



Artificial Neural Network control of thermoelectrically-cooled microfluidics using computer vision based on IR thermography

Benjamin A. Rizkin, Karina Popovich, Ryan L. Hartman*

Department of Chemical and Biomolecular Engineering, New York University, 6 MetroTech Center, Brooklyn, NY 11201, United States

ARTICLE INFO

Article history:

Received 7 August 2018

Revised 10 October 2018

Accepted 19 November 2018

Available online 4 December 2018

Keywords:

Thermal control
Machine learning
Neural networks
Process control
Autonomous microfluidics
Process resiliency

ABSTRACT

High-speed and high-accuracy thermal control of reactors has always been of interest to chemical engineers. In this paper we present a new methodology for thermal control of a continuous-flow chemical reactor using non-contact IR thermography combined with computer vision and a predictive Artificial Neural Network. The system exhibits several key advantages over thermocouples and PID control including the ability to quantify and account for thermal diffusion in the system, to collect and process data very quickly and with high accuracy, to analyze the entire surface of the reactor, and to update its training based not only on the current thermal response, but also on external factors. We have constructed and validated such a system as well as shown improvements in its accuracy, rise time, settling time, set point tracking, and overshoot as compared to more traditional forms of thermal control, validating this as a possible approach for experimental and process control.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

1.1. Contextual overview

Artificial Neural Networks (ANNs) are known for their ability to accurately represent data in highly non-linear and multivariate systems. In fact, ANNs have already been successfully used to model chemical reactions (Ahmadi et al., 2009; Maltarollo et al., 2013), control small modular reactors (Manic and Sabharwal, 2011), optimize chemical systems (Polikar et al., 2001; Zhang et al., 2010), predict stock markets (Zavadskaya, 2017) and even identify malignancies in histological images (Xue and Ray, 2017). It has been demonstrated in the literature that ANNs offer certain

benefits over traditional numerical approaches, both in accuracy, speed of execution and resiliency to sudden changes in input parameters (Ahmad et al., 2017; Parkale, 2012). Additionally, and perhaps most exciting, is that ANNs can self-train through algorithms. This means that in a day and age of distributed modular systems, exploiting lab-on-a-chip (LoC) technologies or advanced materials (such as in photochemical reactors), system control can be greatly simplified, and tuning can be merged with quality assurance in a production environment. Finally, ANNs can process significantly more data in a reasonable amount of time with modest computing resources. This can be leveraged either to control based on multivariate trends or to increase the quality of the input data. ANNs can easily be combined with unorthodox sensors that produce very large amounts of data through capture at high resolution, high speed or a combination of both. Overall ANNs offer many exciting benefits for process control in a variety of laboratory systems that can also aid in the design of production-scale processes.

ANNs have been used in the past for process control quite successfully using various different approaches (Dayal et al., 1994; Hagan et al., 2002; Khalid and Omatu, 1992; Psaltis et al., 1988; Vamvoudakis et al., 2015; Zribi et al., 2015) and the NARMA-L2 neurocontroller algorithm has gained prevalence in the control of highly-nonlinear systems in the last two decades. The issue with many of these previous control schemes has been the difficulty of selecting an appropriate model and translating the system behavior into a data form understandable by the computer. For example,

Abbreviations: ANN, Artificial Neural network; CAD, Computer Aided Design; CAM, Computer Aided Modeling; CNC, Computer Numerical Control; CGB, Conjugate Gradient Backpropagation; CGBPPFRU, Conjugate gradient backpropagation with Fletcher-Reeves updates; CGBPPBR, Conjugate gradient backpropagation with Powell-Beale restarts; CGBPPRU, Conjugate gradient backpropagation with Polak-Ribière updates; DLL, Direct Link Library, compiled C/C++ code; FPS, Frames per Second; IoT, Internet of Things; LM, Levenberg-Marquardt (Damped Least Squares) with forward training; NARMA, Nonlinear Autoregressive Moving Average; OSSBP, One-step Secant backpropagation; PID, Proportional Integral Derivative control; PWM, Pulse Width Modulation; q-NBP, quasi-Newton backpropagation; RBP, Scaled Conjugate Gradient Backpropagation; UART, Universal Asynchronous Receiver-Transmitter; UFPA, Uncooled Focal Plane Array; USB, Universal Serial Bus; VOx, Vanadium Oxide thin film microbolometer.

* Corresponding author.

E-mail address: ryan.hartman@nyu.edu (R.L. Hartman).

the NARMA-L2 control algorithm works by canceling nonlinearities to transform nonlinear dynamics into linear ones and PID-neural net systems require rigorous models of the system, just like first-principles PID tuning. While these existing algorithms have worked well where they have been applied, the fact remains that they need a rigorous model of the system, significant computational power, and expertise to implement. In this work we present a different approach to ANN-based control using computer vision and fully autonomous training for implementation in a thermal control system for a microfluidic reactor.

The use of IR thermography in research is a popular method for monitoring reactions and receiving data from these reactions. Over the years, advanced IR thermography methods have been developed to maximize the efficiency of scientists and their research. IR thermography has demonstrated the ability to provide useful data beyond temperature control, such as enthalpy and kinetics measurements (Romano et al., 2015; Zhang et al., 2016). There have also been studies which have used IR thermography to monitor reactions and indicate the location of their intensity across a two-dimensional system (Fu et al., 2016). These methods have proven IR thermography to be very successful in providing data and monitoring reactions in microreactors and microchannels.

IR thermography has attracted much attention among scientists allowing for further advancements of its applicability and methods to be made. Over the years many new methods and techniques have been developed. Infrared technology led to the development of precision thermometry to track temperature profiles of fast and highly exothermic reactions (Haber et al., 2013). Recently, a non-intrusive method to measure fluid temperature and two-phase flow patterns was developed (Liu and Pan, 2016). IR thermography has also been coupled with temperature frame processing methods to estimate heat distribution of chemical reactions along a channel in a microreactor (Pradere et al., 2006). The development of such measuring and monitoring techniques demonstrate the ability to collect a lot of spatially-dependent information in a short amount of time, hinting at the broad-scale acceptance of IR thermographic studies for micro-scale systems, and validates the benefits of using IR thermography in this study.

Similarly, Artificial Neural Networks have proven to be beneficial for process control in scientific studies and have become frequently used in basic chemistry research. ANN's are highly effective at monitoring process systems with a lot of variables that altogether are too complex to manage without an ANN (Uraikul et al., 2007). Artificial intelligence has been widely applied to fields of study ranging from aquaculture to pollution minimization (Chan and Huang, 2003; Lee, 2000). ANNs have also had much success in representing phenomena in unit operations that are difficult to model. For example, recent research has focused on studying heat exchangers and modeling power consumption (Dudzik, 2011). Although ANN's have become a staple in scientific research the development of new methods involving ANN's have slowed down. Some methods that are currently used have been developed over a decade ago with very few studies involving modern advancements in ANN process control, despite the potential enabled by miniaturization of silicon technologies leading to much smaller, power-efficient and inexpensive processing systems. Despite the lack of many recent publications of ANN-based control, it is still a topic of great interest both to chemical engineers, and those outside the field.

Overall, having resilient process controllers which can adapt to a variety of circumstances and operating parameters is of great interest to industry. In the past a lot of work has been devoted to the formulation of robust multivariate controllers (Ingham et al., 2015; Kourti and MacGregor, 1995; Kresta et al., 1991) and to the understanding of complex system dynamics. By enabling

highly multivariate data input into the ANN algorithm, it is expected that multivariate control, especially in complex industrial environments, can be enabled and simplified, making it competitive with traditional univariate controls. Additionally, due to the increased simplicity of writing and optimizing neural networks, along with new developments in specialized neural network computing platforms, networks that collect, analyze, and act upon data can be distributed throughout a plant on small, low-power computers.

