLP was on-line in the input space region defined by the query. The fact that

any steady offset is eliminated when a query sum is calculated (detailed in the companion paper [Veronica 2010]) helps overcome bias errors incurred by that approach.

Free parameters in the MLP include two matrices ( $M_1$  and  $M_2$ ) that form a network of connective weights that link a hidden layer of three parallel nodes to, respectively, the input vector and the output node. The hidden layer nodes have identical sigmoid activation functions. The remaining free parameters are two bias quantities (vector  $b_1$  and scalar  $b_2$ ), which slide the activation functions along their abscissas, providing added degrees of freedom to minimize output errors.

Issues Crucial When Modeling Coil Dynamics from Sampled Data

Like other physical processes described well by first-order lag models (belonging to the class of low-pass filters in signal processing), an HVAC coil responds to cyclic control valve (i.e., water flow) input with decreasing output gain (peak-to-peak amplitude of leaving air temperature) as the frequency of the input cycle increases. The model architecture must capture this frequency-dependent character adequately. Otherwise, high frequency noise carried in the output (e.g., modeling error) will recur to the input at significant gain, making the output predictions unstable.

An illustration of the low-pass filtering exhibited by a real cooling coil is seen in Figure 4, which consists of two plot panels. The top panel shows, in the time domain, an increasing input frequency of control valve cycles,  $Z_V$ , plotted along with the responding cycles in coil leaving air temperature (i.e., coil output). The loss of coil output gain as input frequency rises is plainly seen. The resulting spectral character is illustrated by a plot in the frequency domain, called a *Bode plot*, seen in the bottom panel of Figure 4. The Bode plot also establishes the maximum time period allowed between sensor samples assimilated by the MLP.

The rate at which data assimilated by a MLP is sampled from sensors (or is simulated as coming from sensors) is not arbitrary. Any sampling frequency,  $f_S$ , selected for data acquisition (having a

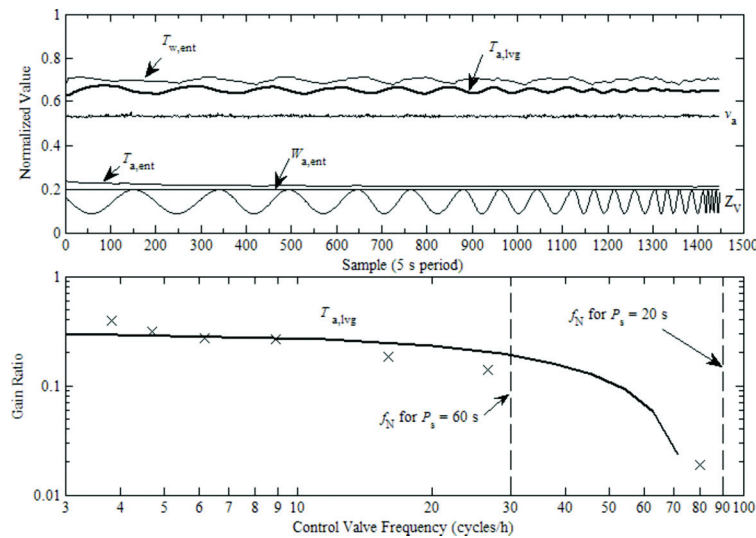
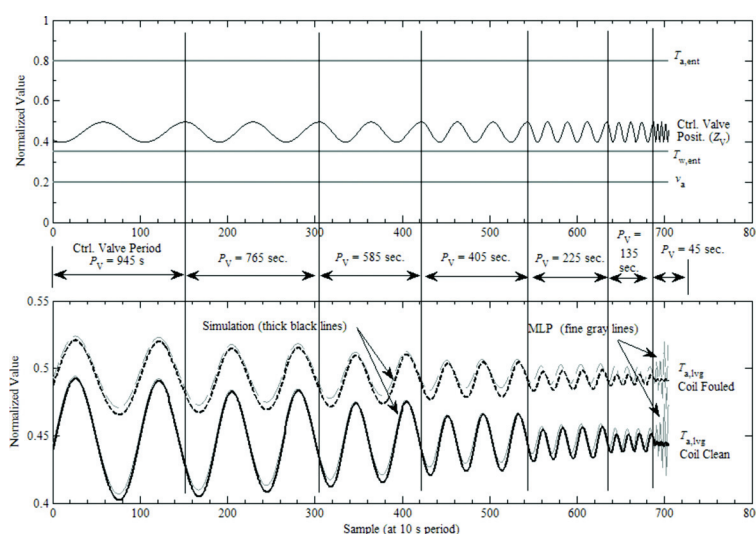


Figure 4. Dynamic response of cooling coil in time domain (top) and frequency domain (bottom).

