# The Limitations of Frequency Analysis for Dendritic Cell Population Modelling

Robert Oates, Graham Kendall, and Jonathan M. Garibaldi

School of Computer Science, The University of Nottingham
Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK
rxo@cs.nott.ac.uk
http://www.cs.nott.ac.uk/rxo

**Abstract.** In previous work we derived a mathematical model which allows the frequency response of a dendritic cell to be predicted. The model has three, key limitations: the model assumes that the intermediate co stimulatory molecule signal is constant; it is only possible to make predictions for a single cell and the model only takes into account the signal processing element of the dendritic cell algorithm, with no attempt to explore the antigen presenting phase. In this paper we explore the original model and attempt to extend it to include the effects of multiple cells. It is found that the complex interactions between the cells creates a one to many relationship between the input frequency and the output frequency. This suggests that traditional frequency-based techniques alone are unlikely to yield an effective automated tuning mechanism.

**Keywords:** Dendritic Cell Algorithm, Frequency Analysis, Tuning.

## 1 Introduction

The dendritic cell algorithm (DCA) is a relatively new addition to the field of artificial immune systems (AIS). The DCA can be viewed as a binary decision-making algorithm, for making Boolean choices in uncertain problem environments. Despite being successfully applied to several problems [1,6,7] little work has been carried out to characterise the operation of the algorithm. In [4] the high sensitivity of the algorithm to its input parameters is discussed. However, too little is known to automatically tune these parameters for a given application. Currently trial and error is used to identify the appropriate input parameters for new applications. This can be time-consuming and does not guarantee to find an optimal parameterisation. An automated tuning algorithm would be able to find an good quality set of input parameters for a given application and would provide a good basis for future comparisons between the performance of the DCA and other techniques. To derive such a tuning methodology it is important to mathematically characterise the behaviour of the algorithm. In [8] a mathematical model of a simplified version of the DCA was derived using frequency analysis. This model provides an accurate prediction of what information an individual cell will use to make decisions. The simplified model makes three, key

assumptions: the co stimulatory molecule ($CSM$) signal is assumed to be constant; the model only provides the response of a single cell and the correlation between signals and antigen is assumed to be trivial so is not modelled. A 'trivial' correlation between signal and antigen implies that the delay between an antigen appearing in the system and its affects appearing in the input signals is constant and negligible. It was proposed that such a model would be able to provide a tuning methodology for the DCA's input parameters, based on removing those frequencies that were deemed to contain misleading, noisy data. However, a preliminary tuning methodology based on the original model provided mixed results. The low quality of the tuning results has been attributed to the model's over-simplification of the algorithm.

An extension of the original mathematical model proposed in [8] should yield a more accurate estimate of the algorithm. Such an improvement could be the basis of a superior tuning methodology.

This paper is organized as follows. Section 2 provides an overview of the implementation of the DCA. Section 3 explains how the DCA can be modelled as a filter and the benefits of doing so. Section 4 discusses the limitations of the original model and provides some justification for these assumptions. Section 5 explores an extension of the original model to incorporate multiple cells, the results of which are given and discussed in Section 6.

## 2   Simplifying the Dendritic Cell Algorithm

The original dendritic cell algorithm is inspired by the biological dendritic cell. As a result many of the original parameters and signals were named after biological signals. For an introduction to the relevant biology, the interested reader should refer to [1,4]. In [8] a simplified version of the algorithm was presented which reduced the amount of processing carried out per cell. This optimisation also provides the basis for later extensions that make the frequency analysis of the algorithm possible.

### 2.1   The Original Dendritic Cell Algorithm

A full description of the original DCA is outside the scope of this paper. The interested reader is referred to [2] for pseudocode and a detailed description of the algorithm's implementation. In this section we provide a brief overview of the operation of the algorithm. The DCA attempts to assign a value to each input symbol between 0 and 1 that describes the likelihood that the antigen is a member of a target set. The algorithm has two parts: a decision making process and a state correlation process. A block diagram of the original dendritic cell algorithm is given in Fig. 1. The algorithm receives four inputs from the problem environment, a stream of enumerated symbols (termed 'antigen') and three normalised signals generated by application-specific heuristics. The output of the algorithm is a stream of enumerated symbols, each associated with a score between 0 and 1. The score is the algorithm's 'decision' about that symbol. The meaning of the