1.2. Artificial Neural Network overview

An ANN is a synthetic representation of the biochemistry seen in nature, whereby the summation of weighted inputs, if accumulated to a threshold value, leads to a firing of the neuron. In turn this first neuron triggers other downstream neurons with each connection having a given weight. Represented mathematically the activation function of a neuron j at time $t + 1$ would be:

$$a_j(t + 1) = f(a_j(t), p_j(t), \theta_j) \quad (1)$$

where a_j symbolizes the activation of a neuron, θ_j represents the threshold/bias value determined during net training, and p_j is the input of the network. The signal then propagates through the network with the function:

$$p_j(t) = \sum_i f_{out}(a_i(t))\omega_{ij} \quad (2)$$

where ω_{ij} is the weight of a connection. Due to the simplicity of these arithmetic operations a shallow neural network can run very quickly, while a deep neural network can process unfathomable amounts of input data. ANNs like these are also known as multilayer perceptrons.

In practice several different types of threshold functions for neurons are used in ANNs, including step functions, linear functions, sigmoid functions, hyperbolic tangents, and Rectified Linear Units (ReLU). These various forms of neurons are employed in different parts of the network including the hidden and output layers. Often, a network would be composed of a variety of different neurons linked to each other. A basic representation of a network with three hidden layers can be seen in Fig. 1.

1.3. Artificial Neural Network training

Before a network can be used, it needs to be trained. This is accomplished via a variety of different possible training algorithms for which the general goal is to adjust the weight ω and bias values θ of the various neurons until the desired output is achieved. Adjustments to the weights change the contribution of each input to a neuron, while adjusting bias values will shift the threshold function/ adjust the steepness of a sigmoid. In practice, the training is carried out much like any standard optimization problem would be solved in engineering- a cost function is defined, and then parameters are adjusted until the 'optimal' solution is found within certain performance and accuracy targets. The cost function defines how far the current solution is from the optimal one, implying that the most optimal solution has the lowest cost. Although ANNs can usually achieve arbitrarily good fit, certain parameters are used to determine when to end training as to enable computation in a reasonable amount of time. There are numerous training algorithms available for network training, generally divided into conjugate gradient and quasi-Newtonian methods. It is important to remember that these algorithms require cleanly differentiable weight, input, and transfer functions to compute successfully.

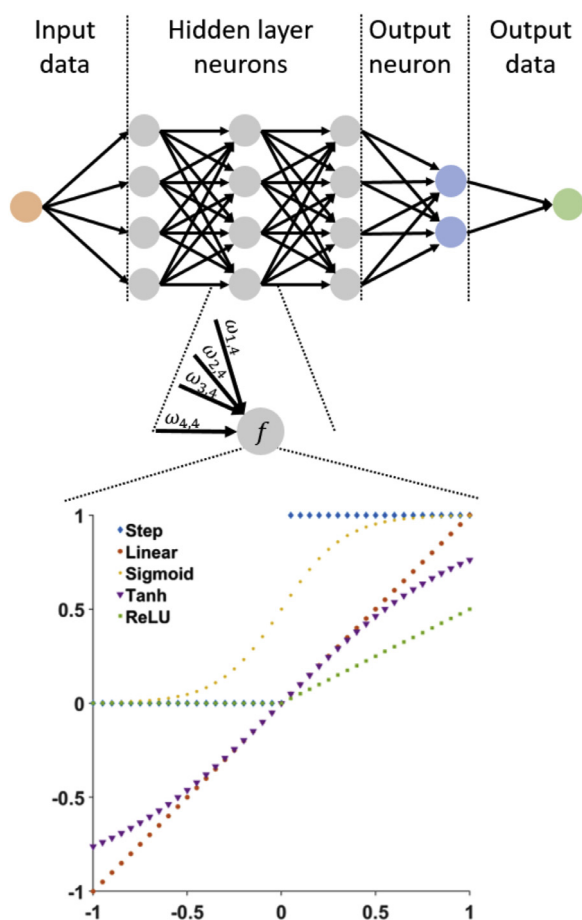


Fig. 1. General structure of an Artificial Neural Net. Top: macro overview of input to output data flow, middle: inputs to individual neuron, bottom: different possible threshold functions for a neuron. Usually a normalized sigmoid is used as it gives the greatest flexibility to the training algorithm and thus best represents the data.

1.4. Training algorithms

The various training algorithms investigated herein include:

1. Levenberg–Marquardt (Damped Least Squares) with forward training: first, the performance of a network is assessed with respect to the bias and weight variables. Backpropagation is used to calculate the Jacobian (Li et al., 2016), from where each variable is then adjusted to the Levenberg–Marquardt rules (Levenberg, 1944). Training occurs until either maximum computation heuristics are exceeded (time, number of epochs, etc.), training parameters are not met (performance gradient, μ or validation failure), or the network reaches its performance goal.
2. quasi-Newton backpropagation: as engineers, we are all familiar with Newton's method, which relies upon computation of a Hessian matrix. However, the Hessian is very computationally intensive to compute as it involves the second derivatives. In quasi-Newtonian methods the Hessian is replaced with an approximation based on the gradient. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm (summarized in Dennis and Schnabel, 1985) uses this approximation to compute updated weights and biases for the network. Training is continued until performance goals are reached or the number of epochs, or maximum time are exceeded.
3. Resilient backpropagation (RProp): essentially, this is a basic first-order optimization algorithm, the weights in a network are updated based on the sign of the partial derivative of the error function (Riedmiller and Braun, 1993). If the sign of the error

is equivalent between two consecutive iterations then the update factor is positive, if it changes sign then the update factor is negative. Training is continued until either performance goals are met, or computational heuristics are exceeded.

4. Scaled conjugate gradient backpropagation: first backpropagation is used to calculate the derivatives of performance, then a scaled conjugate gradient algorithm is used (Möller, 1993). The conjugate gradient algorithm is based on conjugate directions, but a line search is not performed during each iteration of training. Training is continued until computation heuristics (number of epochs, minimum gradient or maximal failure rate) are exceeded or the performance goal is met.
5. Conjugate gradient backpropagation with Fletcher–Reeves updates: again, this algorithm is similar to the conjugate gradient method, but it computes the search direction by dividing the norm square of the previous and current iterations (Fletcher, 1964). Training is continued until computation heuristics (number of epochs, minimum gradient or maximal failure rate) are exceeded or the performance goal is met.
6. Conjugate gradient backpropagation with Powell–Beale restarts: this algorithm is very similar to the conjugate gradient method, except it uses a search algorithm at each iteration (Powell, 1977). This search algorithm computes a search direction from the gradient and previous search direction. The search direction is reset based on a numerical test. Training is continued until computational heuristics (number of epochs, minimum gradient or maximal failure rate) are exceeded or the performance goal is met.
7. Conjugate gradient backpropagation with Polak–Ribière updates: again, this algorithm is similar to the conjugate gradient method, but it computes the search direction by using a formula involving the norm square combined with the gradient (Khoda et al., 1992). Training is continued until computation heuristics (number of epochs, minimum gradient or maximal failure rate) are exceeded or the performance goal is met.
8. One-step secant backpropagation: first, backpropagation is used to calculate the derivatives of the performance vector with respect to weights and biases. Next, the variables are adjusted according to a search algorithm where the direction is calculated as a function of the gradient, step changes in the weights from the previous iteration and the gradient change from the previous iteration (Battiti, 1992). Training is continued until computation heuristics (number of epochs, minimum gradient or maximal failure rate) are exceeded or the performance goal is met.
9. Gradient descent with momentum and adaptive learning rate backpropagation: as in the other methods, first backpropagation is used to compute the necessary derivatives. Next, the variables are adjusted based on the gradient descent with momentum (Moreira and Fiesler, 1995). After each iteration if the performance of the network is closer to the goal then the learning rate is increased, if it is further away the learning rate is decreased and the change is not kept. Training is continued until computation heuristics (number of epochs, minimum gradient or maximal failure rate) are exceeded or the performance goal is met.

