Inferential estimation of high frequency LNA gain performance using machine learning techniques

Peter C. Hung, Seán F. McLoone, Magdalena Sánchez, Ronan Farrell
Institute of Microelectronics and Wireless Systems, Department of Electronic Engineering,
National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland
{phung, sean.mcloone, msanchez, rfarrell}@eeng.nuim.ie

Abstract

Functional testing of radio frequency integrated circuits is a challenging task and one that is becoming an increasingly expensive aspect of circuit manufacture. Due to the difficulties with bringing high frequency signals off-chip, current automated test equipment (ATE) technologies are approaching the limits of their operating capabilities as circuits are pushed to operate at higher and higher frequencies. This paper explores the possibility of extending the operating range of existing ATEs by using machine learning techniques to infer high frequency circuit performance from more accessible lower frequency and DC measurements. Results from a simulation study conducted on a low noise amplifier (LNA) circuit operating at 2.4 GHz demonstrate that the proposed approach has the potential to substantially increase the operating bandwidth of ATE.

1. Introduction

Reliable high frequency testing of radio frequency integrated circuits (RFIC) has become a significant factor in the cost and time-to-market of novel wireless products [1]. Testing of RFICs is generally performed using dedicated automatic test equipment (ATE) that record a set of measurements with the circuits operating in their functional mode so that performance can be compared against design specifications. This traditional testing solution is becoming prohibitively expensive as RFIC designs push into multi-gigahertz operating frequencies due to the increased cost of ATE at these frequencies and the difficulty with bringing multi-gigahertz RF measurements off-chip in a factory environment. Indeed, it is predicted that this later constraint may become a limiting factor in extending wireless technologies to even higher frequencies.

Manufacturers are therefore increasingly looking at alternative approaches to functional testing which avoid taking measurements at very high frequencies. Significant advances have been made in a variety of application domains in recent years through the development of Design for Testability (DfT) and Built-In-Test (BIT) methodologies [2-5]. In BIT, for example, on-chip testing circuitry allows evaluation of ICs using lower frequency or DC external testers. However, these methodologies come with significant overheads in terms of area and power consumption and are not readily applicable to RF mixed signal circuits.

In [6] the authors propose a different strategy. It is hypothesised that for many RFICs knowledge of responses at DC and lower frequencies may provide sufficient information to allow inference of performance at higher frequencies. Training of such inferential performance estimators can be achieved using data generated from detailed circuit simulations covering the space of process-dependent parameter variations [6]. When successful, this approach effectively extends the operating range of existing ATEs, allowing them to be employed to test next generation devices.

In this paper a case study is presented on the application of the proposed test methodology to classification of the gain performance of a low noise amplifier (LNA), a key component in modern telecommunication systems. Inferential classification of the amplifier gain at its operating frequency is performed using feature vectors consisting of DC measurements (quiescent currents and voltages) and gain measurements at other more accessible (lower) frequencies.

Since RF circuit interactions are inherently nonlinear and complex, it is anticipated that machine learning modelling and classification will generally be required to adequately capture the relationships between variables. This is demonstrated in the case

study through comparative results for linear and nonlinear classifiers. In each case, both direct and indirect classifier paradigms are considered. In the latter, linear and neural network models are trained to predict the value of the amplifier gain and a threshold rule applied to the resulting prediction to perform the circuit classification, while in the former, linear discriminant analysis, support vector machine and *k*-nearest neighbour classifiers are trained to directly classify the circuit performance on the basis of the feature vector provided.

The remainder of the paper is structured as follows. The case study and test methodology are described in Section 2 while the classical and machine learning inferential classifiers considered are briefly presented in Section 3. The results of a comprehensive Monte Carlo simulation study are presented in Section 4 together with a discussion of the implications for the proposed test methodology. Finally, the conclusions are given in Section 5.