decision depends on the application that the algorithm is being applied to and the input heuristics being processed. The input enumeration stream provides an asynchronous list of symbols which represent the state of the problem environment. The three input heuristics convey the pertinent information for decision making from the problem environment to the dendritic cell population. These signals are expressed as real-numbers. Min-max normalisation is used to keep each heurisitic output within the range 0 and 1. The *PAMP* heuristic identifies situations that *only* occur when a positive output is required. The *Safe* heuristic identifies situations that *only* occur when a negative output is required. The nature of uncertain decision making environments means that these are rarely the inverse of one another. The *Danger* heuristic identifies situations that *always* occur when a positive decision is required, but *can* occur when a negative decision is required. All information that is provided to the algorithm from the problem environment is stored in a collection of asynchronous buffers termed 'tissue'. In the decision making element of the algorithm, the cells accumulate three internal signals based on weighted sums of the input signals. These internal signals are all real-numbers. The *IL*-10 signal increases proportionally to the *Safe* signal. The *IL*-12 signal increases proportionally to the *PAMP* and *Danger* signals, but can be decreased by the *Safe* signal. The *CSM* signal increases proportionally to the sum of all signals. When the accumulated *CSM* signal in a given cell reaches a cell-specific migration threshold, the cell makes a decision. If the accumulated *IL*-10 signal is greater than the accumulated *IL*-12 signal, the decision is negative. Otherwise the decision is positive. During the sampling life of the cell, it also collects samples of the symbols presented by the enumeration stream. The algorithm can be run continuously in real-time as when a cell finishes its sampling phase, it is removed from the population and a new cell is put in its place, maintaining a constant population of sampling cells. The state correlation element of the algorithm performs statistical analysis on the symbols collected by each cell and each cell's output decision. This correlation is designed to spot patterns between periods of signal activity and the presence of certain antigen.

A crucial factor in the performance of the algorithm is the probability distribution used to allocate migration thresholds to the cell population. If set too high the cells in the population will spend a large amount of time collecting antigen samples before making a decision. This means that correlation becomes an intractable problem as all cells will contain samples of almost all antigen.

If set too low, cells will be vulnerable to noise and in applications where there is a lag between antigen presentation and signal generation, the correlation process will fail.

## 2.2   Similarities to Neurons

The computational implementation of dendritic cells and neurons both involve performing weighted sums of input signals which are ultimately thresholded to produce a binary output. However, there are key differences between this algorithm and perceptrons. Firstly, perceptrons require supervised training periods to calculate the weightings for a given application, while the DCA uses expert
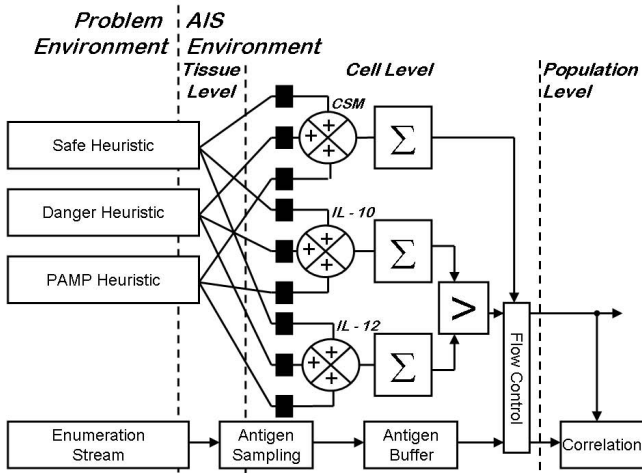
**Fig. 1.** A representation of the original DCA. The decision is made when the cumulated *CSM* is greater than the cell-specific migration threshold. The decision is positive if the cumulated *IL*-12 signal is greater than the cumulated *IL*-10 signal, otherwise the decision is negative.

knowledge to form the input heuristics. The pros and cons of supervised learning vs. encapsulation of expert knowledge are beyond the scope of this paper. Secondly, perceptrons form $N-1$ dimensional hyperplanes, (where $N$ is the number of inputs) and perform linear thresholding using that hyperplane to make decisions. The output from each dendritic cell can also be viewed as the result of applying a hyperplane threshold to the signals that it has been exposed to. However, the dimensionality of the hyperplane is a function of signal strength, as the number of samples taken before migration is determined by the accumulation of the *CSM*. Finally, the DCA processes both signal and antigen over varying size time windows which is not the case with a perceptron.