A final but important factor to consider about ANNs is their ability to self-train during operation. This is accomplished by sequentially updating the weight and bias values of the various neurons during use of the network. This functionality is important for longer-term installations where variables such as wear and tear, weather fluctuations, input power quality or other transient factors can introduce a drift in the system. It is also useful for “teaching” the system how to respond to process changes such as flowrate or composition. By using each run to keep the training accurate, the set point accuracy of the controller algorithm can be kept more constant over time.

Table 1

Comparison of the accuracy and data speed of IR cameras with traditional forms of thermal measurement.

Measurement type	Accuracy (standard)	Accuracy (specialized)	Data speed (standard)	Data speed (specialized)	Number of measurement points
IR Camera	$\pm 0.2^\circ\text{C}$	$< \pm 0.1^\circ\text{C}$	33 ms/frame	5 ms/frame	300k to millions per frame
Thermocouples (T, J, E, K, N, R, S, B, C [™])	$\pm 1\text{--}2.2^\circ\text{C}$	$\pm 0.25\text{--}1^\circ\text{C}$	1000 ms/ measurement	50 ms/ measurement (aerospace grade)	1
Thermistors	$\pm 5^\circ\text{C}$	Very high over narrow range	–	–	1

1.5. Introduction to IR imaging

Infrared is as much part of the electromagnetic spectrum as is visible light, but unlike visible light its intensity is related to the temperature of an object through the concept of a black body emitter, tying it directly to the object's temperature. Due to advances in Vanadium Oxide and Uncooled Focal Plane Arrays (Li et al., 2011), thermal imaging cameras have transitioned from being bulky contraptions requiring cryogenic cooling units to handheld webcams powered from a USB port. This has resulted not only in greater simplicity, but also in drastic price decrease, making them competitive with high accuracy thermocouples. This offers several distinct advantages to the chemical engineer, primarily ease of application, acquisition speed, accuracy, and resolution. Traditional thermocouples need to be placed all around a reactor, resulting in increased complexity, especially where long shielded wiring runs are required due to the analog nature of the signal. Thermal cameras can cover a large area of a reactor, or even an entire process, providing multiple measurements with minimal hardware. Also, IR cameras are capable of non-contact measurement, giving a distinct benefit when dealing with either extreme temperatures or sensitive environments such as bioreactors. This is also particularly useful for *in-situ* measurements as an entire reactor can be constructed of IR-transparent material. Finally, thermal cameras can capture data very quickly; standard models can capture 30 FPS and specialty units are capable of greater than 3000 FPS. This provides much higher granularity to the data as opposed to traditional thermocouples. Overall, IR imaging offers many benefits over traditional forms of temperature measurement, and it is useful in a variety of scenarios for which a detailed comparison can be seen in Table 1.

1.6. Computer vision concepts

Computer vision is the application of numerical algorithms to image data for the purpose of extracting important parameters. There are a few important concepts to understand; all are based on the fact that an image from the IR camera is basically an $M \times N$ matrix of values corresponding to the temperature measured pixels in the array. The first concept is focused on histograms, which are plots of the pixel quantity and a given intensity. Histograms are useful for finding the probability of a certain value as:

$$P(i) = \text{hist}(i)/(M \cdot N) \quad (3)$$

Next, basic algebraic operations can be performed on the matrix for the extraction of useful parameters. Subtracting two images can give information about motion (as only certain pixels will change value), multiplication by an array of 0 or 1 values in given shapes can mask off part of an image, and division by a matrix of intensities can be used to compensate for non-uniformities in the lighting or optics. Another important concept is pixel neighborhoods, or in other words, a pixel and the four pixels directly bordering it. Neighborhoods conform to the principle of symmetry such that:

$$(i, j) \in \mathcal{N} \leftrightarrow (-i, -j) \in \mathcal{N} \quad (4)$$

and a path is defined as a set of ordered indices where consecutive indices are adjacent, or $\mathcal{P} = (I_1, I_2, \dots, I_N)$ such that $I_i \sim I_{i+1}$ where $\forall_i = 1, \dots, n-1$. Finally, thresholding can be used to segment an image based on a given condition,

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases} \quad (5)$$

Using these concepts an image can be manipulated to extract important information, such as the flow regime (slug, laminar or turbulent), and the position and properties of given fluid slugs.

In laminar flow the standard deviation of measurements across a channel should be relatively small, and a consistent thermal gradient should be seen along the walls. This is because non-turbulent (low Reynolds Number) flow has a parabolic flow profile, with a consistent velocity in the center and decreased velocity near the edges. In turbulent (high Reynolds Number) flow the standard deviation of measurements would be higher and a consistent thermal boundary would not be observed near the walls. This is due to fluid mixing during flow and lack of a clear flow profile. Finally, slug flow should also be clearly visible due to the different thermal nature of the different fluids. Certain fluids would change temperature at different speeds, and thermal gradients could be observed between slugs, making identification and segmentation rather straightforward. Overall information from thermal measurements could be used to elucidate the flow regime in a microfluidic system.

Canny edge detection is another important topic to consider in Computer Vision. This algorithm uses a set of operations to find edges in images for segmentation purposes (Canny, 1986). The first step in the algorithm is to apply a Gaussian filter to remove noise from the image. Noise can greatly impact the performance of the algorithm and even cause false detections, so it is important to smooth it out. Gaussian filters with a various kernels can be used depending on the application and signal quality. Next, the Hessian Matrix is computed such that

$$\begin{bmatrix} \frac{\partial}{\partial x^2} & \frac{\partial}{\partial x \partial y} \\ \frac{\partial}{\partial x \partial y} & \frac{\partial}{\partial y^2} \end{bmatrix} \cdot I(x) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix} \quad (6)$$

where the principle 2nd derivatives give then eigenvalues

$$\begin{bmatrix} \lambda_1 & \\ & \lambda_2 \end{bmatrix} \quad (7)$$

and the principle directions are given by the eigenvectors $[e_{v1}, e_{v2}]$. This is implemented in the discrete form by first calculating the 2nd derivatives in the 4 raster directions, then choosing the direction with either the minimum or maximum 2nd derivative, and applying a discrete mask of either $[1, -2, 1]$ for linear changes or $1/2 \cdot [1, -2, 1]$ for diagonals. Finally, the peak intensity in an image can be found by taking

$$n^- = \nabla(G \otimes I) / |\nabla(G \otimes I)| \quad (8)$$

then finding the neighborhood closest to $\bar{n}$, finding the angle α to this neighborhood, and continuing the search until a peak is found. Overall, the methods of Canny edge detection are very useful for isolating segments of interest in a 2D image.

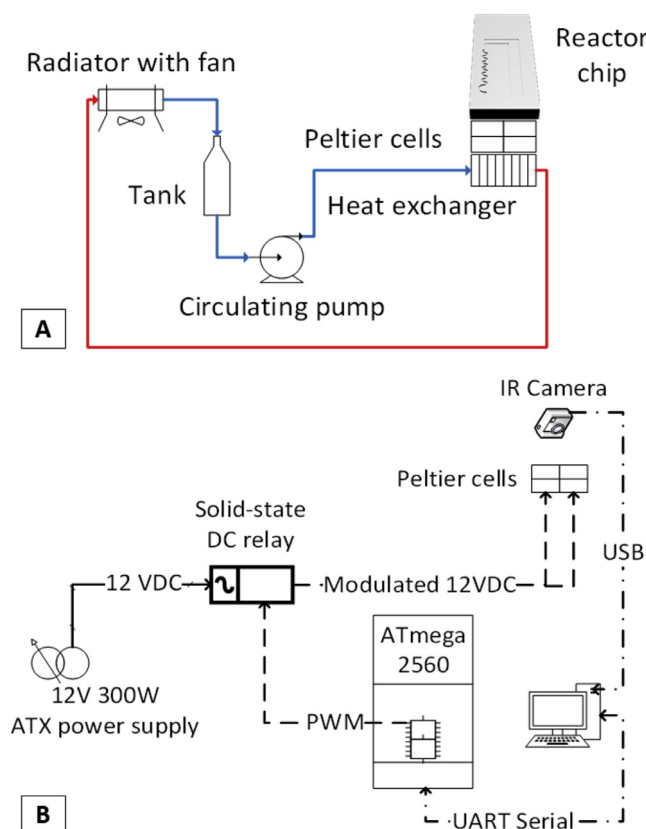


Fig. 2. (A) Overview of experimental setup for heat removal, (B) overview of experimental setup for control and data collection.