2. Case Study and Test Methodology

In this study, the low noise amplifier (LNA) used was a standard low-voltage 2.4 GHz MOSFET design, simulated in ADS® using UMC's 0.18 µm silicon process technology [7]. The LNA circuit consisted of 2 bias transistors, 4 RF transistors (0.18 µm channel length and 0.5 µm channel width), 4 resistors, 3 capacitors and 4 inductors. The gain of the LNA at its operating frequency (2.4 GHz), $g_{2.4}$, was considered to be the critical performance parameter and was deemed to be within specification if 14.7 dB < $g_{2.4}$ < 17.2 dB.

To simulate LNA manufacturing process variations zero mean uniform random variations were introduced into the 38 most significant model parameters, as determined by a sensitivity analysis [6]. While in practice circuit parameters might be expected to vary normally around their nominal values, uniform distributions were chosen to give an even coverage of the LNA parameter space. Catastrophic failures such as short-circuits were excluded as these faults can be detected relatively easily using existing IC testing techniques.

Using this approach, 10,000 circuit simulations were performed. For each circuit the amplifier gain, g_f was recorded at sampling frequencies, f, of 0.1, 0.3, 0.6, 1.2, 1.4, 1.7 and 2.0 GHz and also at the operating frequency (2.4 GHz). In addition, 8 key DC bias voltages and currents were recorded. This data was then normalised to have zero mean and unit variance and divided into training and test data sets, each containing 5,000 samples.

The following notation is used to distinguish between the different feature vectors considered in the study.

$\mathbf{x}_f$	=	vector of the amplifier gains at all sampling frequencies from 0.1 to f GHz
$\mathbf{x}_{DC}$	=	vector of the amplifier DC bias voltage and current measurements
$\mathbf{x}_{f+DC}$	=	$[\mathbf{x}_f, \mathbf{x}_{DC}]$, a concatenation of $\mathbf{x}_f$ and $\mathbf{x}_{DC}$

The goal of testing is to classify amplifiers as being within specification (*'good'*) or not within specification (*'bad'*). In the proposed test methodology this is achieved without direct measurement of $g_{2.4}$ by inferring the classification from other circuit measurements, specifically $\mathbf{x}_{DC}$ and/or $\mathbf{x}_f$, with f ideally much less than 2.4 GHz. This is achieved by building a classifier, h, to map from a circuit feature vector, $\mathbf{x}$, to the appropriate classification

$$h(\mathbf{x}) \rightarrow z_{2.4}(g_{2.4}) = \{\pm 1\}. \quad (1)$$

Here, $z_{2.4}$ is a threshold function defined as

$$z_{2.4}(x) = \begin{cases} +1 & \text{if } 14.7 < x < 17.2 \\ -1 & \text{otherwise} \end{cases} \quad (2)$$

The mapping $h(\mathbf{x})$ is determined through training on simulated data generated using circuit simulation software such as ADS®, where $g_{2.4}$ measurements and target circuit classifications can easily be generated. Once trained, the classifier can then be used to test production circuits, with only the measurements needed to generate the feature vector, $\mathbf{x}$, required for testing. In the case study the 5,000 sample test data set is used to represent production circuits.

3. LNA Inferential Classifiers

The inferential LNA classifier can be implemented directly as in (1) using either linear classifiers such as linear discriminant analysis (LDA) or machine learning classifiers such as support vector machines (SVM) and *k*-nearest neighbour (*k*NN) classifiers. Alternatively it can be estimated indirectly by first learning a model that predicts the value of $g_{2.4}$ from the feature vector,

$$g(\mathbf{x}) \rightarrow \hat{g}_{2.4} \quad (3)$$

and then using a threshold function to perform the classification, that is

$$h(\mathbf{x}) \equiv z_{2.4}(g(\mathbf{x})). \quad (4)$$

A simple linear model or an artificial neural network such as a Multilayer Perceptron can be used to learn the mapping defined by (3). A brief overview of each of these models and classifiers will now be given.