## 2.3   The Optimised Dendritic Cell Algorithm

By rearranging the block diagram in Fig. 1 it is possible to make improvements to the performance of the DCA. The final comparison between *IL*-10 and *IL*-12 can be replaced by comparing the difference between the two signals with zero. As the two signals are both weighted sums of the same three input signals, the instantaneous difference between the two can be expressed as a single weighted sum. The new abstract signal is termed $K$. Finally, all calculations that require no persistent state can be calculated per population rather than per cell, significantly reducing the number of calculations required to implement the algorithm. The optimised algorithm is illustrated in Fig. 2. It is estimated that for a population of 100 cells this reduces the number of operations per iteration from 180 (3 multiplications and 3 additions for 3 signals per cell) to 12
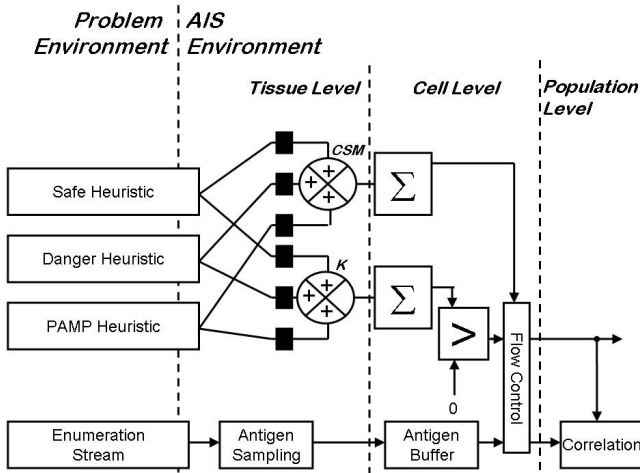
**Fig. 2.** A representation of the optimised DCA. More processing has been moved into the tissue and fewer calculations are required for the intermediate signal generation.

(3 multiplications and 3 additions for 2 signals per cell). This estimate is only based on arithmetic operations, not assignment operations.

## 3    Modelling the Dendritic Cell Algorithm as a Filter

It is hoped that by modelling the DCA as a filter it is possible to gain an insight into the workings of the algorithm that will make automated population tuning possible. In the optimised version of the DCA, the tissue now provides 2 input signals to the DC population, $K$ and $CSM$. These signals are both weighted sums of the input heuristics. $K$ represents the information used to make the decision and $CSM$ is a control signal which affects how long the cell will remain sampling $K$. The tolerance of the algorithm to noise, as discussed in [7], suggests that not all frequencies of the $K$ signal are processed by the DC population. In order to gain some insight into which frequencies are used and which frequencies are ignored, it is possible to reconstruct the $K$ signal from samples taken by the DC population. Comparing the magnitudes and frequencies of the reconstructed signal $\hat{K}$ and the input signal, $K$ allows a model to be produced of what information is passed through the cell. Note that this estimation of $\hat{K}$ is not a suggested extension or improvement to the algorithm, merely a tool to analyse the standard algorithm. In order to estimate $\hat{K}$ it is necessary to keep track of how long each cell samples for. By dividing the accumulated $K$ signal by the length of time each cell samples for it is possible to estimate $\hat{K}$ for a given cell. The full derivation of the model can be found in [8]. The final result, relating migration threshold $M_i$, the constant $CSM$ signal, $C$ and the frequency response of the cell is given by:

$$H(\omega) = \frac{\sum_{g=0}^{W_L-1} \sum_{b=0}^{W_L-1} e^{-jb((\omega+(2g\pi)))}}{W_L^2} \tag{1}$$

where $\omega$ is the frequency of the input signal, $j$ is the square root of $-1$ and $W_L$ is defined as:

$$W_L = \left\lceil \frac{M_i}{C} \right\rceil \tag{2}$$

### 3.1   Verification of the Model

To verify the model, the response generated by the model was compared to the output of a DC. To measure the frequency response of a DC, sine waves with an amplitude of 1 were presented as inputs, at varying frequencies. The maximum magnitude of the output was used as an approximation of the gain of the cell for each frequency. Figs. 3 and 4 show the results of these experiments. In every case the model is evaluated in the range from 0Hz to half of the sampling frequency of the system. The Nyquist frequency of any system is $\frac{1}{2}f_s$ where $f_s$ is the sampling rate. It is only necessary to examine the system within this range, as accuracy up to the Nyquist frequency guarantees the same level of accuracy for all frequencies. For details please see pages 41-43 of [5].
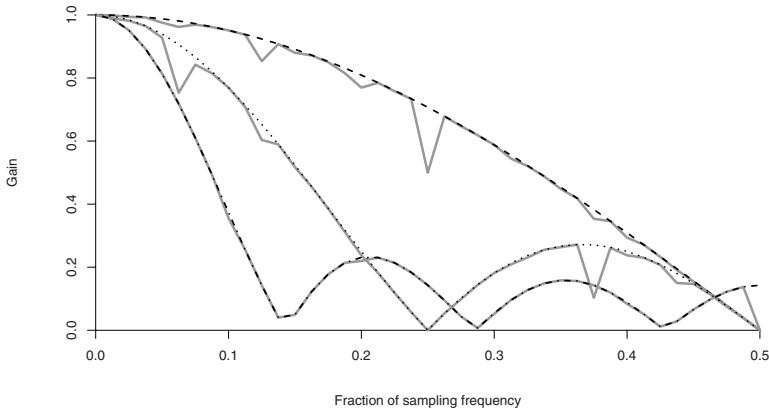


**Fig. 3.** The Effects of Varying the Migration Threshold. For these experiments the value of the *CSM* is held at 20 and the migration threshold is 30 (dashed line), 60 (dotted line) and 120 (dot-dashed line0). In each case the corresponding actual response is shown as a solid grey line. Data taken from [8].

The model predictions are reasonably accurate across the range of input parameters. Some transient drops in the algorithm's response are not predicted but the general shape of the response is well matched.
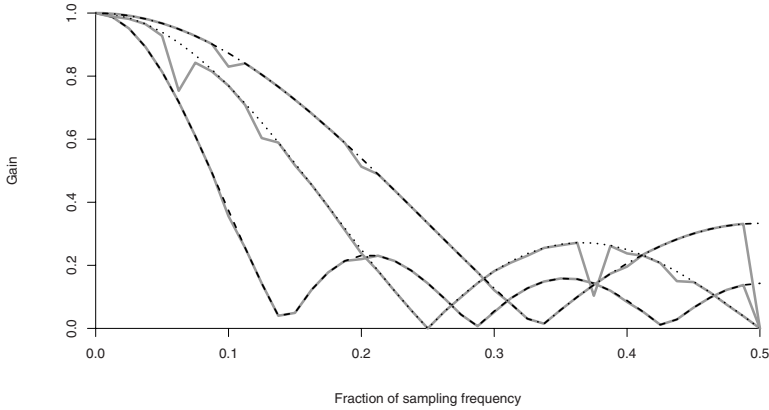
**Fig. 4.** The Effects of Varying the *CSM* Value. For these experiments the value of the migration threshold is set to 60 and the *CSM* value is 10, (dashed line) 20, (dotted line) and 30, (dot-dashed line). In each case the corresponding actual response is shown as a solid grey line. Data taken from [8].

## 4   Limitations of the Frequency Model

The assumptions made to derive the model limit how useful the results are for predicting the response of the DCA. Here we discuss the key assumptions and the effects that these assumptions have on the model.