2. Methods

2.1. Thermal control setup

Temperature control was provided by a water-cooled thermoelectric module operating with a high-speed PWM controller. The 61 W Marlow TR060-6.5-40L thermoelectric module was connected directly to an Eaton D96115ACZ3 solid-state relay module. In turn, the relay was connected to a generic 300 W ATX-style power supply. A voltmeter was placed on the same rail in the power supply to monitor that no voltage drop was occurring due to load fluctuations. The gate of the relay was connected to an ATmega 2560 PWM output (Arduino Mega) running at 8-bit resolution and ~ 490 Hz. In practice, it is possible to increase the resolution up to 16-bits using standard controllers and up to 32 bits for pico-second level control using specialized hardware.

Thermal dissipation for the thermoelectric module consisted of a water-cooling system comprised of a thermal block, tank, pump, radiator, and fans. The module was affixed to the block using thermal compound (Protronix Series 7), and a water-glycol solution was circulated through the block. Water would enter one side, pick up heat from the module, and carry it to the outlet. From here it would travel to the 360 mm radiator with three 120 mm fans operating at 1650 rpm with ~ 110 m³/h of air flowrate. The water would then flow into a small expansion tank with a bottom-mounted circulating pump. A diagram of the system can be seen in Fig. 2(A). This system removed heat very efficiently and no significant rise in cooling water temperature was observed when the system was operating at load. Since thermoelectric cells operate as heat pumps with a defined maximum ΔT , the radiator can be replaced with a chiller to provide sub-freezing or potentially even cryogenic operation.

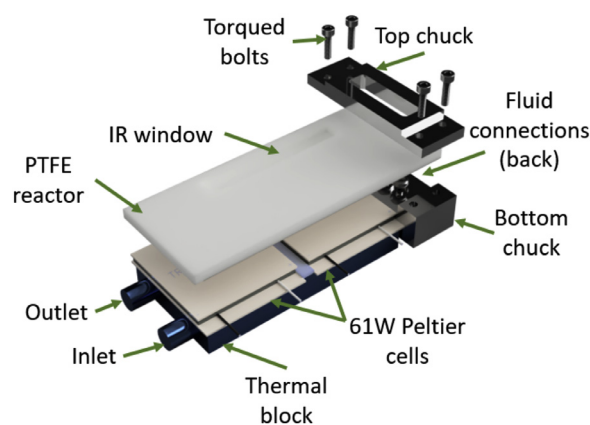


Fig. 3. Overview of microreactor and thermal control system. The reactor was machined from PTFE and the IR window is a very thin layer of IR-transparent PTFE film over the channel.

2.2. Control and data acquisition

Data acquisition was provided by an ICI 9640P IR camera running at 640×480 resolution with a VOx UFGA detector measuring a spectral band of 7–14 μ m at a sensitivity of $< 0.02^\circ\text{C}$ at 14-bits and 30 Hz. Data was fed using USB to a computer with an Intel i7-8550U CPU, 16 GB of RAM and a nvidia GEFORCE 980 M running MATLAB R2017B and a proprietary Direct Link Library (DLL) to extract data from the IR camera. Instructions from the algorithm about PWM setpoints were sent, via USB, to the ATmega 2560 using up to a 1 Mbps connection. This setup provides a robust platform for various experiments, allowing for very fast and precise temperature control and a systems diagram can be seen in Fig. 2(B).

A representative microfluidic system was constructed from bondable PTFE and placed on top of the thermoelectric modules. Two channels were machined into the PTFE block, a second piece placed over it, and both were bonded at 360°C while secured in an aluminum chuck. Temperature was measured in a serpentine channel which had a 1 mm square cross-section and was 30 mm long. The whole chip measured 41.5 mm wide, 100 mm long and 10 mm thick. An IR window was then machined into the finished reactor. All CNC operations were carried out using a Tormach PCNC440 and Autodesk Fusion 360 for CAM. The reactor was then placed into a flow chuck and aligned with the thermoelectric modules. PTFE was chosen because it provides very good chemical resistance necessary for future experiments while also being transparent to IR in the desired wavelengths. A thin layer of thermal compound was placed between the reactor and the thermoelectric cells to facilitate heat transfer.

The reactor-thermal system (seen in Fig. 3) was placed inside of a vacuum chamber and all electrical and fluidic connections routed through the enclosure and sealed with epoxy. By placing the system in a vacuum, thermal losses to the surrounding environment are minimized and the quality of data produced by the IR camera is improved.

2.3. System training

To perform training, the algorithm would first establish communications with the microcontroller and the IR camera. Next, the system would send a 0% setpoint and monitor the response to ensure that the system is performing as expected. Then, a 100% signal is sent and after equilibrium is reached this is used as the high-temperature shutoff limit of the system. When the system is at a low temperature, a flow of liquid would be established into

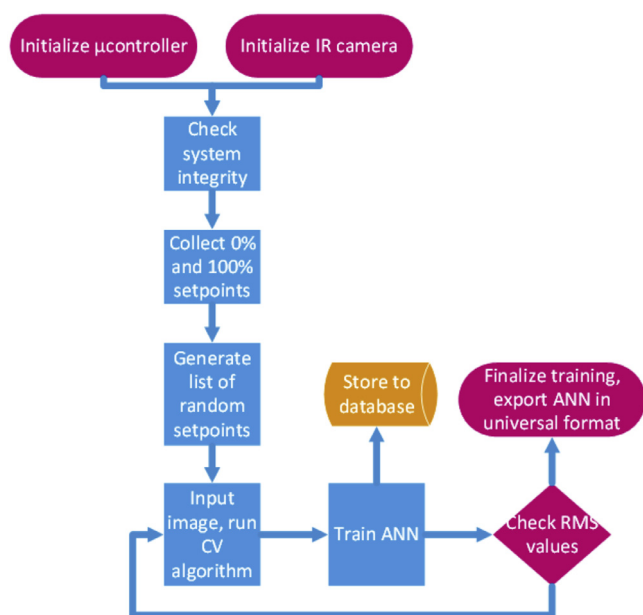


Fig. 4. Flowchart for control training algorithm.

the channel. This would allow the computer vision algorithm, using a Canny filter, to establish the boundaries of the thermally controlled channel. Knowing the minimum and maximum temperatures achievable in the setup, an array of random set points is generated. Randomness ensures that as the training algorithm varies between the points, the effects of consecutive and repetitive changes in temperature do not affect the final training. The system then sends the controller each individual set point, waits for the temperature to equilibrate, and retrain the ANN using the new data, thereby correlating desired temperature with applied voltage to the thermoelectric devices. A flowchart of the process can be seen in Fig. 4. Training is repeated until a desirable result is reached, as defined by the performance characteristics of the ANN (Root Mean Squared error values). Additional information can be included in the training algorithm, such as the flowrate of reactants, ambient temperature, vacuum pressure, etc. to increase the quality of the algorithm's predictions during operation.

