Microfluidic Injector Models Based On Neural Networks

R. Magargle*, J.F. Hoburg** and T. Mukherjee***

Carnegie Mellon University, Pittsburgh, PA 15213, USA
*magargle@cmu.edu, **hoburg@ece.cmu.edu, ***tamal@ece.cmu.edu

ABSTRACT

A functional modeling technique is developed for components of a microfluidic system and applied to three common injector topologies. This technique uses sparse numerical simulation to train a neural network to provide compact, explicit, and accurate component models. The resulting models are compatible with analytic system simulation environments, making complex design synthesis and optimization feasible, unlike standard techniques using computationally expensive numerical simulation. The neural network models are accurate to numerical simulation with mean squared errors less than 10^{-4}. In explicit form, the neural network models are very fast taking less than 1s per evaluation.

Keywords: microfluidic, electrokinetic, injector, model, neural, network, system

1 INTRODUCTION

The simulation of complex LOC systems requires computationally expensive numerical solutions to partial differential equations. The *design* of LOC systems then requires many repeated simulations. These two compounding difficulties can quickly lead to computationally infeasible descriptions. A much more efficient alternative for CAD of microfluidic systems involves functional decomposition into a series of interconnected blocks, as previously proposed for the mixer, injector, and separator [1, 2]. Functional neural network (NN) modeling makes such an efficient decomposition possible. In this paper, a neural network has been used for the first time, to the authors' knowledge, in microfluidic applications to create models for the injector.

The importance of the injector as a component in a microfluidic separation system derives from the fact that it defines the shape and quantity of analyte that will be used for separation and analysis. Prior work described the various forms of microfluidic electrokinetic injectors, such as the tee, double-tee, cross, double-cross, and gated-cross [3-7]. A first generation injector model produced by the authors was specific to the cross injector and was defined by a two-dimensional parameter space [8]. This work improves on previous generation injector modeling by using neural network functional modeling concepts, as created in the context of VLSI CAD, to create a library of models for injectors, including the cross, double-tee, and gated-cross, each defined by a four-dimensional parameter space. The methodology is not specific to any one injector topology and has been used in CAD spaces with many more dimensions.

2 INJECTOR TOPOLOGIES

The cross injector, shown in Figure 1a, operates using two cross flows. In the loading stage, the analyte is pumped from the sample reservoir to the waste channel. After filling the injection chamber, defined by the intersection of the two perpendicular channels, the analyte is injected into the separation channel by using a cross flow from a buffer reservoir to the separation waste reservoir. In the loading stage, accessory fields can be used to focus the band in the injection chamber. In the dispensing stage, accessory fields can be used to pull the analyte from the injection chamber preventing sample leakage. In both cases, the accessory fields are assumed to be applied symmetrically.

The double-tee, shown in Figure 1b, has the same principle of operation as the cross. The difference between the two injectors is the offset in loading channels, which defines a larger injection chamber for the double-tee. The cross is a special case of the double-tee topology as seen in the analysis of the physical space shown later, however in this case, the two are treated separately since the double-tee is most commonly found with an offset between loading channels fixed at twice the channel width.

The gated-cross, shown in Figure 1c, has a significantly different principle of operation. The gated-cross uses two counter-flows to establish a gate within the injection chamber across which analyte does not convect. Once the gate is established, sample can be injected into the separation channel by floating a single node to flood the injection chamber, then reestablishing the gate using the counter flows to flush the sample into the separation channel.

3 METHODOLOGY

The three injectors in Figure 1 are used to demonstrate the more general methodology developed here. The same methodology can be applied to other microfluidic components. For any components four steps are included:

