Parameter Optimization for LTE Handover using an Advanced SOM Algorithm

Neil Sinclair*, David Harle*, Ian A. Glover†, James Irvine* and Robert C. Atkinson*
*Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK
†Department of Engineering and Technology, University of Huddersfield, Huddersfield, UK
Email:{neil.sinclair, robert.atkinson}@strath.ac.uk

***Abstract*—A novel approach to enhance the robustness of handovers in LTE femtocells is presented. A modified Self Organizing Map is used to allow femtocells to learn about their specific indoor environment including the locations that have prompted handover requests. Optimized handover parameter values are then used that are specific to these locations. This approach reduces both the number of handover failures and the occurrence of ping-pong handovers. It also improves network efficiency by reducing the signaling overhead. The application of machine learning to this task complies with the plug-and-play functionality that is a requirement of Self Organizing Networks in LTE systems.**

I. Introduction

Due to the rapid proliferation of smartphones and increased data rates demanded by subscribers, Long Term Evolution (LTE) will use femtocells and picocells to meet future traffic requirements. The addition of so many base stations will require a more efficient network generally and more efficient handover management specifically. Self Organizing Networks (SON) will be used to operate and optimize the LTE network to realize increased network efficiency. Base stations within the network will be able to automatically configure their radio parameters with minimal human interaction using 3 key facets: self-configuration, self-optimization and self-healing. Self-optimization of handover is the particular focus of this paper.

Handover ensures that users remain connected to the network with a defined quality-of-service (QoS) as they move through the coverage area. Handover optimization within LTE is concerned with managing the conflicting requirements to minimize the likelihood of dropped calls whilst minimizing the number of unnecessary handovers. Two tuneable parameters can be used to govern handover performance. These parameters are Time-To-Trigger (*TTT*) and Handover Hysteresis value (*Hys*). Handover to a candidate base station can only be executed if that candidate provides better signal strength than the serving base station by an amount equal to, or exceeding, *Hys* for a duration equal to the *TTT*. Making these parameters specific to the location of the user, rather than adjusted on a cell-wide basis, would allow significant improvement in both the number of dropped calls and the number of unnecessary handovers. Finding the optimal values for *TTT* and *Hys* using an unsupervised neural network is the aim of the research presented here.

Much work has been completed in the area of handover optimization due to Mobility Robustness/Handover Optimization being one of the use cases of SON defined by Next Generation Mobile Networks (NGMN). Jansen *et al* [1] presented a successful parameter optimization algorithm that involved adjusting the parameters based on the resulting key performance indicators (Handover Failure ratio, Ping-pong handover ratio and call dropping ratio). Yang *et al* [2] conducted research on altering the handover call flow in femtocell to macrocell handover to improve handover performance. The novelty of the work presented here is that parameter optimization is completed on a location specific basis rather than for the entire base station; demonstrated in a realistic indoor scenario.

The problem under investigation can be best described by explaining two scenarios involving an active and mobile user moving around an indoor environment served by a femtocell. In the first scenario, as the mobile terminal approaches, and passes through, an external door it is likely to detect an increase in the Reference Signal Received Power (RSRP) from an externally located macrocell. As a consequence, a measurement report will be transmitted from the mobile terminal to the femtocell base station informing the femtocell that handover may be required. This geographical location resides within a Permissive Zone: an illustration of where handover should be able to take place due to the mobile user leaving the building. However, failed handovers can occur here (when a user leaves the serving area of the femtocell) if a handover mechanism is too conservative (i.e. the *TTT* and *Hys* values are too high). Such failed handovers are likely to lower the *Hys* and *TTT* parameters, making future handover decisions more aggressive. The second scenario is slightly different: an active mobile terminal approaches a large window (with low penetration loss). The increase in RSRP from the macrocell will cause a measurement report to be transmitted from the mobile terminal to the femtocell which may invoke a handover, as in the first scenario. However, as the mobile terminal continues to move past the window, the relatively high RSRP from the macrocell is likely to decline rapidly and thus trigger another measurement report from the mobile terminal to the macrocell, indicating that a better RSRP can be obtained from the femtocell. Such actions will invoke a second handover, in quick succession, from the macrocell back to the femtocell (ping-pong handover). Clearly, this geographical location resides within a Prohibition zone: an illustration of where unnecessary handovers can occur (i.e. the *TTT* and *Hys* values are too low). A handover parameter optimization algorithm is required to correctly balance the effect of these scenarios. To this end, a novel XSOM algorithm is used to optimize the handover parameters.