### 4.1   Constant Co Stimulatory Molecule (*CSM*)

The model assumes that the *CSM* signal is kept constant over the lifetime of the cell. This is unlikely, as the *K* signal and the *CSM* signal are both weighted sums of the same three input signals, so whilst it is possible for one to move independently of the other, it is highly unlikely. However, it is doubtful that this is a factor in the model's accuracy. The *CSM* signal is accumulated by a cell over its lifetime. This means that any constant model of *CSM* is equivalent to any selection of the *CSM* signal with the same accumulated total over the lifetime of the cell. The implication of this is that the model allows the user to inspect the cell's behaviour for a small range of *CSM* values. Thus any tuning methodology based on this model would be valid if the selected *CSM* was a good representation of important regions of activity for the application.

### 4.2   Antigen Correlation

The model makes no attempt to take into account the antigen correlation of the algorithm, so it can make no predictions about how this element of the algorithm is effected by the input parameters. For applications where the correlation between antigen presentation and signal presentation is trivial this is unimportant.

For example, if there is no delay between the antigen being presented and its effects being felt, the results of the model would be adequate for representing the application's needs. However, when there is a delay between antigen presentation and the resulting signal presentation, or where the relationship between antigen presentation and signal presentation is combinatorial, (i.e. no one antigen is responsible for a positive decision, but certain combinations of antigen can cause this to happen) the model will not provide enough information to select the migration threshold range. It is of note that there are no applications of the DCA in the literature where combinatorial effects have been investigated. The model could be used in the future to investigate the effects of $M_i$ on cases where there is a time delay between antigen presentation and signal presentation, as the phase of the $\hat{K}$ signal will provide information about the lag introduced by the algorithm and thus, the largest possible time between sampling an antigen and ceasing to process signal.

### 4.3   Single-Cell Modelling

The model only considers a single cell operating in isolation from the rest of the population. This is considered to be the most significant drawback to the practical use of this model for migration threshold tuning. The DCA relies on the use of a population of cells to ensure that samples are processed frequently and to gather a wide range of data from multiple frequencies. By ignoring the interaction between a population of cells it is likely that the model is an oversimplification. For this reason it was decided to extend the model to incorporate multiple cells.

## 5   Extending the Frequency Model

In order to model multiple cells in the frequency domain, it is necessary to specify how they will interact during the normal operation of the algorithm. To produce a population-wide $\hat{K}$ we must find a reliable way of combining the data from a population of cells. For the purposes of this investigation it was decided to simply periodically sample the cell population and check for migrated cells. The $\hat{K}$ output from each migrated cell would be averaged together to produce a population-wide estimate of $K$ for that window. By averaging together the output from multiple cells, the process of generating a multi-cell model is made much easier. In the frequency domain, the averaged output from multiple filters can be modelled as simply the sum of the gains. The averaging process has no effect on the shape of the response, but scales it to be in the range 0-1. To explore the effects of this multi-cell model a 2 cell system was created using one cell with a migration threshold of 90 and one cell with a migration threshold of 110. The $CSM$ signal was held at 20 and the sampling rate was held at 1Hz. The output of the cells was checked every algorithm cycle. All of the experiments were carried out using the Octave environment. Fig. 5 shows the frequency responses for the single cell models using an $M_i$ of 90 and 110 respectively.
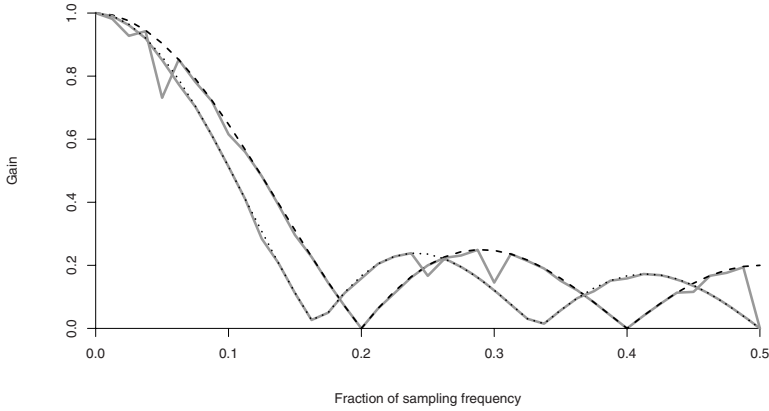
**Fig. 5.** The frequency responses of the single cell models for a migration threshold of 90 and 110. The dotted-line illustrates $M_i = 90$ and the dashed-line illustrates $M_i = 110$. In both cases the corresponding solid, grey line is the actual response.