3. Results and discussion

3.1. Computer vision

Performance of the IR transparent flow channel with the thermal control system was verified, and even at very low flowrates

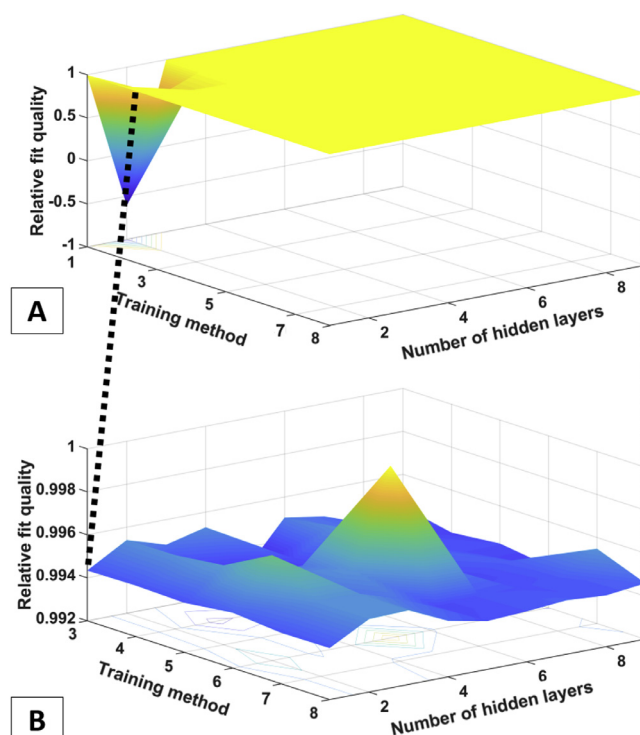


Fig. 6. Surface plot showing the relative fit quality vs. training method and the number of hidden layers. B indicated an enlargement of the indicated area of A to make the Z-axis scale visible. 1 = Levenberg–Marquardt (Damped Least Squares), 2 = quasi-Newton backpropagation, 3 = Resilient backpropagation (RProp), 4 = Scaled conjugate gradient backpropagation, 5 = Conjugate gradient backpropagation with Fletcher–Reeves updates, 6 = Conjugate gradient backpropagation with Powell–Beale restarts, 7 = Conjugate gradient backpropagation with Polak–Ribière updates, 8 = One-step secant backpropagation.

and ΔT , a visible thermal gradient was very quickly observed. A representative set of frames captured from the camera can be seen in Fig. 5, both with no flow and low Re-flow. A Canny filter (weight = 0.7) is applied to the frame to determine in which area temperature is to be computed. This filtering is especially useful when changing flow regimes or alignment of the image, as it very quickly, and with minimal computing power, established the controlled boundaries inside of the thermal envelope established by natural conduction within the material. This is a noteworthy distinction from systems controlled by thermocouples, because thermocouples can only sense the temperature in the exact point where they are located and do not account for natural conduction in the material. In certain systems the thermocouple's thermal mass may be large enough that this conduction within the sensing

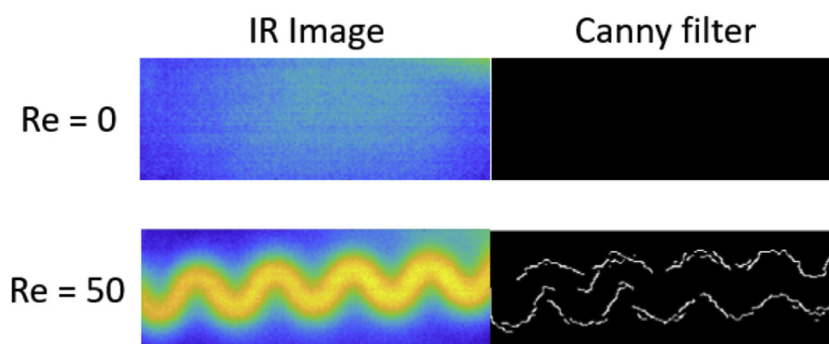


Fig. 5. IR image as seen by the thermal camera for no flow ($Re=0$) and laminar flow ($Re=50$) both as a raw image and with a Canny filter applied. The volume of the channel imaged here is $\sim 22.5 \mu\text{L}$.

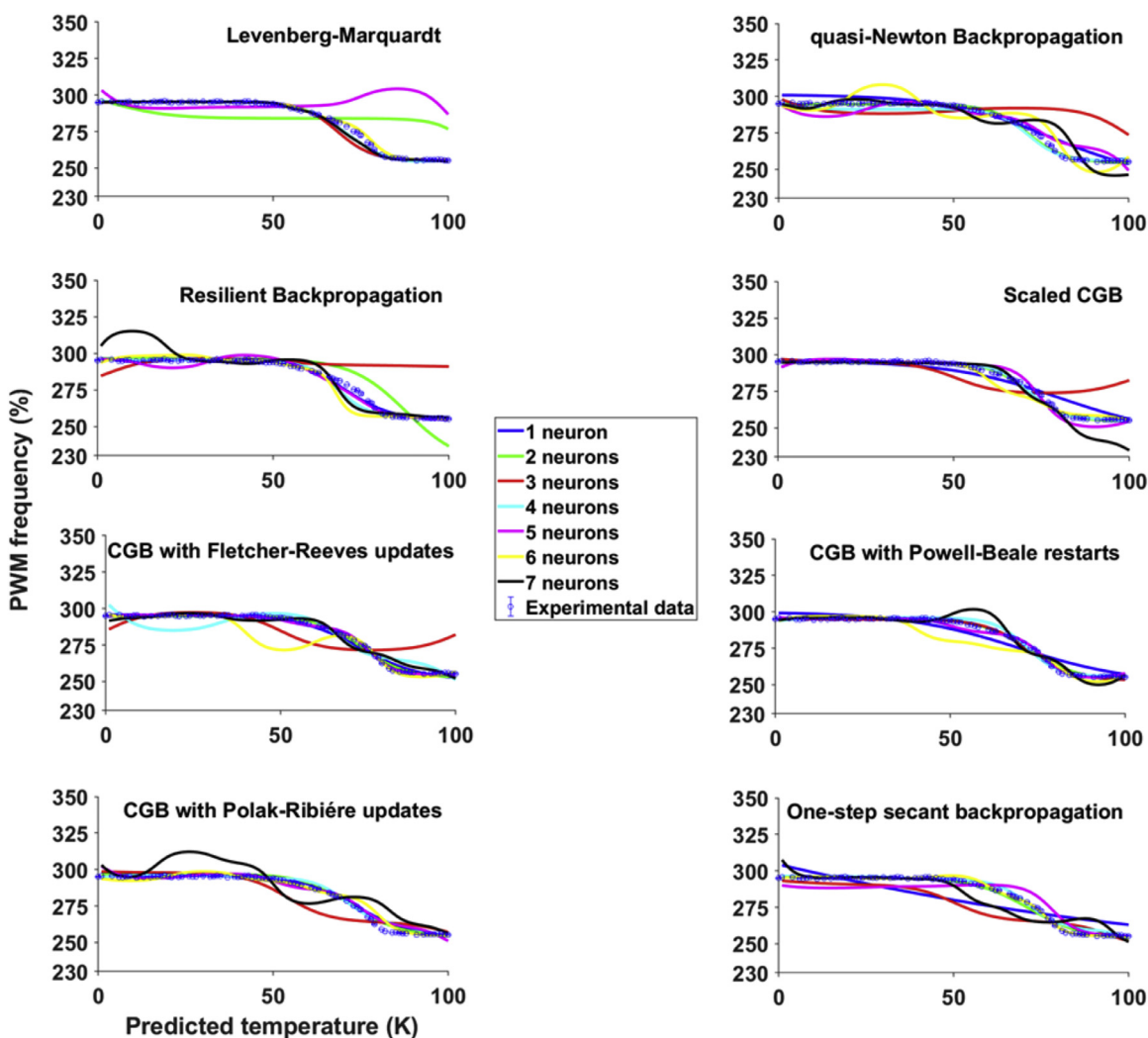


Fig. 7. Raw data for the comparison of different training methods. It can be seen that as the number of hidden neurons increases, the quality of the fit becomes better, until about 4 neurons, when the quality of the fit rapidly degrades as the data becomes overrepresented.

junction itself would impact the quality of the measurement. The data can also be segmented at various points, both laterally and horizontally, and the average from each of these regions given a certain weight in the controller.