The remainder of this paper is structured as follows. Section II provides a brief overview of Autonomics and the tuning of the handover parameters, Section III provides a brief explanation of the advanced kernel Self Organizing Map used to alter the handover parameters, Section IV describes simulation results that verifies the utility of the algorithm, and, finally, Section V summarizes the paper and draws conclusions.

II. Self Organizing Networks

The term Autonomic Computing refers to computing systems having the ability to self manage and autonomically react to unpredictable events while hiding the complexities of the system to the end user. The inspiration for autonomic computing is the way in which the nervous system regulates the operations of biological organisms [3]. The autonomic paradigm is one in which time-consuming and error-prone tasks are undertaken by self managing components, leaving human administrators free to formulate high-level policies. Autonomic networks use the same paradigms created for autonomic computing systems but apply these ideas to network management [4].

The four stages involved in any autonomic system are Monitor, Analyze, Plan, Execute. The control loop starts by monitoring its environment and gathering information about the scenario in question. This information is then analyzed to determine if actions are required to optimize the performance of the system. If actions are required, the planning stage determines what these actions are. The Execute stage implements the actions in a technology specific manner. This is a useful structure for any self optimizing system as the inherent feedback creates an autonomically adjusting management system. This control loop constitutes the process of autonomic networks and, by extension, the process of SON.

The SON paradigm [5], developed by NGMN and 3GPP, applies autonomic networking to the mobile communications domain. Using SON allows the network to handle the large dimensions of the network, the large number of base stations (Femtocells and Macrocells) and the large number of complex network parameters. The plug-n-play functionality allows femtocells (HeNodeBs) to be deployed without the provision (by user or network provider) of information about the femtocell environment. This complicates the creation of an autonomous system because all radio environments are unique. Any algorithm used within femtocells for handover optimization must, therefore, be able to configure and optimize itself for the particular environment in which it operates.

Self-optimization is defined as the process whereby eNodeB and UE measurements are used to autonomically tune the radio access network to its specific environment to improve performance. Self-optimization can be applied to handover in both femtocells and macrocells as a constant and automated process with no human interaction. Efficient handover management is required to constantly support high quality voice and data traffic. Handover management is one of the use cases of the SON paradigm defined by the operators alliance NGMN and is used to optimize handover performance between neighboring base stations, including femtocells.

The tuning of handover parameters is a complex task due to the irregular coverage areas of base stations coupled with the effects of shadowing and multipath propagation which gives rise to stochastic variation in RSRP and signal quality. This results in terminals receiving a better RSRP from a neighboring base station at one instant and a worse RSRP the next due to movement of environmental scatterers, even for stationary terminals. Such changes in RSRP can trigger unnecessary and unwanted handovers adding stress to the network. The *TTT* and *Hys* parameters control the timing of handover triggers. The permissive *TTT* and *Hys* values are pre-defined in LTE networks [6]. There are 16 valid *TTT* values, i.e. 0s, 0.04s, 0.064s, 0.08s, 0.1s, 0.128s, 0.16s, 0.256s, 0.32s, 0.48s, 0.512s, 0.64s, 1.024s,1.280s, 2.560s and 5.120s. The *Hys* value varies in 0.5dB steps between 0 and 10dB. Handover-too-early and handover-too-late metrics [7] are defined to capture incorrect handover timing. Handover-too-early occurs when a triggered handover is unnecessary and handover-too-late occurs when a call is dropped. When a handover-too-early is detected, the *TTT* and *Hys* are likely to be increased to reduce the probability of it occurring again. This increases the probability of a call being dropped. If a handover-too-late is detected then the *TTT* and *Hys* are likely to be decreased to reduce the probability of future calls being dropped. This increases the likelihood of handover-too-early. An optimization algorithm must find the best statistical balance between these undesirable events.