3.1 Defining the Physical Variable Space

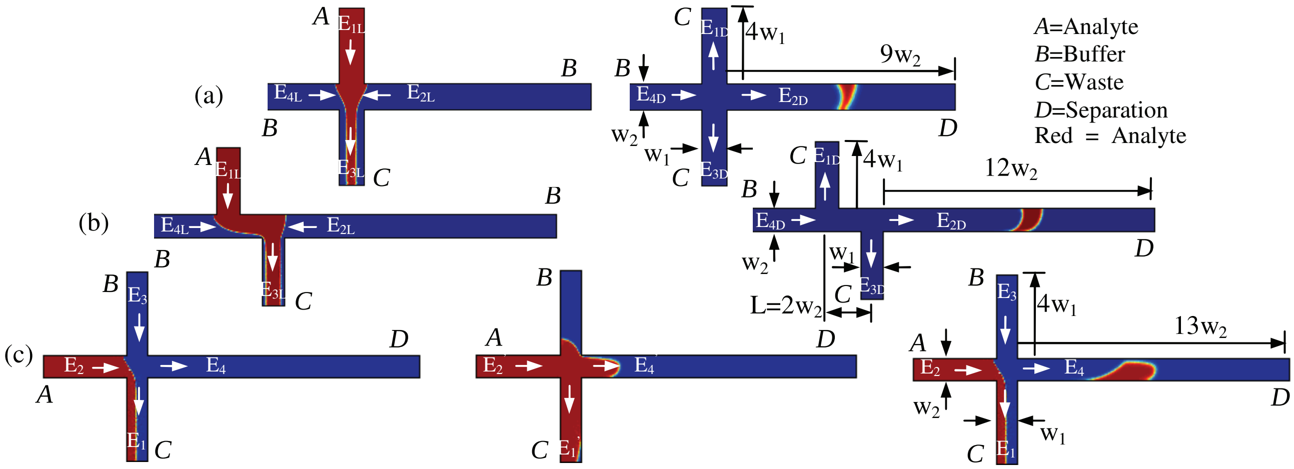

Figure 1. Example injections and geometries for (a) cross, (b) double-tee, and (c) gated-cross.

This step defines the data used to train the neural network. First, all of the physical parameters defining the system are identified. Second, the performance functions describing the behavior of the device are chosen.

The (input) physical parameters fall into four categories: the species, electrical, transient, and geometric properties of the system. These physical parameters, respectively, are the independent variables of the desired (output) performance functions:

$$\overline{P} = \bar{f}\left(\overline{\mu}, \overline{E}, \overline{L}, \overline{T}\right) \tag{1}$$

The over-bar vector notation indicates that there could be any number of performance functions and any number of dependent variables in any of the four categories. For injectors, the performance functions are defined to be the peak height and variance of a Gaussian distribution for the transversely average concentration with the same area and variance as the actual output of the injectors. This reduces the complexity of the number of parameters that need to be passed between the injector and separator in a system simulation environment. The approximation becomes very accurate after the band travels a small distance into the separation channel. By examining two particles diffusing from opposite sides of the channel, a bound for when this approximation is accurate can be derived:

$$\tau = \frac{Pe_w}{L/w} < 2, \tag{2}$$

where Pe_w is the Peclet number based on the channel width, w, and L is the length the band must travel before reaching the detector or other significant source of dispersion.

Figure 2 shows the results of this approximation for a double-tee injection. The picture on the left shows a band immediately after the injector, and the picture on the right shows the good agreement between the approximate Gaussian parameters and the actual concentration profile after 4.7 mm.

Before the training data for the neural network is generated, Buckingham's pi theorem is used to minimize the number of independent variables. The result of this reduction for the injectors is summarized in Table 1.

The dynamic parameters of the three injectors, $\pi_{1(C,D,G)}$ through $\pi_{4(C,D,G)}$, are used to create the data to train the neural network. The remaining geometric parameters are fixed at their most common values, $\pi_{5(C,D,G)} = 1$ and $\pi_{6D} = 2$.

From a synthesis point of view, the cross is simply a special case of the double-tee, where $\pi_{6D} = 0$. For both of these topologies, the $\pi_{1(C,D)}$ and $\pi_{2(C,D)}$ parameters describe the ratio of the accessory fields to the driving fields. For the loading stage, this describes the amount of pinching that is applied to the incoming analyte stream, and for the dispensing stage, this describes the amount of pullback applied to the dispensed band. The $\pi_{3(C,D)}$ and $\pi_{4(C,D)}$ parameters describe the Peclet number for each stage.