3.1. Linear Model (LM)

A linear model (LM) for predicting $g_{2.4}$ can be constructed as

$$\hat{g}_{2.4} = g_{\mathrm{LM}}(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}, \quad (5)$$

where $\mathbf{w}$, the model parameters, are estimated from training data using least squares optimisation. Defining the matrix of training feature vectors as

$$\mathbf{X} = [\mathbf{x}_1 \;\; \mathbf{x}_2 \cdots \mathbf{x}_n], \quad (6)$$

where n is the number of training samples, and the corresponding vector of targets as

$$\mathbf{y} = [g_{2.4(1)} \;\; g_{2.4(2)} \cdots g_{2.4(n)}]^{\mathrm{T}}, \quad (7)$$

the least squares solution is given by

$$\mathbf{w} = [\mathbf{X}^{\mathrm{T}}\mathbf{X}]^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \mathbf{X}^{+}\mathbf{y}, \quad (8)$$

where X^{+} is the Moore-Penrose pseudo-inverse of $\mathbf{X}$.

3.2. Multilayer Perceptron (MLP)

The multilayer perceptron (MLP) is one of the best known and most widely used neural network architectures because of its universal function approximation capabilities, good generalisation properties and the availability of robust efficient training algorithms [8]. In this application a single hidden layer MLP is used to capture a nonlinear model between the LNA feature vector, $\mathbf{x}$, and gain, $g_{2.4}$:

$$\hat{g}_{2.4} = g_{\mathrm{MLP}}(\mathbf{x}), \quad (9)$$

where

$$g_{\mathrm{MLP}}(\mathbf{x}) = b^h + \sum_{i=1}^{M} \frac{w_i^h}{1+\exp(\mathbf{w}_i^u \cdot \mathbf{x} + b_i^u)}. \quad (10)$$

Parameters w_i^h, $\mathbf{w}_i^u$, $b_i^u, (i = 1, 2, \cdots, M)$ and b^h are weights and biases which collectively form the network weights vector, $\mathbf{w}$, and M is the number of hidden layer neurons. Defining a Mean Squared Error (MSE) cost function over the training data,

$$E(\mathbf{w}) = \frac{1}{n}\sum_{p=1}^{n}(g_{MLP}(\mathbf{x}_p, \mathbf{w}) - g_{2.4(p)})^2, \quad (11)$$

the optimum weights can be determined using gradient based optimisation techniques. Here, training was performed using the hybrid BFGS training algorithm with stopped minimisation used to prevent over-fitting [9]. The optimum number of neurons (M) was determined for each model by cross-validation on the test data set.

3.3. Linear Discriminant Analysis (LDA)

In linear discriminant analysis (LDA) the feature vectors are projected onto directions that have maximum discriminatory power, P, as measured by the ratio of between and within class variances. Here, the classical two-class Fisher's LDA [10] is used, in which the feature data is projected onto a single discriminatory direction

$$d = \mathbf{w}^{\mathrm{T}}\mathbf{x} \quad (12)$$

and $\mathbf{w}$ is chosen to maximise

$$P(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}. \quad (13)$$

The between-class and within-class scatter matrices ($\mathbf{S}_{\mathrm{B}}$ and $\mathbf{S}_{\mathrm{W}}$), defined as

$$\mathbf{S}_{\mathrm{B}} = (\bar{\mathbf{x}}_{\mathrm{G}} - \bar{\mathbf{x}}_{\mathrm{B}})(\bar{\mathbf{x}}_{\mathrm{G}} - \bar{\mathbf{x}}_{\mathrm{B}})^{\mathrm{T}} \quad (14)$$

and

$$\mathbf{S}_{\mathrm{W}} = \sum_{j \in G,B} \sum_{i \in C_j} (\mathbf{x}_i - \bar{\mathbf{x}}_j)(\mathbf{x}_i - \bar{\mathbf{x}}_j)^{\mathrm{T}}, \quad (15)$$

are computed over the training data. In (14) and (15) vector $\bar{\mathbf{x}}_j$ is the mean of the feature vectors in the jth class, C_j. $P(\mathbf{w})$, which is in the form of a generalized Rayleigh quotient, is maximised when $\mathbf{w}$ is chosen as the eigenvector corresponding to the largest eigenvalue of the generalised eigenvector problem

$$\mathbf{S}_{\mathrm{B}}\mathbf{w} = \lambda \mathbf{S}_{\mathrm{W}}\mathbf{w}. \quad (16)$$