3.2. Network training and selection

A series of experimental trials was performed using 1000 sequential random set point changes to monitor the time-based behavior of the system, including monitoring for any kind of hysteresis or set point drift. The self-training system would take a set point, monitor the behavior of the system, and retrain itself using the new data.

A number of different training algorithms were inspected along with various numbers of hidden neurons. The results are summarized in Fig. 6, with detailed representative fits given in Fig. 7. It was observed that the fit of one neuron exactly resembles a sigmoid, as one would expect. Using two neurons resulted in fit variability, including some training states which are clearly not physically representative of the system's behavior. With more hidden neurons the quality of the fit progressively improved until the system is overrepresented after four neurons (theoretically paralleling the transfer functions of measurement, control, actuation and the system). Overrepresentation results in oscillations around the

correct value, and in some cases, has led to undamped behavior in the system. In Fig. 6, a comparison of the quality of the fit of different training algorithms (1)–(7) is shown vs. the number of hidden layers. The one-step secant backpropagation method completely failed to represent the system in some cases, but the other methods yielded relatively similar results. Upon close examination, it is observed that the best representation is achieved by a model that uses the conjugate gradient methods with four neurons and the best fit being derived from the Conjugate Gradient Backpropagation with Powell–Beale Restarts methods. It is predicted that the Conjugate Gradient Backpropagation methods result in the best fit because the training algorithm performs a search at each iteration. This ensures that the search direction of the system is correct at each point during the training routine, eliminating the inconsistencies caused by multiple iterations following a search in the incorrect direction.

3.3. Network optimization

Upon determination that the Conjugate Gradient Backpropagation with Powell–Beale Restarts method yielded the best fit quality, it was analyzed what kind of impact the number of training points had on the system. This is useful in determining the least possible amount of necessary trials to train the system. Fig. 8 shows

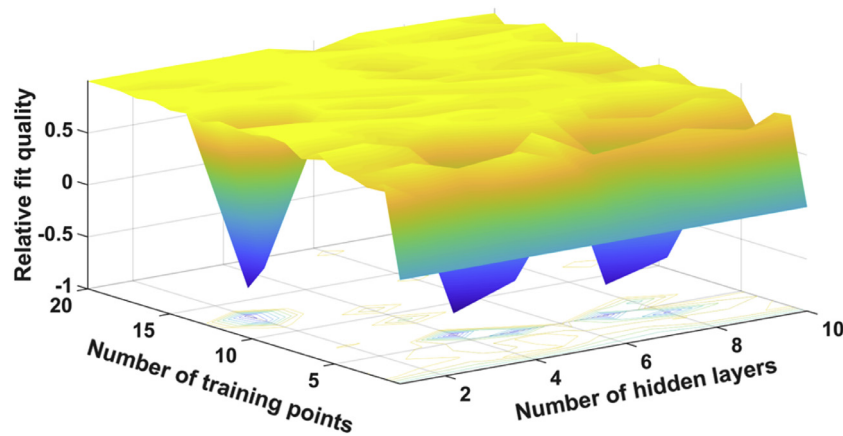


Fig. 8. Surface plot showing the fit quality vs. number of training points and hidden layers for Conjugate gradient backpropagation with Powell–Beale restarts. For very few training points (<5) the fit quality is rather poor (except for 1 and 2 data points, which corresponds to a linear fit). Fit quality improves at ~ 10 training points and remains mostly constant except for one dip around data point 12. This was caused because the data point in the setpoint array had a very large change from the previous 11, causing a brief period where the network was poorly trained. This can be overcome in future trials by ensuring that the first few setpoints adequately cover the dynamic range of the system.

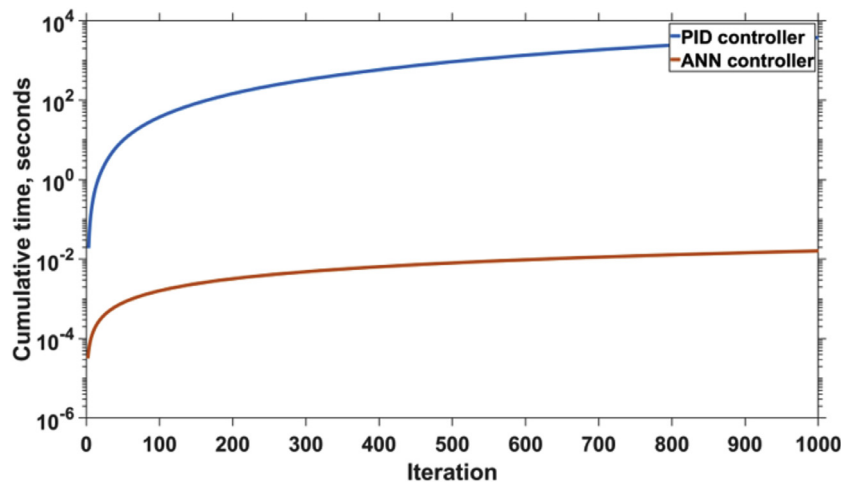


Fig. 9. Comparison of computation time for an ideal PID and an ANN controller for the same model system using 1000 random setpoint changes.

the relationship of fit quality to the number of training points and hidden layers. As before, it is observed that after 4 hidden layers the fit develops oscillatory behavior. Fits assessed using one or two data points resulted in good R -values simply due to the linear nature of the fit represented. It was also observed that systems with fewer hidden layers were sensitive to changes in setpoint between trials. Finally, it was observed that fifteen data points was the minimum necessary to resiliently train the controller.

It is important to note that overrepresenting the system (more than 4–5 neurons) can lead to oscillatory or undamped operation of the network. A good fit must establish a balance between the bias and the variance. If the model has too few hidden layers, it displays characteristics of high bias, meaning the algorithm fails to represent the data fully. It is predicted that when the network has too few layers to adequately represent the real number of transfer functions in the system (measurement, control, actuation and the system itself) it fails to fully model the physics. If the network has too many layers, it can display high variance, meaning it will catch onto random noise. If there are too many layers there are two likely scenarios—oscillatory and undamped. In the oscillatory scenario the controller controls the temperature around the setpoint, but never settles to the correct value. In the undamped scenario, the controller sets the temperature to stay at some totally unrelated value, either higher or lower. For this reason, it is

necessary to choose the right number of layers for the system, which we found to be the same as the number of transfer functions in the system.

3.4. Comparison with PID control

Another important element of the system's performance was the time required to perform computations within the control loop. This is often an area where an inevitable time delay is introduced which limits the amount of useful control actuations that could be accomplished per second. This also has implications for the computational power required by the control loop, as an application of this technology lies in low-power IoT or distributed systems. Fig. 9 depicts the time required for the calculations during a setpoint change, comparing the ANN run over a matrix of set point changes with a traditional ideally-tuned PID simulated on the experimental data using transfer functions in MATLAB. It was observed that when the ANN and PID are both run under ideal computation conditions the total cumulative calculation time is two orders of magnitude lower for the ANN. This has significant implications for both high speed and low power systems.