period  $P_S$ ) implies a corresponding Nyquist frequency ( $f_N$ ) that is half the sampling frequency. All dynamic behavior a process quantity exhibits at frequencies above the Nyquist frequency will sample into the data record only as a lower frequency alias of its true frequency. To model a cooling coil for fouling detection, the gain ratio of leaving air temperature amplitude over coil water flow (or, control valve) amplitude is a critical quantity, since the fouling detection query involves only those two variables dynamically. As this gain ratio falls to insignificance with rising control valve frequency, aliasing becomes irrelevant to modeling the coil because valve modulation has lost its practical effect on leaving air temperature. Dashed vertical lines on the Bode plot of Figure 4 show a sample period of 20 s puts the Nyquist frequency at 90 cycles/h, frequent enough to capture all significant dynamics of the real coil. A longer sample period of 60 s would put the Nyquist frequency at 30 cycles/h, which aliases some coil character having significant gain. The three other coil input quantities (entering air temperature, entering air humidity, and entering water temperature), though static during the query, are also relevant, as they affect sensitivity of the coil response. The Nyquist frequency must be satisfactory given the most responsive coil condition expected. That requirement establishes the maximum interval in real time allowed to be the data sampling period and the horizon mapped into MLP predictions.

Training in the frequency domain must at least result in the MLP being stable around the frequency of the water flow pulse (i.e., the period of control valve motion) present in the query. The top panel of Figure 5 shows control valve oscillations of rising frequency being applied to a cooling coil, as represented by a first-principles numerical simulation and a MLP model. In the bottom panel, the response in leaving air temperature of both the simulation and the MLP exhibit the falling output gain necessary to represent the coil dynamics properly. Only the MLP, though, becomes unstable at the highest frequency step (a 45 s valve period). The problem is not aliasing, as that frequency is still within the Nyquist limit implied by the 10 s period at which this MLP samples data. The instability is caused instead by such a high frequency not having been present in the data used to train the MLP.



**Figure 5. Spectral response of MLP compared to a simulation.**

There is no minimum real time interval applying to the data sampling period, but there is a growing cost incurred as the time horizon for MLP predictions is made shorter. The horizon must be far enough ahead of the current sample of a dynamic quantity that a nominally slow change in the quantity will be significant compared to MLP uncertainty (e.g., the internal noise of MLP modeling error). A horizon shorter than that means the MLP cannot discriminate the nominal slow change from no change at all. The issue is addressed in more detail with diagrams in Veronica (2008). Specifying the horizon length is thus a heuristic barter between not masking coil dynamics slower than an acceptable rate and not aliasing fast dynamics having significant gain.

The filtering evident in dynamic (nonequilibrium) data sampled from a real coil results from thermal capacitance and resistance constituting a memory of past energetic states within the coil. This memory extends back in time with diminishing significance. To approximate that effect computationally, the MLP input vector is structured as a tapped-delay line (TDL) of time-associated values of input quantities (e.g., states of entering air and water) and output (temperature of leaving air) from the real coil. The TDL is a circular buffer, or shift register, holding samples of a multivariate time series, a matrix of coil instrument data, moving in time. Upon each time step acquiring real-time data (or extracting archived data), a new round of coincidentally sampled instrument readings is shifted into the front of the buffer, while the oldest values held are shifted out the back. Propagating a prediction by the MLP involves the buffered values being weighted, summed, and transformed by the hidden layer of the MLP based on their age (e.g., column index) and physical property (row index) in the tapped delay line.

## TRAINING THE MLP ON SYNTHESIZED DATA

### Training Arranged to Test Query Performance Alone

Nothing in this paper addresses the task of the MLP assimilating sampled data on-line and learning, or maintaining, a real-time dynamic coil model. Although that task is vital to the FDD concept, it is not as primary as the task of the MLP predicting output (leaving air temperature) recursively throughout a query with stability and accuracy sufficient so that coil fouling can be identified from the resulting query sum. Until a MLP demonstrates that task satisfactorily, there is no point in addressing on-line learning. The wealth of existing research addressing on-line learning (cited in the introduction) has come about with no complementary research that uses a MLP recursively as a surrogate for a real process subject to system identification. The recursive prediction task appears to present more unknowns. To isolate it, the on-line learning task is supplanted by positing the two end cases involved (one coil clean and the other coil fouled), with a separate MLP for each. Queries are run using both as though they were the same MLP presenting itself differently on two occasions separated in time. To ensure this approach is sound, training and validation of each MLP must be equivalent.

Both the clean MLP and the fouled MLP are trained and validated using the same individual pairings of model inputs to output (example pairs for the MLP mapping), presented in the same order, with the same training parameters in effect. Samples of acquired data are synthesized by first-principles simulations, generating four separate outputs (clean, air-side fouled, water-side fouled, and both sides fouled) for each of two cases of leaving air temperature sensor response (fast and slow), for a total of eight concurrent output streams. Thus, eight MLP are trained and validated in parallel, all to represent how one MLP would present itself for query under different circumstances. The severity of fouling simulated is the nominal amount defined in Veronica (2010). A routine pilots the simulation around a dynamic input space of AHU operations, including regulator control with setpoint changes, steady states, and imposed transients. Values of entering air and water temperatures and air humidity are invoked randomly within scheduled ranges. Given the

parallel synthesis of eight output data streams, followed by parallel training of eight MLP, differences of MLP performance in the plots that follow are attributable only to the responsiveness of the leaving air temperature sensor and the location and impact of coil fouling.