In the work described here, the autonomic system will monitor when, and where, unnecessary handovers and dropped calls occur between the femtocell and macrocell and seek to reduce their numbers over time by adapting the handover parameters. The direction finding capability of MIMO systems is exploited to provide a profile of locations (i.e. regions in the radio environment) where handover is likely to occur. The kernel Self Organizing Map is used to continually map the locations where handovers have occurred and use this information to identify the permissive and prohibition zones. This work assumes that the femtocell layer is largely unaffected by macrocellular interference, i.e. some form of interference management is in operation [8], [9].

III. Improved Kernel Self Organizing Feature Map using X-means

The purpose of the autonomic managed element, to be included within SON, is to optimize the handover process based on the application of a modified version of the Self Organizing (Feature) Map (SOM) [10], called XSOM. This novel approach uses a SOM with kernel methods and X-means [11] to learn the environment and optimize the parameters. The Monitor phase of the SON algorithm is location detection of the user when a handover measurement report has been triggered. The Analyze phase is based on a kernel SOM and allows the femtocell to learn the locations of the propagation environment that correspond to both permissive and prohibition zones. Next, the Plan phase takes this information and decides on an appropriate response; i.e. to increase or decrease the handover parameters. Finally, the Execute phase translates the decision from the Plan phase into LTE specific commands. The Kernel SOM algorithm is used within the Analyze and Plan phases.

A SOM is an unsupervised neural network that uses group learning to produce a low-dimensional output space from a high-dimensional, discretized, input space. The kernel SOM algorithm [12] non-linearly transforms the data into a feature space which results in additional detail (accuracy) at the point of interest and reduces the vector quantization error that is inherent to SOM. This transformation allows for the distances between the weights and the inputs to be calculated nonlinearly. The kernel SOM is particularly useful for detecting clusters within data and in this work it is used to perform location fingerprinting based on RSRP and angle of arrival. There are four phases which describe the learning process of the kernel SOM: initialization, competition, cooperation, and synaptic adaptation. This algorithm has been augmented with a fifth stage, X-means. X-means allows for the area of the SOM to be handled as a series of Voronoi cells.

A. Kernel SOM: Initialization

Initialization of the kernel SOM network presets the individual weight values of each neuron in the lattice to values drawn from a uniform distribution. The initial weight values in this work are distributed across the propagation region of the femtocell.

B. Kernel SOM: Competition

In kernel SOMs, all neurons are connected to all inputs and (unlike other neural networks) the neurons have no activation function. In the competition phase, inputs are applied to the system. This will occur every time a mobile terminal generates a measurement report (triggered by detection of a base station other than the serving base station). The representation for an a-dimensional input is defined in Equation (1) and the weight vector associated with each neuron in the lattice ($\mathbf{w}_j$ where j is the neuron index) has the same structure and dimensions as the input.

$$\mathbf{x} = [x_1, x_2, ..., x_a]^T,\ \mathbf{x} \in \mathbb{R}^a \tag{1}$$

Given that there is no activation function, the output of each neuron will be a combination of both the input and weight vectors. From a geometrical perspective, the winning neuron in the kernel SOM is calculated based on Euclidean distance after implicitly transferring the map into a non-linear feature space using the kernel trick [13]. The kernel trick is used because the non-linear mapping of the input and weights allows for more detail at the point of interest which reduces the vector quantization error of the system. The mapping of $\mathbf{x}$ to $\phi(\mathbf{x})$ can be implicitly accomplished with no knowledge of ϕ. The mapping to the feature space is completed using a kernel such that $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ where $\phi(\mathbf{x})$ is the function that maps the data onto the feature space, as shown in Equation (2). The selected (i.e. winning) neuron is that whose weight vector provides the best match to the input vector. This is the competitive aspect of the algorithm. The index of the winning neuron, $i(\mathbf{x})$ within the lattice $\mathcal{L}$ (denoting the grid of neurons in the weight space) is given by Equation (3).

$$\begin{aligned} \|\mathbf{x} - \mathbf{w}_j\|^2 &= \|\phi(\mathbf{x}) - \phi(\mathbf{w}_j)\|^2 \\ &= K(\mathbf{x}, \mathbf{x}) + K(\mathbf{w}_j, \mathbf{w}_j) - 2K(\mathbf{x}, \mathbf{w}_j) \end{aligned} \tag{2}$$

$$K(\mathbf{x}, \mathbf{w}_j) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{w}_j\|}{2\sigma^2}\right) \tag{3}$$