The gated-cross has a set of parameters that differ from those of the cross and double-tee. The first parameter, π_{1G}, represents the extent to which the gate is closed. As discussed in [7], as long as $E_1 \geq E_2$, the gate will remain closed in the limit of no diffusion. As π_{1G} is reduced, the gate is further closed. The second parameter, π_{2G}, represents the ratio of the buffer electric fields (E_3, E_4) to the analyte electric fields (E_2, E_1). As π_{2G} increases, the

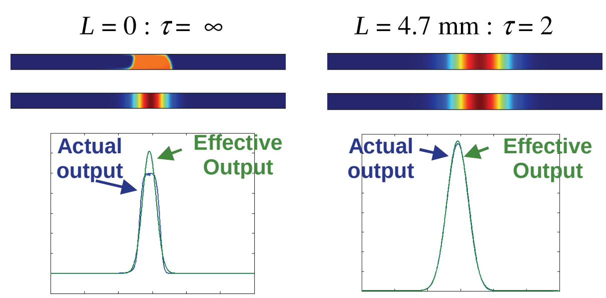

Figure 2. Actual output of double-tee injector (top) compared to effective parameterized output of analytical model (bottom). The band on the right traveled 4.7 mm, which is when $\tau = 2$. For these simulations $Pe = 186$.

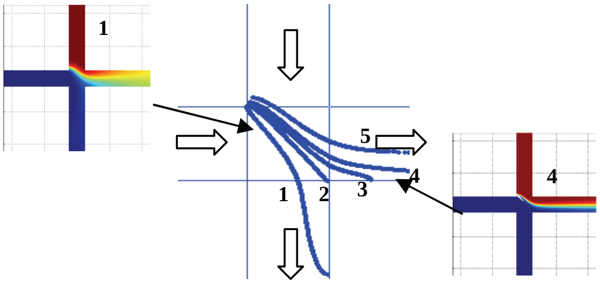

Figure 3. Gated-cross leakage tested at Peclet numbers from 19 at contour 1 to ∞ at contour 5. The contours are defined by 7% of the maximum concentration, and agree very well with the numerical and experimental results found in [7].

buffer fields become larger relative to the analyte fields. The third parameter, π_{3G}, represents the Peclet number during the loading phase. The final dynamic parameter, π_{4G}, measures the ratio of injected plug length, as determined by the floating time, T_{LD}, to the channel width.

A significant concern in the operation of a gated-cross injector is the leakage that can occur in the separation channel if the gate is not sufficiently closed, as seen in Figure 3. If the leakage is too great, the injector will not operate because the increased noise floor will make a separation impossible. A region of feasibility must be determined within which the injector can be modeled. Ermakov, et. al., analyzed this leakage as a function of the gate closure and the system Peclet number [7]. They determined a boundary indicated by 1% leakage of the flux of the analyte from the source reservoir into the separation channel. In this work, we show results for a more complete set of physical parameters to create a region of feasibility.

3.2 Numerical Simulation

As a one-time computational expense, numerical simulations are created to train the neural network on the performance function relationships. These simulations were completed using FEMLAB 3.0a [9]. Less than 200 simulations were needed per injector, with the average simulation taking approximately 1 hour, and the simulations can be run in parallel.

3.3 Neural Network Training

A neural network is a mathematical structure that is adept at learning non-linear functional relations and complex item categorizations [10]. The general principle of operation of the feed-forward neural network is shown in Figure 4a, and the topology of the two layer neural network used to model the injectors for this work is shown in Figure 4b. The feed-forward neural network is a set of nodes that are connected only to the layers above and below. Each node takes the weighted sum of the outputs of the layer below as the argument of an activation function. If the network is to perform non-linear regression, the hidden-layer consists of arbitrary bound non-linear activation functions, and the output nodes are unbound linear activation functions. If the network makes discrete classifications, then all nodes are bound non-linear functions. The adeptness of neural networks at functional modeling has been demonstrated in VLSI CAD where many of the techniques described here were inspired [11].

Using the algorithms of Matlab 7 [12], a single-hidden layer neural network with 20 nodes is trained to learn the data from the numerical simulation of the dynamic non-dimensional parameters using the Levenberg-Marquardt algorithm. To measure the accuracy of the trained neural network, the mean square error (MSE) is used to verify the network is not underfitting the data, and a k-fold cross validation (KFCV) technique is used to verify the network is generalized and does not overfit the training data [10].