## Details of Data Synthesis and MLP Training

Data samples are synthesized using Zhou and Braun's (2007) dynamic cooling coil simulation, as modified by Veronica (2008), to incorporate fouling in the coil. Each sample is one round of coincidental values obtained from the simulation, including five coil process inputs and one output (leaving air temperature). Synthesizing many hours of coil operations results in thousands of samples, handled in nonoverlapping units (sequence blocks) of 40 consecutive samples at a 10 s period. The sequence blocks are melded into varied training sets, with each set deliberately constructed to have a specific dynamic content. Typically, 100 to 400 sequence blocks (i.e., 11 to 44 h in simulation time) constitute one training set. Thus, the performance of a particular MLP can be correlated to the content of its training set, as well as its configuration.

Data are synthesized by a simulation stepped at relatively short, 1 s intervals. Sampling is to parse the 1 s period simulation data at the 10 s period representing data acquisition by sensors, collecting the samples into sequence blocks. The higher the index is of a sample in a sequence block, the more recent that sample is relative to the others. Example pairs for the MLP result from laying its TDL like a moveable stencil within sequence blocks. Given the MLP has a 30-sample TDL, extending 20 samples back in time, the earliest input sample in the block from which a prediction can be made is at Index 20. Given the one sample mapped horizon, the latest accessible input index in a block of 40 examples is 39. One sequence block thus offers 19 unique example pairs. An epoch of training is one pass through a number of randomly drawn example pairs equaling the number of unique example pairs. Here, one epoch of 400 sequence blocks (300 for training and 100 for validating) visits 5700 training pairs and 1900 validation pairs. All pairs are drawn randomly with replacement so diversity between epochs is encouraged.

Early trials showed it is vitally important that training examples be presented to the MLP randomly, scrambled thoroughly away from their historically factual order. The goal here is not that the MLP work as a coil state estimator (i.e., a time-series model, which does require the historically correct sequence), but as a coil process estimator, able to predict coil dynamics across state space more broadly than can be inferred from any single time series alone. Training conducted while on-line with real-time data should be randomized as well.

A very slow learning rate (e.g., 0.0001 to 0.0002) is invoked over the first dozen epochs. The rate is then raised in steps over later epochs, usually ending between 0.005 and 0.010. For early epochs, only one example pair is presented from each sequence block that is drawn. Later, this number steps to two and then three to speed training progress along. Since a gradient descent optimization is used, the object is to avoid dropping too far down the error surface until all data regions have been well visited. This way, the trained MLP better generalizes all training set examples.

Satisfactory training is generally evidenced by a steep fall in RMS validation error over the first 50 to 100 epochs. This is generally followed by diminishing improvement (a gradual leveling of error decrease) to approach an obviously terminal value within the succeeding 100 to 200 epochs. In the work here, terminal error typically was less than  $\pm 0.25^\circ\text{C}$  ( $0.45^\circ\text{F}$ ). Learning benefitted in some cases from bumpiness in middle epochs of the leveling trend, evidence of idiosyncratic association building (Mitchell 1997).

## MLP Configuration and Training Mixes Theory, Intuition, and Heuristics

Veronica (2002) showed that high terminal values of RMS validation error can result if a MLP has too few free parameters for the process complexity being modeled, too few free

parameters for the range of generalization expected, too many free parameters for the number of unique training examples available, or predicts a time horizon too short for slow dynamics of the process to evolve significantly. Presently, there is no mathematical formalism by which those issues, the proper number of parameters (e.g., hidden nodes), or the proper content for the TDL can be addressed fully by theory. The MLP configuration seen in Figure 3 resulted empirically as acceptable performance on training validations and recursive predictions became regularly achieved while modeling a real cooling coil for adaptive control in Veronica (2002).

The empirical approach is aided somewhat by theory. For example, given control valve position is the only input variable that is cyclical during a query, it is necessary that it be tapped within the TDL by a sufficient number of values to prevent aliasing the query (a triangle pulse). Thus, 20 successive values (depicted by  $x_4$  in Figure 3) map it over the entire TDL depth, while the four other input quantities ( $x_1$  through  $x_3$ , etc.), which lie static, are defined sufficiently by just two taps each, one at each end of the TDL. Conversely, the number of hidden units comes more intuitively. For a given amount of data, the best number of parameters is the fewest giving good results. It seemed intuitive that the MLP mapping have the capability to express a point of inflection without saturating its output. A simple network able to do that is a single hidden layer of three nodes. The resulting performance met the objectives presently sought, so the MLP remained in that configuration for this work. The details of the training routine described earlier were arrived at by a similar mix of trials and intuition.

The MLP coil model need not be trained comprehensively across the full input space to perform its fouling detection task. Queries are specified at preferred points of input space. The MLP needs to model the coil very well only near the input regions queried. A training quality ranking can be assigned to the MLP based upon its experience and performance, with example pairs in the regions of input state space preferred for queries. As discussed in the companion paper (Veronica 2010), the supervising agent might manage multiple MLP instead of just one, each being a specialist in a particular region of input space to localize its generalization burden. Each specialist MLP would be fed data only from its assigned region. The trials simulated here not only consider MLP trained on synthetic, sampled data having global (whole input space) content, but also MLP trained on focused data, limited to regions of input space preferred for fouling detection queries.