Having determined d, the optimum projection of the feature vector, circuit classification is performed by applying a threshold function to d, that is:

$$h_{\mathrm{LDA}}(\mathbf{x}) = z_{\alpha}(d) = z_{\alpha}(\mathbf{w}^{\mathrm{T}}\mathbf{x}), \quad (17)$$

where

$$z_{\alpha}(x) = \begin{cases} +1 & \text{if } x \geq \alpha \\ -1 & \text{if } x < \alpha \end{cases}. \quad (18)$$

The value of the threshold, α, is chosen to minimise the misclassification rate (MCR) over the training data.

Linear discriminant analysis assumes classes are approximately normally distributed. However, the performance specification for the amplifier gain in the case study is such that the class of out-of-spec (‘*bad*’) circuits is bi-modal. To take this into account, circuit classification is implemented as a three-class problem with classes defined as bad-lower (BL), good (G) and bad-upper (BU). Classification is then performed in two-stages. Firstly, two *one-versus-all* classifiers are trained, one to classify LNAs as either ‘BL’ or ‘not-BL’, and the other to classify them as either ‘BU’ or ‘not-BU’. The overall classification is then obtained by taking a logical AND of the output of these classifiers, where ‘not-BL’ and ‘not-BU’ are considered to be ‘*good*’ and all other combinations ‘*bad*’.

3.4. Support Vector Machines (SVM)

Support Vector machines (SVM), proposed by Vladimir Vapnik in 1963, classify data sets by determining the margin which achieves the maximum separation between classes [11]. Consider a separating hyperplane, with orientation vector **w** and location parameter b, that divides two classes of data:

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} - b = 0, \tag{19}$$

and two additional hyperplanes that are parallel to the separating hyperplane:

$$\begin{aligned} \mathbf{w}^{\mathrm{T}}\mathbf{x} - b &= 1 \\ \mathbf{w}^{\mathrm{T}}\mathbf{x} - b &= -1 \end{aligned} . \tag{20}$$

The perpendicular distance between the parallel hyperplanes, 2/||**w**||, is referred to as the classifier margin and the optimal hyperplane is the one which results in the maximum margin. The problem of determining the maximum margin can be expressed mathematically as

$$\begin{aligned} &\max_{\mathbf{w}} \frac{2}{\|\mathbf{w}\|} \equiv \min_{\mathbf{w}}(\mathbf{w}^{\mathrm{T}}\mathbf{w}), \\ &\text{subject to} \\ &z_{2.4}(g_{2.4(i)})(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i - b) \geq 1, \quad i = 1,2,\ldots,n. \end{aligned} \tag{21}$$

This is a constrained quadratic optimisation problem whose solution **w** has an expansion

$$\mathbf{w} = \sum_s v_s \mathbf{x}_s , \tag{22}$$

where $\mathbf{x}_s$ are the subset of the training data, referred to as support vectors, located on the parallel hyperplanes, and v_s are the corresponding weighting factors. The linear SVM (LSVM) decision function is then given by

$$h_{\mathrm{LSVM}}(\mathbf{x}) = z_\alpha(\mathbf{w}^{\mathrm{T}}\mathbf{x} - b) \tag{23}$$

where $\alpha = 0$.

