Implementation of Identification Algorithms in Systems with Reduced Computational Resources

Anastasios Mourikis and Anthony Tzes†

Abstract—In this paper, the implementation of a classical identification technique with medium computational complexity over a system with limited computing capability, is addressed. As a testbed, the Frequency-Domain Least Mean Squares (FD-LMS) algorithm [1] is implemented with fixed--point (finite wordlength) rather than floating--point routines. The fundamental issue of faster sampling at a reduced wordlength, compared to the case of a slower sampling rate with increased accuracy (smaller roundoff errors) is investigated in the ensuing simulation studies. This problem is typical in embedded systems (controllers) with limited number crunching capabilities, where their computational power significantly limits the maximum number of operations (multiplications and additions) that can be executed within a time interval. The results of this study point towards the need of jointly optimizing the sampling rate, the wordlength size and the complexity of the assumed filter (model) in system identification cases.

***Index Terms*—System identification, Finite wordlength, Computational resources, LMS**

I. Introduction

System identification is an essential issue in adaptive control. Several identification algorithms, suitable for various system-characteristics have been developed, whose behavior in the case of precise arithmetic has been studied thoroughly in the literature (for example [2,3]). However, the task of identification may be subject to computational constrains, resulting from low-power or low-cost limitations, set by the problem at hand. In that case, the underlying arithmetic cannot be precise anymore, and this affects the accuracy of the identification process.

In this paper, we use a medium-complexity identification algorithm as a test bed for the ensuing simulations. The FD-LMS algorithm has been selected, due to the fact that it has superior convergence rate properties compared to the standard time-domain LMS-algorithm [1], while its implementation keeps the computational overhead inflicted by the use of the DFT to a minimum.

The article has the following structure. Section II presents the algorithm, and the utilized implementation [1]. In Section III, relying on the presented simulation results, we highlight several fundamental issues related to the need to achieve a compromise between the used wordlength, the accuracy on the estimated transfer function, and the need to implement the algorithm as fast as possible with limited computing resources. Section IV summarizes the conclusions of the preceding discussion.

† The authors are with University of Patras, Electrical & Computer Engineering Department, Rio 26500, Greece. Corresponding Author's e-mail: tzes@ee.upatras.gr

II. Frequency domain LMS algorithm

The estimation scheme's objective is to identify an unknown continuous-time time-invariant system. The system's input and output are sampled at a sampling rate f_s, and the estimated model corresponds to an N-th order FIR filter. Thus the system's estimated output at the nth time step is given by

$$y_n = X_n{}^T A_n \ , \qquad (1)$$

where the system's estimated impulse response at the nth iteration is denoted by $A_n = \begin{bmatrix} a_{n0} & a_{n1} & \cdots & a_{n(N-1)} \end{bmatrix}^T$ and the input signal vector, containing the delayed samples of the input x_n, by $X_n = \begin{bmatrix} x_n & x_{n-1} & \cdots & x_{n-(N-1)} \end{bmatrix}^T$.

The input signal vector and the parameter vector are transformed by the DFT:

$Z_n = W_N X_n = \begin{bmatrix} z_{n0} & z_{n1} & \cdots & z_{n(N-1)} \end{bmatrix}^T$, and $B_n = W_N A_n$, where we have denoted the $N \times N$ transform matrix of the N-point DFT by W_N. Due to the orthogonality of the DFT, we have $W_N{}^T W_N = I$, thus (1) yields:

$$y_n = X_n{}^T W_N{}^T W_N A_n = Z_n{}^T B_n \ . \qquad (2)$$

The adaptation process has now been transformed into the frequency domain, and the new system parameter vector, B_n, is recursively updated at each iteration, according to

$$B_{n+1} = B_n + 2\mu\Lambda^{-2}\varepsilon_n \overline{Z_n} \ . \qquad (3)$$

In this relation, ε_n is the error between the estimated system output, given by (2), and the measured system output d_n: ($\varepsilon_n = d_n - y_n$), μ is the adaptive step size, a positive constant value that governs the rate of convergence of the algorithm.

To ensure stability, μ must satisfy the condition $0 < \mu < \frac{2}{NP_{inp}}$, where P_{inp} is the power of the input signal, i.e. the power of each of the DFT coefficients: $P_{inp} = E(\|z_{ni}\|^2)$, $i = 0,2,..,N-1$. Λ^{-2} is a diagonal $N \times N$ matrix, whose ith element equals the estimate of the power of the ith DFT coefficient, computed by a moving average window.

To reduce the computational cost of the DFT, the algorithm is implemented recursively, using the following recursion formula $z_{nk} = z_{(n-1)k} w_N^{\ k} + x_n - x_{n-N}$. Furthermore, for the case of a real system output, $d_n \in \mathbb{R}$, we can reduce the required computations, by noting that $z_{ni} = \overline{z_{n(N-i)}}$, for $i = 1,2,..,N/2$.