## RESULTS OF QUERIES RUN ON A MLP COIL MODEL

Visualizing the action in a MLP being employed dynamically and recursively is a difficult challenge. This approach uses separate panes of information in a composition comparing what happens during two successive queries of a MLP, one run at baseline and one run later after fouling develops.

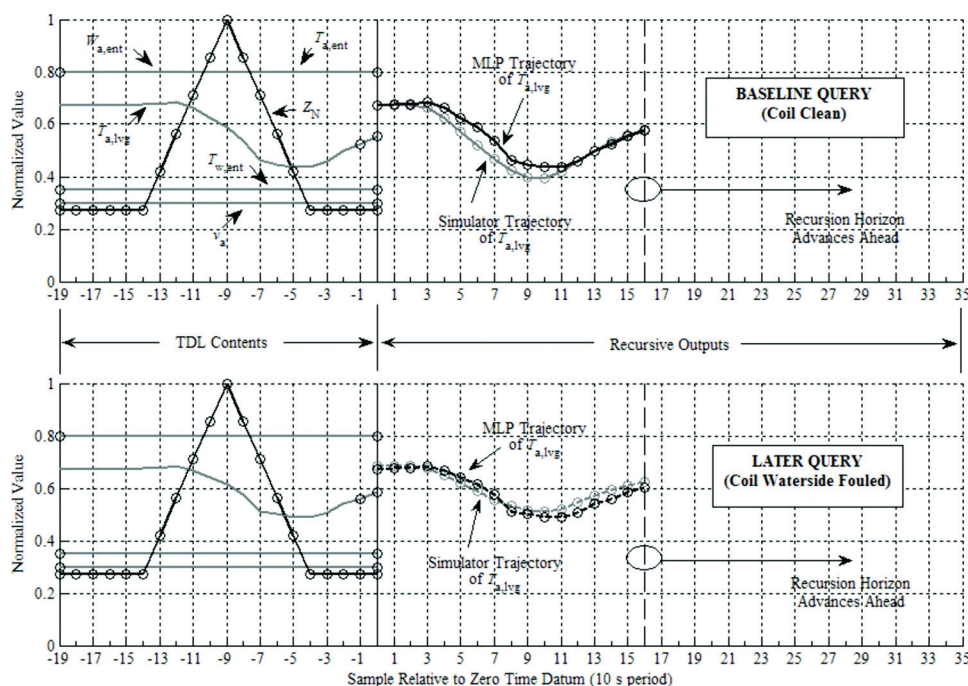
### Viewing a MLP in Action

Viewing the response of a MLP to a query is aided greatly by plotting simultaneously: (1) the current contents of the MLP tapped delay line; (2) the current recursively predicted MLP output (leaving air temperature,  $T_{a,lvg}$ ); and (3) the history (i.e., trajectory) of such predictions back to the query start. Furthermore, coincidentally plotting a parallel trajectory from the numerical simulator readily compares the data-driven response of the MLP to the first-principles that trained it. Over a query involving, for example, a 350 s summation length and a 10 s data acquisition period, there are 35 frames. Each panel (top and bottom) of Figure 6 is one of those frames, captured at the middle of a query sequence. In the top panel, the MLP queried is that trained on clean coil training examples. The bottom panel MLP is that trained on the coincidental water-side fouled examples. Exercising the detection concept involves summing up, value-by-value, the differences between the fouled (i.e., any later query) and clean (the archived

baseline query) MLP trajectory. The summation value expected is that obtained by first-principles simulation of the same query, tabulated as surfaces such as Figure 2.

A vertical black bar at Period 0 splits the panel into two windows, both showing simultaneous information. The 30 values comprising the MLP input vector  $X$  in Figure 3 are circled in the window left of the zero split, appearing as they lay out in the TDL to generate the current prediction. The window right of the zero split shows the trajectory of predictions of coil output (leaving air temperature) made by the MLP, leading up to the current prediction. That is, the TDL content seen in the left window is current for Period 15, which generated the output predicted for Period 16 (marked by the advancing vertical dashed bar). TDL samples are labeled with negative indices. Samples in the TDL grow older by moving from right to left across the left (TDL) window as recursions iterate to the right across the right (output) window.

Five instrumented coil process input quantities are utilized by the TDL as inputs to the MLP: temperature of entering air ( $T_{a,ent}$ ), humidity of entering air ( $W_{a,ent}$ ), velocity of air at coil face ( $v_a$ ), temperature of entering water ( $T_{w,ent}$ ), and chilled-water control valve position. Coil water flow instrumentation is not often seen at installations in the field, but a physical position ( $Z_V$ ) of the control valve typically can be assumed from the last positioning command sent to the actuator. A zero  $Z_V$  value corresponds by human intuition to the valve being shut. A zero input is intolerable to the MLP, however, as this implies to it not a defined zero state, but no defined state at all. The  $Z_V$  value is thus mapped to a compressed scale ( $Z_N$ ) of valve position, ranging from 0.1 (valve shut) to 1.0 (wide open). Current and past values of the coil process output (temperature of leaving air,  $T_{a,lvg}$ )



**Figure 6. Views of MLP input and output during a baseline query (top) and a later query (bottom).**

are also recurred to the TDL as inputs to the MLP. A total of six physical quantities are thus found in the TDL as MLP inputs. Only those samples of the six quantities having a specified age relative to the current time step of the query sequence are put in the TDL, shown as circles in the TDL contents buffer in Figure 6. Note that all 30 samples are used for each prediction (two each for  $T_{a,ent}$ ,  $W_{a,ent}$ ,  $v_a$ ,  $T_{w,ent}$ , and  $T_{a,hvg}$ , and 20 samples of  $Z_N$ ).