Substituting (22) into (23) allows the SVM classifier to be expressed as a linear combination of support vector dot products, that is,

$$h_{\mathrm{LSVM}}(\mathbf{x}) = z_0\left(\sum_s v_s(\mathbf{x}\cdot\mathbf{x}_s) - b\right). \tag{24}$$

This important property allows SVMs to be extended to nonlinear classification problems using the kernel trick. This involves replacing the dot products by a kernel function which meets Mercer’s condition [12]:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j), \tag{25}$$

so that the data is mapped into a higher dimension feature space, $\mathbf{x} \rightarrow \Phi(\mathbf{x})$, where classification can be performed using a linear SVM. Here the Gaussian radial basis function (RBF) kernel,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\sigma^2}\right), \tag{26}$$

is used. The parameter, σ, controls the kernel width and is determined as part of the classifier training process. Note, in the original data space, the resulting decision function will be nonlinear and takes the form

$$h_{\mathrm{SVM}}(\mathbf{x}) = z_0\left(\sum_s v_s . k(\mathbf{x}, \mathbf{x}_s) - b\right). \tag{27}$$

In non-separable problems where different classes of data overlap, slack variables are introduced so that a certain amount of misclassification of data is allowed. A smoothing parameter C controls the amount of misclassification permitted with large values resulting in a larger penalty on errors. This ‘soft margin’ classifier is the most general form of SVM [12].

SVM training is generally formulated as a quadratic programming problem and, as such, is computationally complex for large training sets. Consequently much research effort has gone into developing efficient training algorithms for SVMs [13]. Here, training was performed using the Matlab® package simpleSVM [14], a fast SVM solver based on active constraint sets. The kernel width parameter, σ, and smoothing parameter C were optimised on the basis of classification performance on the test data set. The *a*

priori knowledge of the bi-modal distribution of '*bad*' circuits was taken into account by implementing classification as a three-class problem using the *one-versus-all* approach, as outlined in Section 3.3.

3.5. *k*-Nearest Neighbours (*k*NN)

The third classifier considered in the case study is the *k*-nearest neighbour (*k*NN) classifier [10], a nonparametric method in which a test feature vector, $\mathbf{x}$, is assigned the most frequently occurring class among the k most similar feature vectors in the training set. Mathematically, the *k*NN classifier can be expressed as

$$h_{k\mathrm{NN}}(\mathbf{x}) = z_0\left(\sum_{i \in K(\mathbf{x})} z_{2.4}(g_{2.4(i)})\right) \quad (28)$$

where $K(\mathbf{x})$ is the index set of the k nearest training set feature vectors. The similarity measure used was the Euclidian distance between vectors,

$$d_i(\mathbf{x}) = \left\|\mathbf{x} - \mathbf{x}_i\right\|. \quad (29)$$

and the optimum value of k for each classifier was determined by cross-validation on the test data set.

4. Simulation Study

To evaluate the performance of the various inferential performance classifiers under consideration and assess the value of different choices of feature vector, a Monte Carlo simulation was performed, as described in Section 2. The first 5,000 LNAs were used to train the various classifiers, while the second 5,000 formed the test set. Performance was measured in terms of the misclassification rate (MCR), the '*good*' circuit pass rate (GPR) and the '*bad*' circuit fail rate (BFR) estimated over the test LNAs. To provide robust estimates, these metrics were computed by averaging over 100 batches of 500 '*good*' and 500 '*bad*' LNAs selected from the test data set using sampling with replacement.

Table 1 shows the mean MCR obtained with each of the classifiers for five specific feature vector combinations, namely $\mathbf{x}_{DC}$, $\mathbf{x}_{1.4}$, $\mathbf{x}_{1.4+DC}$, $\mathbf{x}_{2.0}$ and $\mathbf{x}_{2.0+DC}$. The standard deviation of the MCR estimates is given in parenthesises. For comparison purposes the MCRs obtained for single frequency measurements at 1.4 ($g_{1.4}$) and 2.0 GHz ($g_{2.0}$) are also included.

The results clearly show that the indirect classifiers outperform the direct classifiers with the MLP consistently giving the lowest MCR. The *k*NN classifiers (with optimised k values) perform very poorly for this problem, but this is to be expected given the high level of overlap between classes in the feature space, particularly when the DC and lowest frequency measurements are included. More surprisingly, the SVM classifiers do not yield good results. This may be a consequence of the nature of the problem and the fact that the indirect classification approach is more suited to the form of the data. It is also suspected that the simpleSVM algorithm used to train the SVM may have deficiencies for this type of problem. This is currently under investigation.