Further comparing the ANN and PID control methodologies, it should be noted that the rise time, settling time, and overshoot were all improved. The rise time was reduced from 41.4 to 32.6 s

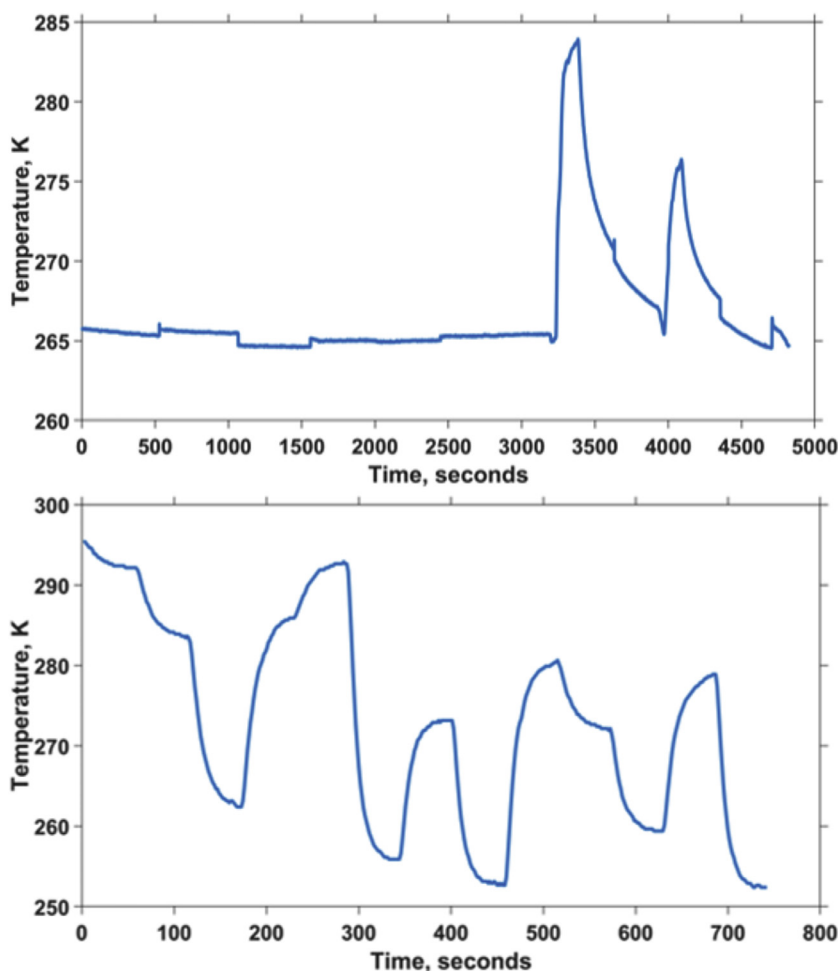


Fig. 10. Setpoint tracking for the ANN controller, showing temperature (Kelvin) vs. time. Both small changes to setpoint (@500 s, 1000 s, 1500 s, 2500 s) and very large disturbances (3300 s, 4000 s, 4700 s) can be seen. Below an example for setpoint tracking for large step changes can be seen.

(21.26% improvement), the settling time was reduced from 144 to 11.43 seconds (92.06% improvement), and the overshoot was reduced from 6.08% to ~ 0.2 °C.

3.5. Setpoint tracking

The final and most important aspect validated for the new control methodology was the ability of the controller to track setpoint changes and disturbances in the system. In Fig. 10, it can be observed how the setpoint was varied several times up and down by a small degree and the system was able to track very quickly with only a minor overshoot on upward changes, settling into the new setpoint within seconds. In the last third of the chart, three large disturbances are seen and the ability of the controller to respond and return to setpoint is observed. The system is changed from a flowrate of 0.2 mL/min to 1, 3 and then 5 mL/min. The flowrate along with the temperature were input into an ANN which would adapt its training between runs. It is seen that by the third setpoint change the system “knows” how to react and minimizes the spike quite dramatically. It is also worth noting that the change in input flowrate results in thermal diffusion through the reactor. Since thermocouples in microfluidic systems are usually not placed directly into the flow path, this diffusion time would have introduced a time delay into a traditional thermocouple-based system. Overall, this tentatively validates that such a control algorithm can be used for control chemical reactor systems.

4. Conclusions

The findings presented here demonstrate that an Artificial Neural Network can be self-trained with IR camera images in order to control the temperature of a microfluidic reactor with high speed and accuracy. A system was constructed using two 61-watt Peltier thermoelectric cells with a water-cooling system to remove heat. A PTFE microreactor was fabricated using CNC machining and placed onto the cells. The system was trained to predict the correct setpoint based on the required temperature of the system and the flowrate. It was discovered that a network trained using the Conjugate Gradient Backpropagation with Powell–Beale Restarts method with four hidden neurons best represented the system. Tests were conducted to ensure the stability and response of the system to setpoint changes and disturbances. The performance of the system was also compared with traditional PID control and thermocouple measurements. The results of the comparison have shown the system to have a faster settling time and less overshoot while using marginally fewer computing resources. The system was able to maintain the setpoint with high accuracy and it trained itself to reduce temperature changes upon a change in flowrate through the reactor. Systems like these will be able to further adapt their training over time, gaining resiliency for all sorts of different process conditions, ranging from outdoor temperature to feed composition. Overall, these results indicate that such a control methodology could be useful in both research and industrial applications.

It should also be noted that the use of an IR camera instead of thermocouples offers several advantages. First, an IR camera

allows for non-contact measurement. This means that more points can be scanned in a system, and thus the granularity of the data could be greatly enhanced. This also offers advantages for systems that either operate at extreme temperatures, extreme conditions, or require high levels of sterility (such as bioreactors). It also allows for better quantification of heat transfer in systems and reduces the time delay caused by thermal diffusion to and through a thermocouple. Second, an IR camera offers advantages both in the accuracy and speed that data is delivered as compared to traditional thermocouples. Finally, IR cameras have self-calibrating abilities built-in, allowing long-term process stability over time. All in all, the use of IR cameras instead of thermocouples offers several advantages to engineers focusing on process and controls.

It is our prediction that systems like these will be of interest in IoT, distributed, or otherwise resource-limited environments where it is currently impractical to set up multiple thermocouples with readers. It could also be of use in areas where very fast and precise control are required or the tuning of the system changes over time. Finally, systems like the one presented here could have implications in data mining, where it would be useful to trace how certain (seemingly disjoint) inputs impact production systems.

Declaration of interest

None.

Funding sources

This material is based upon work supported by the National Science Foundation under Grant number [CBET-1701393](#). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Acknowledgments

The authors acknowledge the contribution of the NYU Makerspace staff, particularly of Victoria Bill and Sean Kennon who were instrumental in the procurement and setup of the CNC machine.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.compchemeng.2018.11.016](#).