Viewing MLP performance on a desktop computer is done by executing a runtime script written using a runtime interpreted mathematics software package. The script steps the trained MLP through all frames of a query sequence (35 in this case). The fixed context of the input at  $T_{a,ent}$  32°C (89.6°F),  $T_{w,ent}$  = 7°C (44.6°F),  $W_{a,ent}$  = 80% relative humidity, and  $v_a$  = 0.63 m/s (124 fpm) locates the query at the point labeled *C* on the Figure 2 surface where the largest summation value of 274°C · s (493°F · s) is expected. The triangular pulse in valve position first enters the TDL on the right, then steps through the TDL to eventually fall out of it on the left. Figure 6 captures one frame near the middle of a query. The TDL content seen is that buffered at Step 15 (i.e., 15 steps into the query), and the MLP in each case has predicted  $T_{a,hvg}$  for Step 16. The trajectories of Figures 7 through 10 are similarly split-frame views but show the end of the 35 step query; the water flow pulse having come and gone from the 20-step deep TDL, so that only the flat line of post-pulse  $Z_N$  values are seen in the TDL. Such a frame captured from a dynamic desktop simulation, though a static view, still usefully expresses the performance of a particularly configured MLP in answering a query.

## Results from a MLP Trained Only Near the Point Queried

A set of queries is first simulated at a point of input space shown by Figure 2 to respond well to water-side fouling. The MLP models used for this set were trained on 15,200 example pairs having a 10 s period (about 42 h of richly dynamic data) focused near this very responsive point of input state space. Figures 7 and 8 are similar to Figure 6, but show the situation at the end of each query, which occurs here at Period 35 (i.e., a 350 s query). The triangle pulse in control valve position  $Z_N$  (and hence, water flow) has come and gone, and is now flat at the  $Z_N$  value corresponding to the valve being 20% open. The MLP producing Figure 7 involved the slow-responding, shielded leaving air temperature sensor. The traces of clean coil and fouled coil values of  $T_{a,hvg}$  predicted each dip by MLP and then recover to near initial value, generally agreeing with the reference trace provided by the first-principles simulation. As expected, the dip in the MLP trace is more shallow for the fouled coil case. The 219°C·s sum from the MLP is a significant 24% below the 288°C·s (518°F·s) sum expected per the simulation, but is reasonably good considering its early development toward this task. The simulation produces a slightly larger sum (288°C·s [518°F·s]) for the query than it did generating the Figure 2 surface (274°C·s [493°F·s]), because comparison to the MLP sum requires the 1 s time steps of the simulation to be integrated at the coarser 10 s sample period the MLP uses. Further development can address this discrepancy, as well as other issues. For example, the trace of MLP predictions rolls into the dip more slowly than the simulation reference but recovers at a similar rate, an asymmetry warranting further study. Encouragingly, the MLP predictions remained stable through the 35 recursions required and apparently would continue to be stable even further.

Figure 8 shows the same input context and queries as Figure 7, except that the fouled MLP is trained on data generated by simulations of a coil having air-side, rather than water-side, fouling of nominal severity. The 62°C·s (112°F·s) sum obtained from the MLP is within 10% of the expected sum obtained from the simulation. The MLP replicates with reasonable accuracy a crucial feature expected from first-principles simulations. At the point of input state space specified in this query, a 418% larger MLP sum should result from nominal fouling on the water side than results from nominal fouling on the air side. The MLP tested here delivers a 353% distinction, an encouraging outcome for this early stage of development.

In Figures 7 and 8, the query is at a point of input state space having very high entering air temperature and humidity, which are most favorable to a large query sum, given water-side fouling, but not typical of the operation of most air handler units. A more realistic query context would be between 50% relative humidity and 65% relative humidity, with  $T_{a,ent}$  at 25°C (77°F). This situation is considered in Figure 9.

Results from a MLP Trained Globally

Besides the more typical query context, the MLPs in Figure 9 are more broadly trained, using example pairs totaling to 84 h of data, reflecting both dynamic and steady states drawn throughout the input state space (instead of coming only from the region near the query). Another change is that the data used to train the MLP models of Figure 9 reflects a fast-responding leaving air temperature sensor. The 29°C·s (52°F·s) sum obtained between the clean (top panel) and water-side fouled (bottom panel) traces of MLP predictions in Figure 9 is only 20% of the 146°C·s (263°F·s) sum expected. At several points, the MLP predictions markedly overshoot the trace expected. An encouraging result is that the overshoots exhibit under-damped stability. However, in some of the cases, where the same MLP configuration was trained on other sets of global data, the predictions are unstable, or become recursively stuck in upward or downward trends.