Table 1. MCRs for various LNA classifier paradigm-feature vector combinations

Input	MCR mean (standard deviation)				
	Indirect		Direct		
	Linear	**NN**	**LDA**	**SVM**	***k*NN**
$\mathbf{x}_{DC}$	36.92 (1.57)	**35.96** (1.49)	36.95 (1.44)	37.04 (1.47)	39.01 (1.60)
$g_{1.4}$	**35.96** (1.36)	-	-	-	-
$\mathbf{x}_{1.4}$	22.16 (1.33)	**20.84** (1.31)	22.78 (1.22)	23.28 (1.46)	33.84 (1.44)
$g_{2.0}$	**20.06** (1.20)	-	-	-	-
$\mathbf{x}_{1.4+DC}$	20.58 (1.21)	**18.53** (1.25)	22.50 (1.19)	23.26 (1.29)	34.10 (1.62)
$\mathbf{x}_{2.0}$	7.12 (0.74)	**3.69** (0.55)	8.75 (0.94)	7.55 (0.88)	21.71 (1.35)
$\mathbf{x}_{2.0+DC}$	6.88 (0.77)	**2.46** (0.83)	12.31 (1.08)	7.55 (0.84)	26.18 (1.44)

The feature vectors in Table 1 are in the order of reducing MCR. It is clear that the DC measurements have only very limited value for classification, while gain measurements at high sample frequencies have the most value. The best results are obtained when the gain measurements from several sample frequencies are combined. For example, $\mathbf{x}_{1.4}$ has similar discriminatory power to $g_{2.0}$ and $\mathbf{x}_{1.4+DC}$ is marginally superior to $g_{2.0}$. This is further highlighted in Figure 1, which shows a plot of the mean MCR as a function of sample frequency, f, for MLP classifiers with feature vectors $\mathbf{x}_f$, $\mathbf{x}_{f+DC}$, g_f and g_{f+DC}. Note that the DC measurements contribute significantly to classifier performance at lower frequencies (f<1.2 GHz) but their value decreases rapidly thereafter.

Since the good pass rate (GPR) and bad fail rate (BFR) of a classifier vary as a function the classification threshold, with one increasing as the

other decreases, the threshold can be adjusted to control one or other of these metrics. The relationship between GPR and BFR for the MLP LNA classifier is captured in the operating curves plotted in Figure 2 for various choices of feature vector. Again this shows the relative merits of the different feature vectors, but more importantly, from a manufacturing perspective it shows the trade-off that can be obtained with each classifier. For example, for a target BFR of 90% a GPR of 55% is obtained with $\mathbf{x}_{1.4}$, and 65% with $\mathbf{x}_{1.4+DC}$. This increases to over 99% with $\mathbf{x}_{2.0+DC}$. Alternatively, a 100% BFR can be obtained when using $\mathbf{x}_{2.0+DC}$ if the GPR is dropped to 80%. The level of wastage at $\mathbf{x}_{1.4}$ may well be acceptable in a manufacturing context where the cost of delivering faulty product to consumers may be substantial.

5. Conclusions

In this paper machine learning based inferential classifiers are used to predict the high-frequency gain performance of a 2.4 GHz LNA. Of the various classifiers investigated, the indirect MLP classifier gives the best overall results, effectively extending the operating frequency range of ATE for LNA gain classification by 20% at 2 GHz and 42% at 1.4 GHz.

6. Acknowledgments

The authors gratefully acknowledge the financial support of Enterprise Ireland.

7. References

[1] J. Ferrario, R. Wolf, H. Ding, "Moving from mixed signal to RF test hardware development", *IEEE Int. Test Conf.*, 2001, pp. 948-956.