C. X-means

X-means [11] is an advanced method of cluster analysis that allows for a Voronoi cell diagram to be automatically generated based on the locations of the weights within the system. This algorithm is based on the k-means algorithm but is supplemented by autonomically calculating how many clusters are required for the set of weights within the SOM (in this case the number of doors and windows). Within X-means, there is a requirement to know the range, $k_{min} \leq k \leq k_{max}$, that the number of clusters, k, will fall within. This range can have a default set of values for every femtocell. The inclusion of X-means within the kernel SOM results in faster convergence times as a consequence of a reduction in the level of false learning within the system due to only updating the weights within the network that are within the region of the input. The algorithm is structured as follows:

1) The partitioning is completed by, initially, allocating k_{min} centroids randomly within the area of the network.

2) Each weight can then be allocated to its nearest centroid using Equation (4) where m denotes the centroid, c the index and $q(\mathbf{w})$ the index of the winning centroid. This results in the generation of Voronoi cells.

$$q(\mathbf{w}) = \arg\max_c \|\mathbf{w}_j - \mathbf{m}_c\|,\ j \in \mathcal{L},\ c \in [0, k] \tag{4}$$

3) After each weight has been allocated to its corresponding centroid, the centroid must be updated using Equation (5). Each new centroid location ($\mathbf{m}_c$) is the mean value of all the allocated weights. n is the number of weights allocated to mean c.

$$\mathbf{m}_c = \frac{1}{n}\sum_{j=1}^{n} \mathbf{w}_j \tag{5}$$

4) Steps 2 and 3 are repeated until convergence of the centroid and allocated weights have been achieved.

5) The number of centroids can now be updated. The algorithm works by initially using k_{min} centroids, splits each of the centroids into two when required and finishes using any value within the range, that best fits the data. Determining whether this split is valid is facilitated by the Bayesian Information Criterion (BIC). BIC scoring operates by using a posteriori probabilities to score the models. To approximate these probabilities Equation (6) is used.

$$BIC(M_s) = \hat{l}_s(D) - \frac{p_s}{2} \cdot \log R \tag{6}$$

Here, $\hat{l}_s(D)$ is the log-likelihood of the data taken at the maximum likelihood point, p_s is the number of parameters in M_s and R is the number of weights in data set D. The maximum likelihood estimate for the variance is calculated using Equation (7).

$$\hat{\sigma}^2 = \frac{1}{R-k} \sum_i (\mathbf{x}_i - \mathbf{m}_{q(\mathbf{w})})^2 \tag{7}$$

where k is the current number of centroids being used in the X-means algorithm and i is the input index. The log-likelihood of the data points that belong to centroid m_c ($\hat{l}_s(D_c)$) and including the maximum likelihood estimates, yields Equation (8).

$$\begin{aligned}\hat{l}_s(D_c) = &-\frac{R_c}{2}\log(2\pi) - \frac{R_c \cdot M}{2}\log(\hat{\sigma}^2) \\ &-\frac{R_c - k}{2} + R_c \log R_c - R_c \log R\end{aligned} \tag{8}$$

Within this equation, R_c is the number of weights allocated to m_c. The number of parameters p_s is shown in Equation (9).

$$p_s = (k-1) + (M \cdot k) + k \tag{9}$$

The number of clusters, k, is increased based on the resultant BIC score until either the solution has converged or the condition $k_{min} \leq k \leq k_{max}$ no longer holds. Convergence is checked by comparing the BIC score of the final network to the BIC score of the initial solution.

D. Kernel SOM: Cooperation

Once the winner for a given input vector has been selected and the weight has been assigned to its closest centroid, the weights of the neurons within the winner's sphere of influence are updated if they are linked to the same centroid as the winner.

The sphere of influence is governed by a neighborhood function which determines how many of the winner's neighbors can undergo learning, and also the degree to which they will learn. The neighborhood function should decay monotonically with distance from the winner. Furthermore, it should be maximum at the winner ($d_{f,e} = 0$) and tend to zero as $d_{f,e} \to \infty$ where e and f are neurons in the lattice. A popular choice for the neighborhood function which satisfies these requirements is the Gaussian function, as shown in Equation (10). This is the function adopted in this work.

$$h_{f,e}(n) = \exp\left(-\frac{d_{f,e}^2}{2\sigma^2(n)}\right),\ j \in \mathcal{L} \tag{10}$$

The parameter σ defines the width of the Gaussian function and in essence σ determines the size of the sphere of influence around the winning neuron.