Figures 5 and 6 show the results of the neural networks for only the double-tee and gated cross injectors, since the regular cross does not provide significant additional insight. Figure 6a shows the results of a classification network for the region of feasibility for the gated cross. Figure 6b shows this network working with a regression network for the performance functions, as evidenced by the data missing from the surfaces in infeasible regions. For all graphs representing the performance functions, a two-dimensional slice of the four-dimensional pi-space is extracted for visual clarity. For each performance function, the result of the sparse numerical simulations is shown on top compared to the much more dense results achieved with the neural network that has been trained on that data. The neural network evaluations take less than one second, allowing many thousands to be done in the time it takes one average numerical simulation to complete. This increase in speed is obtained while maintaining accurate results relative to the original numerical simulations, as summarized by low values of MSE and KCFV, shown in Table 2.

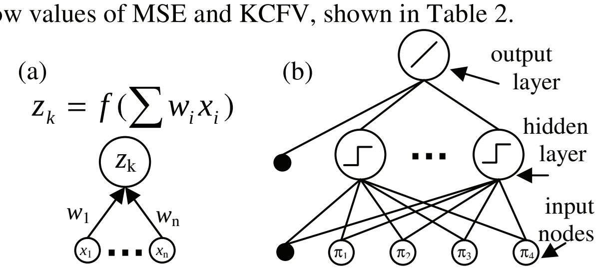

Figure 4. (a) The activation, z_k, of a node is a function of the weighted inputs of the previous layer's activations. (b) Feedforward neural network topology utilizing one hidden layer with non-linear activation functions, and a single output node using a linear activation function.

Cross	Double-Tee	Gated-Cross
$\pi_{1C}=E_{2,4L}/E_{1L}$	$\pi_{1D}=E_{2,4L}/E_{1L}$	$\pi_{1G}=E_2/E_1$
$\pi_{2C}=E_{1,3D}/E_{2D}$	$\pi_{2D}=E_{1,3D}/E_{2D}$	$\pi_{2G}=E_3/E_1$
$\pi_{3C}=\mu E_{3L}w_2/\kappa$	$\pi_{3D}=\mu E_{3L}w_2/\kappa$	$\pi_{3G}=\mu E_3 w_2/\kappa$
$\pi_{4C}=\mu E_{2D}w_2/\kappa$	$\pi_{4D}=\mu E_{2D}w_2/\kappa$	$\pi_{4G}=w_2/\mu E_3 T_{LD}$
$\pi_{5C}=w_1/w_2$	$\pi_{5D}=w_1/w_2$	$\pi_{5G}=w_1/w_2$
	$\pi_{6D}=L/w_2$	

NSTI-Nanotech 2005, www.nsti.org, ISBN 0-9767985-0-6 Vol. 1, 2005

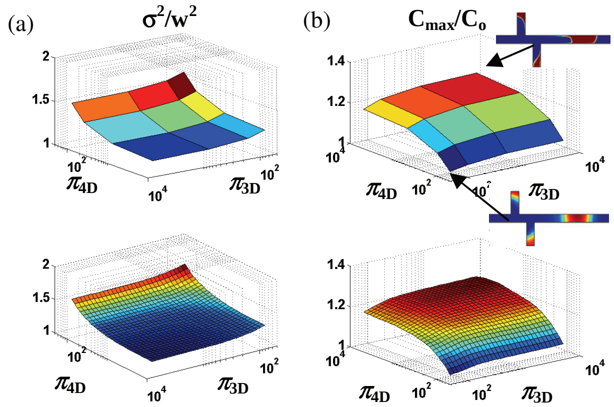

Figure 5. (a) variance and (b) effective peak concentration results for the double-tee, where π_{1D} and π_{2D} fixed at 0.05 and 0.3 respectively. The top plots show FEMLAB simulations at a course set of points, and the bottom plots show the much faster and higher resolution evaluations produced by the NN trained on the FEMLAB simulations.

3.4 Analytical Form Extraction

The feed-forward configuration of the neural network allows the extraction of an explicit relation for the output node of the following form for the two layer topology:

$$y_k = g_{out}\left[\sum_{j=0}^{M} v_{kj}\, g_{hid}\left[\sum_{i=0}^{N} w_{ji} x_i\right]\right] \qquad (3)$$

where y_k is the output of the network, $g_{out}[\xi]$ is the linear output node activation function, $g_{hid}[\zeta]$ is the non-linear hidden node activation function from Figure 4a, and v_{kj} and w_{ji} are the weights between layers.