[2] W.Y. Lau, "Measurement challenges for on-wafer RF-SOC test", *Annual IEEE/SEMI Int. Elect. Manufact. Tech. Symp.*, 2002, pp. 353-359.

[3] M. Negreiros, L. Carro, A. Susin, "Low cost on-line testing of RF circuits", *10th IEEE Int. On-Line Testing Symp.*, 2004, pp. 73-78.

[4] D.C. Doskocil, "Advanced RF built in test", *AUTOTESTCON '92 IEEE Sys. Readiness Tech. Conf.*, 1992, pp. 213-217.

[5] M.E. Goff, C.A. Barratt, "DC to 40 GHz MMIC power sensor", *Gallium Arsenide IC Symp.*, 1990, pp. 105-108.

[6] M. Sánchez, P.C. Hung, R. Farrell, S.F. McLoone and G. Zhang, "A Structural Test Methodology Utilising Indirect Performance Estimation For Low-Cost RF Test", *Technical Report, IMWS, NUI Maynooth.*.

[7] P.E. Allen, DR. Holberg, *CMOS analog circuit design*, 2nd Edn. Oxford University Press, Oxford, 2002.

[8] S. Haykin, *Neural Networks: A comprehensive foundation*, 2nd Edn. Prentice Hall, New Jersey, 1998.

[9] S. McLoone, M. Brown, G. Irwin, G. Lightbody, "A hybrid linear/nonlinear training algorithm for feedforward neural networks", *IEEE Trans. on Neural Networks*, 9, 1998, pp. 669-684.

[10] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, Wiley Inc., 2001.

[11] V. Vapnik, A. Lerner, "Pattern recognition using generalised portrait method", *Automation and Remote Control*, 24, 1963, pp. 774-780.

[12] C.J. Burges, "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, 2, 1998, pp. 273-297.

[13] I.W. Tsang, J.T. Kwok, P.M. Cheung. "Core vector machines: Fast SVM training on very large data sets", *Journal of Machine Learning Research*, 6:363-392, 2005.

[14] S.V.N. Vishwanathan, A.J. Smola, M.N. Murty, "SimpleSVM", *Proc. 20th Int. Conf. Machine Learning*, 2003, pp. 760-767.

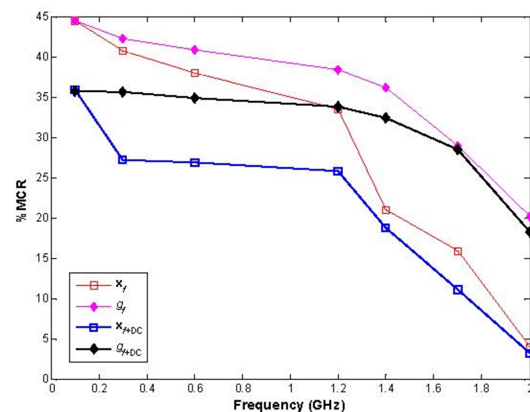

Fig. 1: MCR as a function of frequency for the MLP LNA classifier for feature vectors $\mathbf{x}_f$, $\mathbf{x}_{f+DC}$, g_f and g_{f+DC}.

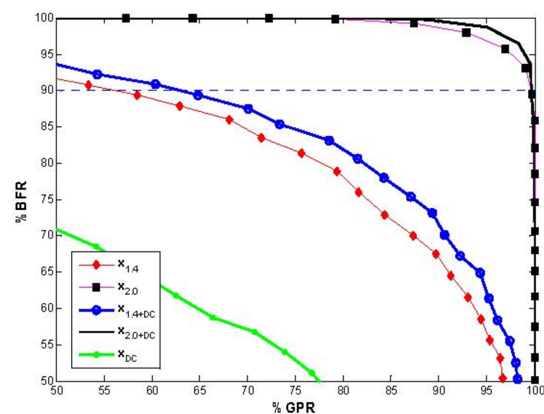

Fig. 2: MLP LNA classifier BFR v GPR operating curves for various feature vector choices.