Figure 10 shows predictions traced by another clean/fouled pairing of MLP models. These have the same configuration and are trained on training set example pairs having the same input

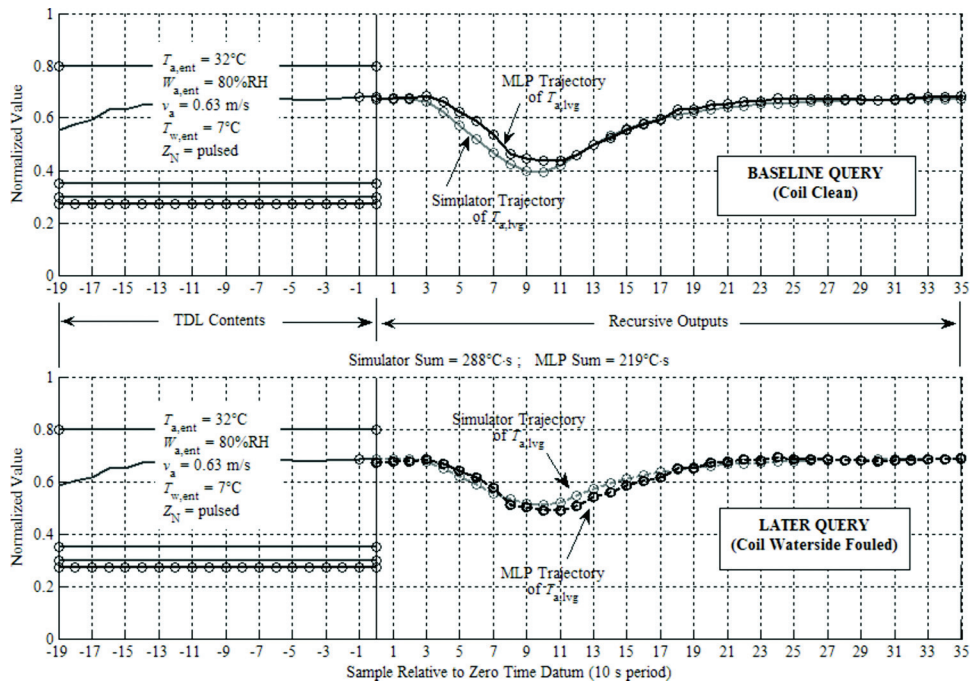
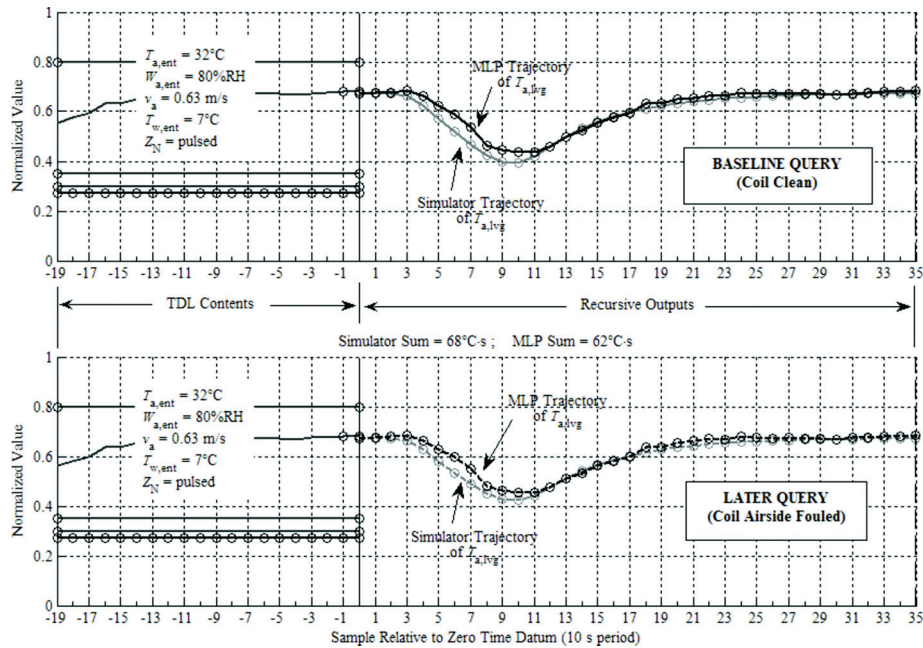
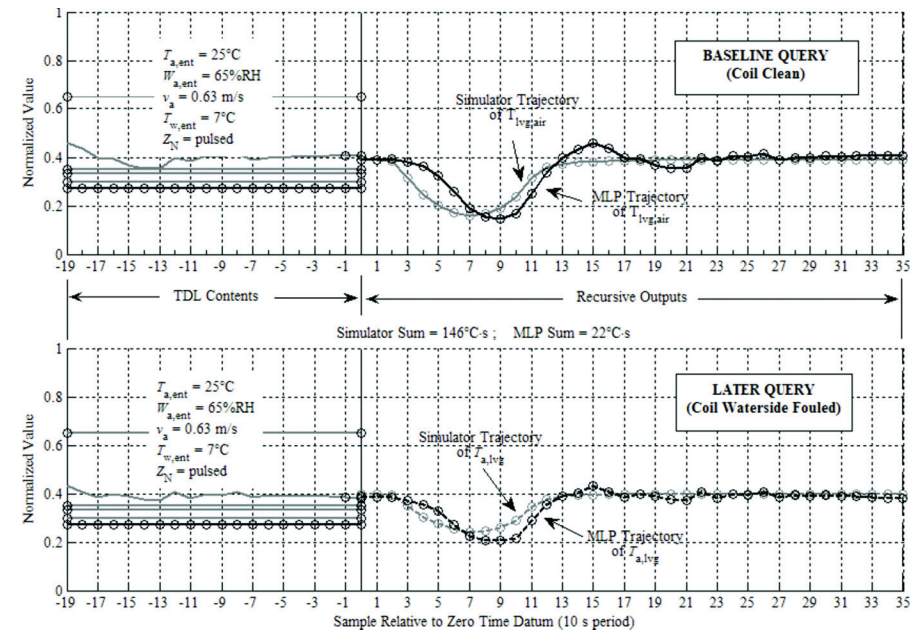