## 6    Results and Discussion

Fig. 6 shows the frequency response of the actual system and the predicted output. The two lines clearly diverge more than the other models. The source of the difference is a combination of the asynchronous nature of the dendritic cell algorithm and the way in which the actual system gain is calculated. To calculate the gain of the actual system, the peak value of the output is recorded by the simulator. As the cells have different migration thresholds there will be occasions when one cell reports and the other does not. On other, rarer occasions,
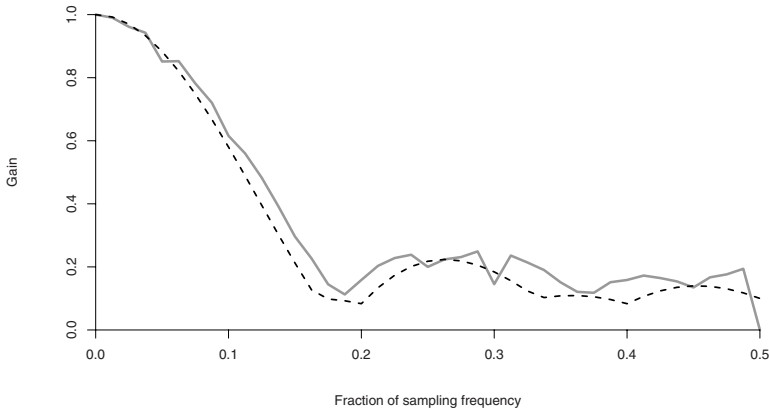


**Fig. 6.** The frequency response of the two cell system. The dashed line is the predicted response and the solid, grey line is the actual response.
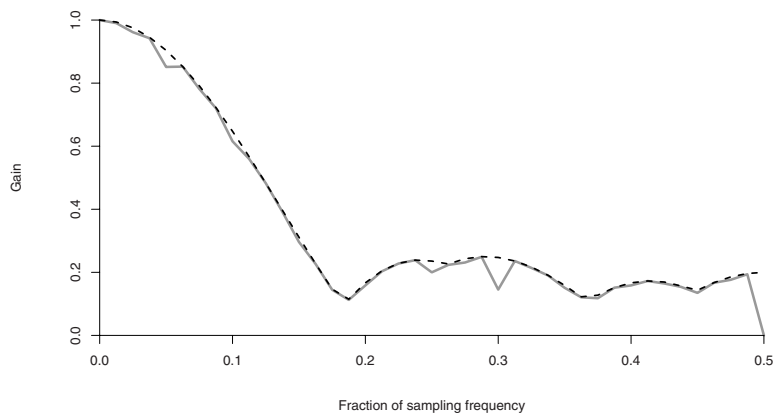
**Fig. 7.** The frequency response of the two cell system. The dashed line was generated by using the largest gain out of the two, single cell predictions for each frequency. The solid, grey line is the actual response.

both cells will synchronise and report at the same time. As the maximum peak is recorded as a measure of gain, the cell with the larger gain for that frequency will dominate the results from the simulator. This can be verified by comparing the measured response from the algorithm with the maximum of the two single-cell model predictions. In Fig. 7 the output from the actual system clearly follows
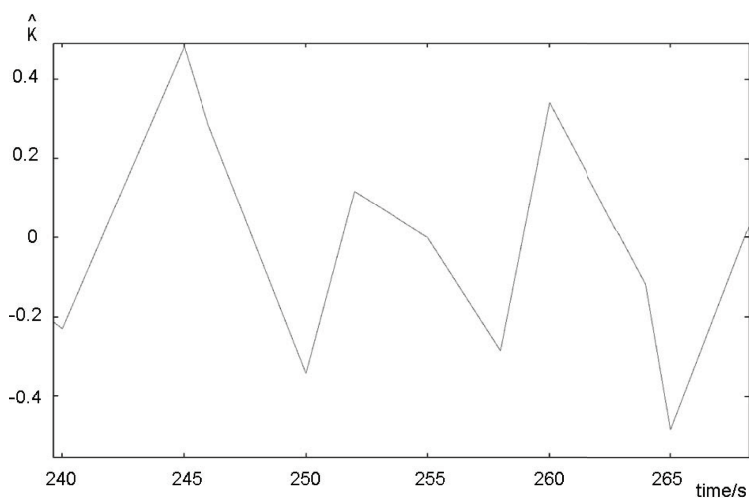