E. Kernel SOM: Synaptic Adaptation

The adaption process is concerned with the execution of the weight update procedure for all neurons within the sphere of influence of the winner (governed by the neighborhood function). This involves utilizing not only the sphere of influence but a learning rate too. Parameter η is the learning rate. In practice, the learning rate also decays with time (or iterations) and is an exponentially decreasing function as shown in Equation (11):

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right) \tag{11}$$

η_0 is the initial value and τ_2 is a second time constant. The augmented Hebbian weight update equation can be written as shown in Equation (12).

$$\Delta\mathbf{w}_j(n) = \eta(n) h_{j,i(\mathbf{x})}\left(\mathbf{x}(n) - \mathbf{w}_j(n)\right) \tag{12}$$

Thus, the weights for neuron j within the sphere of influence of the winner are updated iteratively according to the rule given by Equation (13).

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n) h_{j,i(\mathbf{x})}(n)\left(\mathbf{x}(n) - \mathbf{w}_j(n)\right) \tag{13}$$

As the neurons are updated their locations will eventually converge; the learning rate having become low and the neighborhood no longer updating nodes other than the winning node. The locations of both permissive and prohibition zones have then been identified.

IV. Simulation Model and Results

The proposed algorithm has been implemented using Network Simulator 3 (NS3). When a handover trigger has been detected (Monitor phase), the XSOM algorithm allows the femtocell to learn where handover can be triggered in the environment (Analyze phase). A decision on how to tune the parameters can then take place in the Plan phase and the action implemented in the Execute phase of the autonomous system. The scenario described here consists of one permissive region and one prohibitive region but the algorithm is also effective in more complex environments. The parameters of the simulation used to demonstrate the effectiveness of the parameter optimization algorithm are summarized in Table I.

A random walk mobility model allows for a random change of direction of the user after a prescribed period of time (or distance travelled). Within the proposed scenario, the mobility of the user is modeled by a random walk that has been modified to ensure that (i) the user leaves the room when a prohibition zone is entered (i.e. the user walks through the door) and that (ii) the users velocity remains constant whilst in a permissive zone. A single-slope propagation model has been used to determine RSRP. The RSRP of a femtocell

TABLE I
SIMULATION DETAILS

Parameter	Value
Simulation dimensions	7m × 9m
Room dimensions	7m × 7m
Exit area	2m × 7m
No. of mobile terminals	1
Direction change time	1.0s
Movement speed	1 - 3m/s
Initial position	center
Mobility model	random walk
Propagation model	single-slope
Error	0m
Initial *TTT*	320ms
Initial *Hys*teresis	5dB

and a macrocell are used to determine whether there is a requirement for handover. As the modeled system operates using an event-based paradigm focusing on handover instances and takes a generic approach, the choice of propagation and mobility models has only a secondary effect upon the results. The choice of such models does not affect the generality of the results. Since indoor radio environments are inherently complex (due to scatter from clutter) the effect of a position estimation error (of up to 3m) has been investigated. This error does not significantly affect the accuracy of the system. The error incurred for the neuron due to incorrect position detection, in effect, is nullified due to the uniform distribution of the error, i.e. zero mean.

When a handover trigger is detected (i.e. when the mobile terminal has detected another base station with a stronger RSRP (by a *Hys* value) for a prescribed period of time (*TTT*)), the location of the user (as perceived by the femtocell) is input to the algorithm. In this modified LTE system, the *TTT* and *Hys* values used are specific to the location of the user and are optimized as the system learns the success or failure of handover in this region. The values for the *Hys* and *TTT* will be different for permissive and prohibition zones. Specifically, the system detects the regions within the radio environment where unnecessary handovers or failed handovers occur and seeks to reduce these to an optimum level over time.