Figure 7. MLP predictions through a baseline query (top) and a later query (bottom) compared with values from numerical simulation (fouling on water side).



**Figure 8. MLP predictions through a baseline query (top) and a later query (bottom) compared with values from numerical simulation (fouling on air side).**



**Figure 9. Predictions from a MLP trained globally on data from a fast-acting sensor.**

values as the MLP pair exercised in Figure 9. One difference in Figure 10 is the output training examples reflect a slow-responding temperature sensor. Also, the triangle pulse of control valve position is driven at one-half the speed of that in Figures 6 through 9, so that at the end of 35 steps, some of the pulse remains in the TDL buffer. The predictions track the expected values much more closely than in Figure 9, although the MLP configuration and the input space visited during training are the same.

It may be that in Figure 9 the greater demands of a faster-acting sensor and a much broader input state space being generalized combined to overtax the modeling capacity of the simple, three-node MLP structure of Figure 3. This appears to be the case, particularly in light of the better results Figures 7 and 8 give for a slow-sensor and narrow-generalization span. The modeling demands appear relaxed somewhat in Figure 10. There, the slow sensor is again in use, the excitation pulse is slower, and the results improve.

## CONCLUSION

The relatively simple, three-node, 97-parameter MLP of Figure 3 had little development specifically targeting its tasks. Yet, it predicted coil output recursively throughout queries, with sufficient accuracy and stability to warrant further development of MLP for the role. Figures 7 and 8 show the sums resulting from queries of MLP modeling a water-side fouled and an air-side fouled coil were within 24% and 10%, respectively, of the sums expected by first-principles analysis. Training of the MLP to achieve these results consisted of 42 h of 10 s simulated samples focused in the region of input state space near the point queried. Figure 9 shows that performance suffers markedly as a greater modeling burden is imposed upon the same MLP structure,

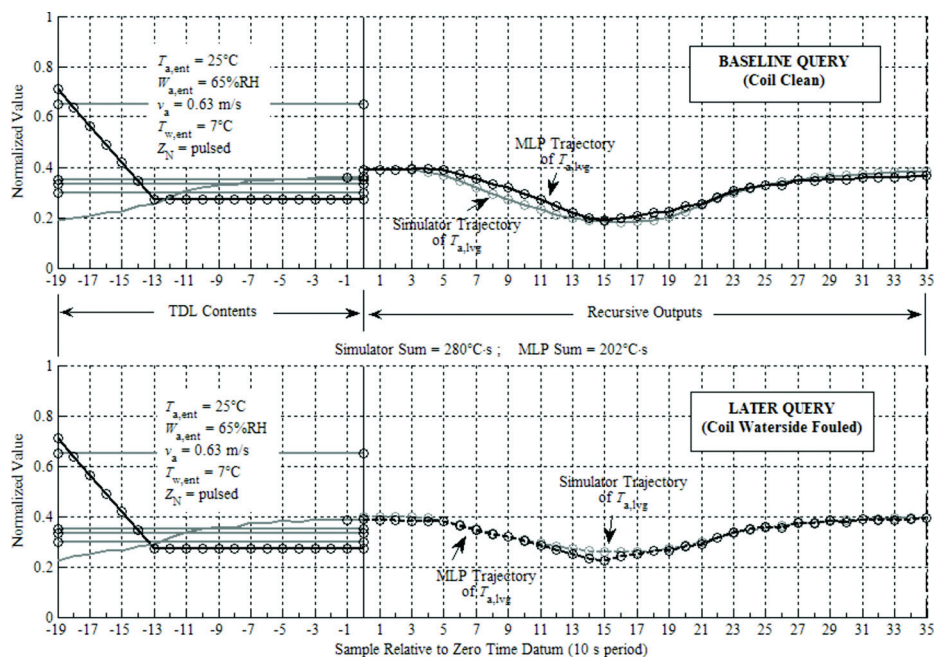


Figure 10. Predictions from a MLP trained globally on data from a slow-acting sensor.

with the input space to be generalized being broader and the temperature sensor being faster. Returning to a slow sensor, as well as a slower water flow transient, relaxed some of the modeling burden, as seen in Figure 10.