**Fig. 8.** An example of the output for a two-cell DCA. The sample frequency is $1Hz$ and the input frequency is a sine wave at $0.125Hz$ with a magnitude of 1. The first peak is the gain of the cell with a migration threshold of 90 (approximately 0.48), the second peak is the gain of the cell with a migration threshold of 110 (approximately 0.31) and the third peak is the average gain of each cell (approximately 0.40).

the maximum path of the two model predictions. Fig. 8 shows an example of the asynchronous system outputting three different sized gains for a single input frequency.

The construction of a model capable of predicting the response of a population of DCs is a non-trivial task. The asynchronous nature of the population means that the differing phases of the cells will have a significant effect on the output of the system. Effectively the relationship between gain and input frequency has ceased to be expressible using conventional means, as the gain for a given frequency is a range of values, depending on the relative phases of the cell population. For a two cell system there are four possible gains for each frequency, the gain of cell 1, the gain of cell 2, the average gain of cell 1 and cell 2 and a gain of zero, when neither cell migrates. It is possible to derive that the number of possible gains for a single input frequency, for a population of cells is given by:

$$N_g = 2^P \tag{3}$$

where $N_g$ is the number of possible gains and $P$ is the number of cells in the population. This is a worst-case that assumes that it is possible for all cells to simultaneously drift in and out of phase with one another. For a standard 100 cell implementation of the DCA this evaluates to approximately $1.27 \times 10^{30}$. Whilst it is possible to calculate the average response, it is questionable if this will be sufficient to provide enough information to effectively tune the system. It is possible that the cells drifting in and out of phase with one another adds another level of filtering to the system. A transient spike will be picked up by some, but not all of the cells migrating at a given interval, thus the average output over the population will potentially remove some of the noise from the inputs.

## 7   Conclusions and Future Work

These results cast doubt on the usefulness of traditional frequency-based techniques for modelling the DCA. An effective, multi-cell model, potentially needs to be able to take into account the differing phases of the cells, but even for standard implementations the space of possible gains is huge. The average response could be calculated with knowledge of how often combinations of cells drift in and out of phase with one another. This is calculable for a constant $CSM$ system by using the different values of $W_L$. Such a model would only be a guideline for the general case of the algorithm and the computational complexity of evaluating such a model could potentially outweigh the benefits of automated parameter tuning vs. the trial and error approach.

## Acknowledgements

# References

1. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing dendritic cells as a novel immune inspired algorithm for anomaly detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627. Springer, Heidelberg (2005)
2. Greensmith, J., Aickelin, U., Twycross, J.: Articulation and clarification of the dendritic cell algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163. Springer, Heidelberg (2006)
3. Greensmith, J., Twycross, J., Aickelin, U.: Dendritic cells for anomaly detection. Congress on Evolutionary Computation (CEC) (2006)
4. Greensmith, J.: The Dendritic Cell Algorithm. PhD Thesis. The University of Nottingham (2007)
5. Ifeachor, E.C., Jervis, B.W.: Digital Signal Processing: A Practical Approach. Prentice-Hall, Englewood Cliffs (2001)
6. Kim, J., Bentley, P.J., Wallenta, C., Ahmed, M., Hailes, S.: Danger is ubiquitous: Detecting mis- behaving nodes in sensor networks using the dendritic cell algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, Springer, Heidelberg (2006)
7. Oates, R., Greensmith, J., Aickelin, U., Garibaldi, J., Kendall, G.: The Application of the Dendritic Cell Algorithm to a Robotic Classifier. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 204–215. Springer, Heidelberg (2007)
8. Oates, R., Kendall, G., Garibaldi, J.M.: Frequency Analysis for Dendritic Cell Population Tuning. Evolutionary Intelligence 1(2), 145–157 (2008)