4 CONCLUSION

The neural network modeling technique works well for injectors, and can be extended to other microfluidic components. The neural networking methodology, borrowed from the VLSI CAD community, allows functional models to be built with increased degrees of freedom compared to the previous generation injector modeling methodology. These injector models combined with other component models allow for a functional block decomposition of the system for efficient simulation. The speed and accuracy of these analytic block models present a far more feasible method of CAD than using numerical solutions of partial differential equations for whole complex systems.

Table 2. Neural Network MSE and Cross Validation: For all three injectors, both validation parameters are very small.

	MSE		KFCV		
	σ^2/w^2	C_{max}/C_o	k	σ^2/w^2	C_{max}/C_o
Cross	1.2e-4	2e-5	10	2e-6	9e-5
Double-Tee	1.8e-5	1e-5	8	3e-6	2e-5
Gated-Cross	4e-5	4.2e-5	10	1.1e-4	2.4e-4

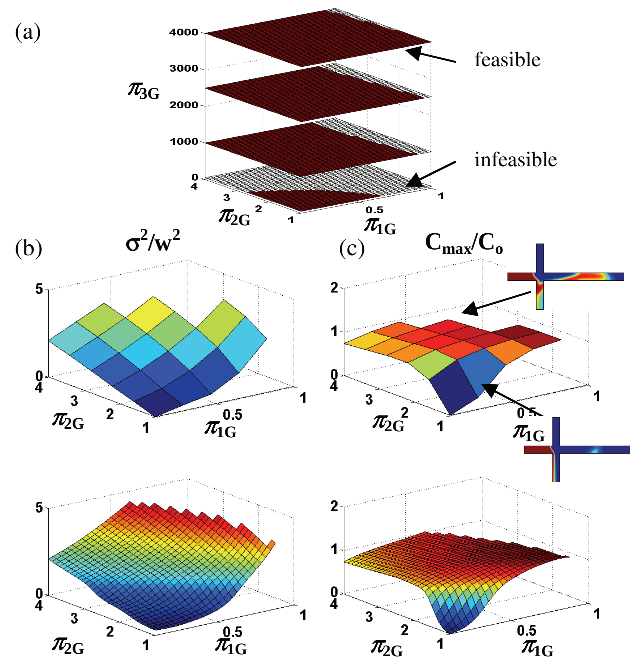

Figure 6. (a)Gated-cross feasibility space (dark red is feasible). (b) variance and (c) effective peak concentration results for the gated-cross, where π_{3G} and π_{4G} are fixed at 232 and 0.1208 respectively.

ACKNOWLEDGEMENTS

This research effort is sponsored by the Defense Advanced Research Projects Agency under the Air Force Research Laboratory, Air Force Material Command, USAF, under grant number F30602-01-2-0587.

REFERENCES

[1] Pfeiffer, A.; Mukherjee, T.; Hauan, S., Vol. 43, no. 14, *Ind. Eng. Chem. Res.* 2004, pp. 3539-3553.
[2] Wang, Y.; Lin, Q.; Mukherjee, T., Vol. 4, *Lab on a Chip* 2004, pp. 453-463.
[3] Jacobson, S.C; Hergenröder, R.; Koutny, L.B.; et. al., Vol. 66, *Analytical Chemistry* 1994, pp. 1107-1113.
[4] Ermakov, S.V.; Jacobson, S.C.; Ramsey, J.M., Vol. 70, *Analytical Chemistry* 1998, pp. 4494-4504.
[5] Shultz-Lockyear, L.L.; Colyer, C.L.; et. al., Vol. 20, *Electrophoresis* 1999, pp. 529-538.
[6] Deshpande, M.; Greiner, K.B.; West, J.; Gilbert, J.R., *Micro Total Analysis Systems* 2000, pp. 339-342.
[7] Ermakov, S.V.; Jacobson, S.C.; Ramsey, J.M., Vol. 72, *Analytical Chemistry* 2000, pp. 3512-3517.
[8] Magargle, R.; Hoburg, J.F.; Mukherjee, T., *Proceedings of the NanoTech* 2004, pp. 77-80.
[9] http://www.comsol.com
[10] Bishop, C.M. Neural Networks for Pattern Recognition. Oxford University Press, N.Y., 1995.
[11] Liu, H.; Singhee, A.; Rutenbar, R.A.; Carley, L.R., *Proceeding of IEEE DAC* 2002, pp. 437-442.
[12] http://www.mathworks.com