Stable returns to ground state were observed after the triangle pulse passed through the TDL even though, in some cases, under-damped oscillations intervened. This is encouraging, because unlike numerical methods based upon difference equations, a MLP comes under no analytical formalism ensuring its iterative stability. It cannot be concluded at this stage why stable behavior was observed in some cases and not in others. One possible explanation is that stability occurs when the prevailing time constant of the combined coil and sensor process the MLP has learned, which varies with the operating regime of its data, is sufficiently long compared to the MLP sample period. It is a topic for further study.

Further development of the concept introduced by the companion paper (Veronica 2010) must resolve other issues not addressed here. A principal issue is learning. Through on-line learning, the MLP must reflect the latest coil condition given two or more regions of input space known (from the embedded table) to yield significantly different query sums, with those regions all represented at comparable accuracy and stability. If operation of the AHU does not visit the relevant input regions regularly, this requirement may present a significant challenge. In a simple MLP, accuracy in a region visited more recently may come at the expense of accuracy in regions learned earlier. Specialization of multiple MLP, scored by the agent as mentioned in the companion paper, may help. The extent to which training data must have dynamic content also warrants study. Nonlinearity of water flow to valve position is less of a difficulty. The MLP used here satisfactorily mapped the highly nonlinear characteristic of a real, installed valve (Veronica 2002). A question requiring future investigation is the extent to which that installed nonlinearity might affect the sums obtained from queries.

This concept to automatically detect cooling coil fouling requires more development by computer simulation before the expense of hardware trials can be justified. But, the encouraging performance here of the MLP warrants that further effort.

## ACKNOWLEDGMENT

I thank Karl and Madeleine Larson, principal benefactors of the Larson Building Systems Laboratory at the University of Colorado-Boulder (where this research was performed) for the generous support that made this work possible.

## NOMENCLATURE

$b$	=	MLP node bias	$v$	=	velocity (at coil face)
$P$	=	time period	$n$	=	index, discrete sample
$f$	=	frequency	$W$	=	humidity
$Q$	=	flow rate, volumetric	$M$	=	MLP weight matrix
$F$	=	function or factor	$X$	=	MLP input vector
$t$	=	time	$T$	=	temperature
$G$	=	function or factor	$Z$	=	control valve position

## Subscripts

$a$	=	air, or at air-side interface	$S$	=	sampling
$N$	=	Nyquist (in $f_N$ )	$lv_g$	=	leaving (the coil)
	=	compressed (neural) scale (in $Z_N$ )	$V$	=	valve (control)
$ent$	=	entering (the coil)	$w$	=	water, or at water-side interface

## Greek Symbols

$\Delta$	=	difference (finite)	$\tau$	=	time constant
----------	---	---------------------	--------	---	---------------

## REFERENCES

- Bishop, C.M. 2006. *Pattern Recognition and Machine Learning*. New York: Springer Science+Business Media.
- Díaz, G., M. Sen, K. Yang, and R. McClain. 1999. Simulation of heat exchanger performance by artificial neural networks. *HVAC&R Research* 5(3):195–208.
- Hyvarinen, J. 1996. *Real Time Simulation of HVAC Systems for Building Optimization, Fault Detection and Diagnosis—Volume 2, Technical Papers of IEA Annex 25*. Espoo, Finland: Technical Research Centre of Finland, VTT Building Technology.
- Lee, W.Y., C. Park, and G. Kelly. 1996a. Fault diagnosis in an air-handling unit using residual and recursive parameter estimation methods. *ASHRAE Transactions* 102(1): 528–39.
- Lee, W.Y., C. Park, J. House, and G. Kelly. 1996b. Fault diagnosis of an air-handling unit using artificial neural networks. *ASHRAE Transactions* 102(1):540–49.
- Li, X., H. Vaezi-Nejad, and J. Visier. 1996. Development of a fault diagnosis method for heating systems using neural networks. *ASHRAE Transactions* 102(1): 607–14.
- Mitchell, T.M. 1997. *Machine Learning*. Boston: WCB McGraw-Hill.
- Peitsman, H.C., and V. Bakker. 1996. Application of black-box models to HVAC systems for fault detection. *ASHRAE Transactions* 102(1):628–40.
- Veronica, D.A. 2002. Adaptive, predictive control of a cooling coil by an artificial neural network using emulator differencing. Master's thesis, Department of Civil, Environmental, and Architectural Engineering, University of Colorado at Boulder, Boulder, CO.
- Veronica, D.A. 2008. Detecting heat exchanger fouling automatically with an embedded data-driven agent using expert signature maps. PhD dissertation, Department of Civil, Environmental, and Architectural Engineering, University of Colorado at Boulder, Boulder, CO.
- Veronica, D.A. 2010. Detecting cooling coil fouling automatically—Part 1: A novel concept. *HVAC&R Research* 16(4):413–34.
- Zhou, X., and J. Braun. 2007. A simplified dynamic model for chilled-water cooling and dehumidifying coils—part 1: development (RP-1194). *HVAC&R Research* 13(5):785–804.