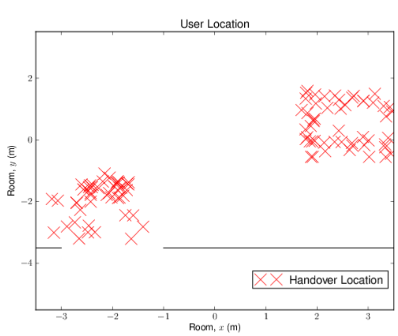

Fig. 1. Handover Locations

The location of the user at the point of a handover trigger is detected by the Monitor stage of the autonomic system. A snapshot of the locations of 100 handover triggers is shown in Figure 1. As can be seen there are 2 clusters of neurons: a single prohibition zone on the right and a single permissive zone on the left. Once the location of the mobile user has been detected, the Analysis phase examines the data and decides on possible actions that can be taken: increase the *TTT* and *Hys* parameters; decrease the *TTT* and *Hys* parameters; or leave the parameters unchanged. The Plan phase uses the data and the possible actions to decide on an appropriate process that will be used to optimize the handover scenario.

At initialization, the handover parameters for all the nodes within the neural network are set to default values. Handover then operates as normal, detecting where handover is unnecessary or where handover has failed. This information is used to optimize the *TTT* and *Hys* values for each weight within the network. The permitted values for *TTT* and *Hys* are defined by 3GPP (Section II). When an unnecessary handover is detected *TTT* and *Hys* are both increased to their next higher permitted values. When a handover failure has been detected, *TTT* and *Hys* are both decreased to their next lower permitted values. The handover triggers (either macrocell to femtocell or femtocell to macrocell) occur in the regions where the macrocell RSRP is greater than the femtocell RSRP. Figures 2 and 3 show the values of *TTT* and *Hys*, respectively, for the weights within the network after 500 handovers. The weights with values above the dashed line are located in a prohibition zone and the weights with values below the dashed line are located in a permissive zone. Due to the tuning of parameters the number of handovers that take place have been optimized.

In order to demonstrate that the algorithm is an improvement on the basic LTE network, the Handover Performance Indicators (HPIs) are evaluated. In this case, the HPIs are the ping-pong handover ratio (HPI_{pp}) and the handover failure ratio (HPI_{fail}) are of prime importance and are calculated by dividing the number of handover ping-pong/failed handovers by the total number of handovers.

Ideally, the number of failed handovers and handover ping-pong occurrences would be zero (and hence so would HPI_{pp} and HPI_{fail}). Unfortunately, practical systems are not ideal and do not operate optimally. In a practical system handover ping-pong and handover failure will occur within the network but the number of such occurrences can be reduced to minimize their effect. HPI_{pp} and N_{Hfail} with and without the proposed optimization algorithm are shown in Figures 4 and 5 for the simulated scenario. These figures have been generated using 30 parallel simulation runs to provide an ensemble mean. The number of handover ping-pongs and failed handovers are both lower for the optimized system showing that the proposed algorithm is successful in optimizing the handover parameters. The algorithm requires no prior information regarding the location of the windows or doors, etc. This knowledge (or, at least, the equivalent radio environment knowledge) is gained via unsupervised learning.