References

- Ahmad, M.W., Mourshed, M., Rezgui, Y., 2017. Trees vs Neurons: comparison between random forest and ANN for high-resolution prediction of building energy consumption. *Energy Build.* 147, 77–89. doi:[10.1016/j.enbuild.2017.04.038](#).
- Ahmadi, M., Nekoomanesh, M., Arabi, H., 2009. New approach in modeling of metallocenecatalyzed olefin polymerization using artificial neural networks. *Macromol. Theory Simul.* 18, 195–200. doi:[10.1002/mats.200800088](#).
- Battiti, R., 1992. First- and second-order methods for learning: between steepest descent and Newton's method. *Neural Comput.* 4, 141–166. doi:[10.1162/neco.1992.4.2.141](#).
- Canny, J., 1986. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8, 679–698. doi:[10.1109/TPAMI.1986.4767851](#).
- Chan, C.W., Huang, G.H., 2003. Artificial intelligence for management and control of pollution minimization and mitigation processes. *Eng. Appl. Artif. Intell.* 16, 75–90. doi:[10.1016/S0952-1976\(03\)00062-9](#).
- Dayal, B.S., Taylor, P.A., Macgregor, J.F., 1994. The design of experiments, training and implementation of nonlinear controllers based on neural networks. *Can. J. Chem. Eng.* 72, 1066–1079. doi:[10.1002/cjce.5450720618](#).
- Dennis, J.E., Schnabel, R.B., 1985. *Numerical methods for unconstrained optimization and nonlinear equations*. J. Am. Stat. Assoc. 80, 247–248.
- Dudzik, S., 2011. Investigations of a heat exchanger using infrared thermography and artificial neural networks. *Sens. Actuators A Phys.* 166, 149–156. doi:[10.1016/j.sna.2010.12.001](#).
- Fletcher, R., 1964. Function minimization by conjugate gradients. *Comput. J.* 7, 149–154. doi:[10.1093/comjnl/7.2.149](#).
- Fu, B.R., Ting, Y.C., Lee, C.F., Huang, Y.J., Su, Y.C., Tseng, F.G., Pan, C., 2016. Real-time monitoring of a micro reformer integrated with a microchannel heat exchanger by infrared thermography and high-speed flow images. *Int. J. Hydrogen Energy* 41, 18610–18620. doi:[10.1016/j.ijhydene.2016.08.023](#).
- Haber, J., Kashid, M.N., Borhani, N., Thome, J., Krtischil, U., Renken, A., Kiwi-Minsker, L., 2013. Infrared imaging of temperature profiles in microreactors for fast and exothermic reactions. *Chem. Eng. J.* 214, 97–105. doi:[10.1016/j.cej.2012.10.021](#).
- Hagan, M.T., Demuth, H.B., Jesús, O.De., 2002. An introduction to the use of neural networks in control systems. *Int. J. Robust Nonlinear Control* 12, 959–985. doi:[10.1002/rnc.727](#).
- Ingham, R.J., Battilocchio, C., Fitzpatrick, D.E., Sliwinski, E., Hawkins, J.M., Ley, S.V., 2015. A systems approach towards an intelligent and self-controlling platform for integrated continuous reaction sequences. *Angew. Chem. Int. Ed.* 54. doi:[10.1002/anie.201409356](#).
- Khalid, M., Omatu, S., 1992. A neural network controller for a temperature control system. *Control Syst. IEEE* doi:[10.1109/37.165518](#).
- Khoda, K.M., Ltu, Y., Storey, A.C., 1992. Generalized Polak–Ribire algorithm. *J. Optim. Theory Appl.* 75, 345–354. doi:[10.1007/BF00941472](#).
- Kourti, T., Macgregor, J.F.J.F., 1995. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemom. Intell. Lab. Syst. 28*, 3–21. doi:[10.1016/0169-7439\(95\)80036-9](#).
- Kresta, J.V., Macgregor, J.F., Marlin, T.E., 1991. Multivariate statistical monitoring of process operating performance. *Can. J. Chem. Eng.* 69, 35–47. doi:[10.1258/phleb.2011.010101](#).
- Lee, P.G., 2000. Process control and artificial intelligence software for aquaculture. *Aquac. Eng.* 23, 13–36. doi:[10.1016/S0144-8609\(00\)00044-3](#).
- Levenberg, K., 1944. A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.* 2, 164–168.
- Li, C., Skidmore, G.D., Han, C.J., 2011. Uncooled VOx infrared sensor development and application 8012, 80121N. doi: [10.1117/12.887113](#).
- Li, F., Karpathy, A., Johnson, J., 2016. Lecture 04: backpropagation and neural networks part 1 1–84.
- Liu, T.L., Pan, C., 2016. Infrared thermography measurement of two-phase boiling flow heat transfer in a microchannel. *Appl. Therm. Eng.* 94, 568–578. doi:[10.1016/j.applthermaleng.2015.10.084](#).
- Maltarollo, V.G., Honorio, K.M., Silva, A.B.F.da., 2013. Applications of artificial neural networks in chemical problems. *Open Access*. doi:[10.1007/BF02706848](#).
- Manic, M., Sabharwal, P., 2011. Computational intelligence as a tool for small modular reactors, in: ASME 2011 small modular reactors symposium. ASME 299–310. doi:[10.1115/SMR2011-6544](#).
- Møller, M.F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 6, 525–533. doi:[10.1016/S0893-6080\(05\)80056-5](#).
- Moreira, M., Fiesler, E., 1995. Neural networks with adaptive learning rate and momentum terms. *Martigny-Valais Suisse*.
- Parkale, Y., 2012. Comparison of ANN controller and PID controller for industrial water bath temperature control system using MATLAB environment. *Int. J. Comput. Appl.* 53, 1–6. doi:[10.5120/8390-1967](#).
- Polikar, R., Shinar, R., Udpal, L., Porter, M., 2001. Artificial intelligence methods for selection of an optimized sensor array for identification of volatile organic compounds. *Sens. Actuators B* 80, 243–254. doi:[10.1016/S0925-4005\(01\)00903-0](#).
- Powell, M.J.D., 1977. Restart procedures for the conjugate gradient method. *Math. Program.* 12, 241–254. doi:[10.1007/BF01593790](#).
- Pradere, C., Joanicot, M., Batsale, J.-C., Toutain, J., Gourdon, C., 2006. Processing of temperature field in chemical microreactors with infrared thermography. *Quant. Infrared Thermogr. J.* 3, 117–135. doi:[10.3166/qirt.3.117-135](#).
- Psaltis, D., Sideris, A., Yamamura, A.A., 1988. A multilayered neural network controller. *IEEE Control Syst. Mag.* 8, 17–21. doi:[10.1109/37.1868](#).
- Riedmiller, M., Braun, H., 1993. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. *IEEE Int. Conf. Neural Netw. – Conf. Proc.* 1993–Janua 586–591. doi:[10.1109/ICNN.1993.298623](#).
- Romano, M., Pradere, C., Sarrazin, F., Toutain, J., Batsale, J.C., 2015. Enthalpy, kinetics and mixing characterization in droplet-flow millifluidic device by infrared thermography. *Chem. Eng. J.* 273, 325–332. doi:[10.1016/j.cej.2015.03.071](#).
- Uraikul, V., Chan, C.W., Tontiwachwuthikul, P., 2007. Artificial intelligence for monitoring and supervisory control of process systems. *Eng. Appl. Artif. Intell.* 20, 115–131. doi:[10.1016/j.engappai.2006.07.002](#).
- Vamvoudakis, K.G., Lewis, F.L., Ge, S.S., 2015. *Neural networks in feedback control systems*. Mechanical Engineers' Handbook. John Wiley & Sons, Inc., Hoboken, NJ, USA, pp. 1–52. doi:[10.1002/9781118985960.meh223](#).
- Xue, Y., Ray, N., 2017. Cell detection in microscopy images with deep convolutional neural network and compressed sensing 1–29.
- Zavadskaya, A., 2017. Artificial intelligence in finance: forecasting stock market returns using artificial neural networks. *Helsinki*.
- Zhang, J.S., Zhang, C.Y., Liu, G.T., Luo, G.S., 2016. Measuring enthalpy of fast exothermic reaction with infrared thermography in a microreactor. *Chem. Eng. J.* 295, 384–390. doi:[10.1016/j.cej.2016.01.100](#).
- Zhang, Y., Xu, J., Yuan, Z., Xu, H., Yu, Q., 2010. Artificial neural network-genetic algorithm based optimization for the immobilization of cellulase on the smart polymer Eudragit L-100. *Bioresour. Technol.* 101, 3153–3158. doi:[10.1016/j.biortech.2009.12.080](#).
- Zribi, A., Chtourou, M., Djemel, M., 2015. A new PID neural network controller design for nonlinear processes. *arXiv1512.07529* 1–10.