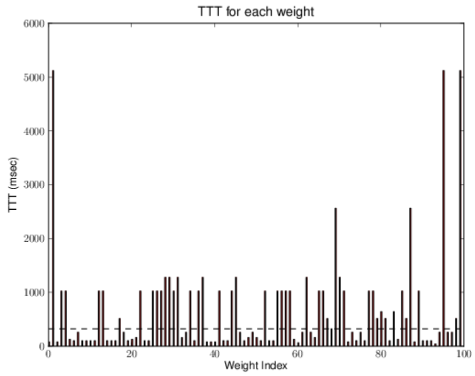

Fig. 2. *TTT* for each weight

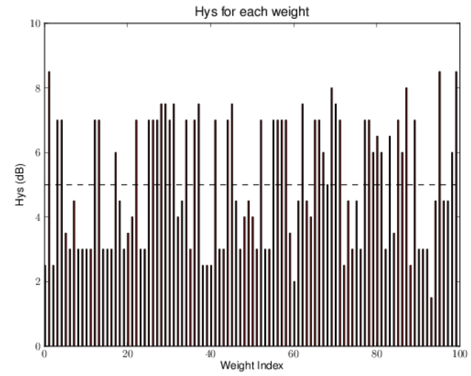

Fig. 3. *Hys* for each weight

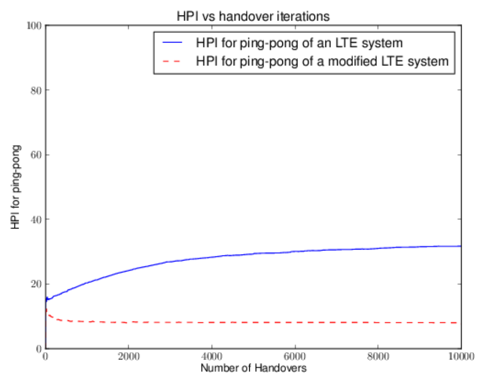

Fig. 4. Handover Performance Indicator: Ping-pong Handover

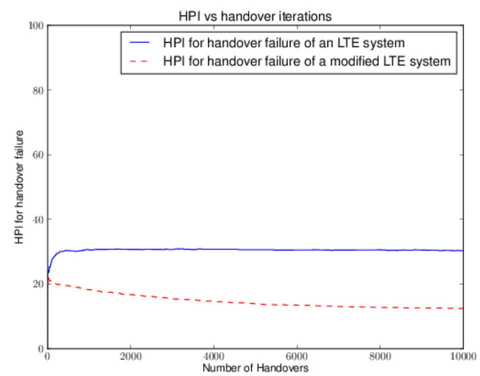

Fig. 5. Handover Performance Indicator: Failed Handovers

V. Summary and Conclusions

In this paper, a novel kernel SOM algorithm has been used to improve the efficiency of handover within an indoor environment. The algorithm has been shown to effectively optimize both *TTT* and *Hys* values to reduce the number of handover-too-early and handover-too-late events. The handover parameters are optimized on the basis of radio environment location (related closely to physical location). The use of the kernel SOM allows the parameters being used in a handover-permissive zone to be different from those being used in a handover-prohibition zone. One of the main advantages of using this algorithm within SON is that it becomes more flexible with regards to the femtocell being able to autonomically adapt to its environment and thus improve handover efficiency in a fast and efficient manner.

Acknowledgements

The authors would like to thank Dan Pelleg and Andrew Moore from the School of Computer Science, Carnegie Mellon University, Pittsburgh, USA for their version of the X-means code and their help with implementation issues.

References

[1] T. Jansen, I. Balan, J. Turk, I. Moerman, and T. K andrner, "Handover parameter optimization in lte self-organizing networks," in *VTC Fall, IEEE 72nd*, sept. 2010, pp. 1 –5.

[2] G. Yang, X. Wang, and X. Chen, "Handover control for lte femtocell networks," in *ICECC,*, sept. 2011, pp. 2670 –2673.

[3] P. Horn, "Autonomic computing: IBM's perspective on the state of information technology," IBM corporation, Technical Report, 2001.

[4] J. Strassner, *Cognitive Networks: Towards Self-Aware Networks*, 1st ed. Wiley, 2007, ch. 2: The Role of Autonomic Networking in Cognitive Networks, pp. 23 – 52.

[5] G. T. . V8.0.0, "Technical Specification Group Services and System Aspects; Telecommunications Management; Self Organizing Networks; Concepts and requirements (Release 8)," December 2008.

[6] G. T. . V9.3, "Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (Release 9)," June 2010.

[7] E. T. . V9.0.0, "Digital cellular telecommunications system (Phase 2+); UMTS; Telecommunications Management; SON; Policy NRM; IRP; Requirements," April 2010.

[8] P. Lee, T. Lee, J. Jeong, and J. Shin, "Interference management in lte femtocell systems using fractional frequency reuse," in *ICACT, 2010 The 12th Conference on*, vol. 2, feb. 2010, pp. 1047 –1051.

[9] C. Bouras, G. Kavourgias, V. Kokkinos, and A. Papazois, "Interference management in lte femtocell systems using an adaptive frequency reuse scheme," in *WTS, 2012*, april 2012, pp. 1 –7.

[10] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.

[11] A. M. Dan Pelleg, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 2000, pp. 727–734.

[12] D. MacDonald and C. Fyfe, "The kernel self-organising map," in *KES, Fourth International Conference on*, 2000.

[13] B. Schlkopf, "The kernel trick for distances," in *TR MSR 2000-51, Microsoft Research*, 1993, pp. 5–3.