III. System Identification Issues

The implementation of the aforementioned algorithm on a system with reduced computational resources, raises several issues related to the selection of various parameters such as the selection of: a) the filter's order, b) the sampling frequency f_s, used in the discretization process of the continuous system, and c) the internal number wordlength representation.

In the sequel, these issues will be highlighted over a simulation study. The need to complete a single iteration of the algorithm, within each sampling period, given a set of computational constraints (i.e., limited CPU clock-frequency), limits the selection of the algorithm's parameters.

In order to highlight several of these issues, consider the following system with transfer function $H(s) = \frac{100}{s^2 + 2s + 100}$ and an induced 3-dB frequency at $f_{BW} = 2.45\text{Hz}$.

Initially, a measure for the evaluation of the quality of the results of each simulation is defined. Rather than using the output error as a measure of goodness for the identification, we define the following error measure in the frequency domain in order to be unbiased against all sampling frequency rates

$$\text{Error} = \sum_{i=1}^{1000} (\|H(j\omega_i)\| - \|H_{\text{est}}(j\omega_i)\|)^2 . \qquad (4)$$

In (4), the frequency bins ω_i are logarithmically spaced between the frequencies of 0.1Hz and $3 \times f_{BW}$ .in order to penalize results that exhibit larger deviations in the lower frequency range.

A. System Identification Intrinsic-Parameter Settings

The algorithm's intrinsic-parameters are the fixed-point word length Q, the sampling frequency f_s, the number of samples of the estimated impulse response N, and the time interval allowed for identification, t.

Fixed point representations of Q = 32, 24, 16, 12, 10, and 8 bits were used, as these are the representations found in most commercial DSPs and microcontrollers.

The sampling frequency f_s varies between 6 and 22 times f_{BW}. Since our error criterion uses the frequencies up to $3 \times f_{BW}$, we have to sample at a frequency of at least $6 \times f_{BW}$, to avoid the effects of aliasing. On the other hand, the upper bound of 22 times f_{BW} is a practical limit set by our computational resources, since increasing f_s leads to a deterioration of the results, as will be shown further on.

The same reasoning applies to the choice of bounds for the number of parameters we use to model the identified system, N. We have found that a value of N less than 30 produces unacceptable results for all used sampling frequencies, while a N of more than 140 samples is either not feasible due to our computational resources, in most of the cases, or leads to actual deterioration of the results, as will be presented.

Finally, the time allotted for the algorithm to adapt ranges between 10 and 170 seconds. Before 10 seconds the algorithm is still behaving under the influence of initial conditions, and results are usually not meaningful, even for high sampling frequencies. On the other hand, after 170 seconds the algorithm has fully converged (up to the convergence amount allowed by the given algorithm parameters), and more iterations are not required.

The system is excited with uniform white noise sequences, and precaution has been taken to avoid overflow of each parameter used in the identification algorithm.

B. System Identification under Computational Constraints

Our objective is to determine the optimal combination of Q, f_s and N that under given computational constraints will produce the smaller identification error, as this is computed by (4). The computational constraints arise from the frequency of our microcontroller's CPU clock, compared to the natural frequency ($\omega_n = 10$) of the system under identification. For the experiments presented in this section, no time constraints were imposed, i.e. the algorithm was run until no further reduction in the output MSE was observed.

Initially, we compute the number of operations executed, and CPU clock cycles used per second by the algorithm, as a function of the finite wordlength, the sampling frequency, and the number of estimated parameters. This is done by counting the number of operations needed per iteration of the FD-LMS algorithm, and multiplying by the number of cycles each operation takes, for all values of Q.

For our implementation of the algorithm $6N$ multiplications, N divisions, and $21N$ additions of real numbers are needed per iteration. If we denote the clock cycles needed for multiplication by M, the cycles needed for

division by D, and the cycles needed for addition by A, we have

$$\text{Cycles per sec} = f_s \times (6N \times \text{M} + N \times \text{D} + 21N \times \text{A}) \quad . \qquad (5)$$

The values for M, D and A are functions of the finite wordlength, Q, and are shown in Table 1. We have used the values provided for the PIC16C5X / PIC16CXX microcontroller fixed point routines described in [4]. Note that the PIC16C5X / PIC16CXX does not support 10 and 12 bit fixed point numbers, therefore the values of cycles per operation for these representations are projections based on the multiplication and division routines. The fourth column of Table 1 contains the total cycles per second needed by the FD-LMS algorithm, computed from (5), and the last column contains the maximum achievable $N \times f_s$ product for a CPU clock frequency of 10MHz, which is the maximum allowable for the PIC16C5X / PIC16CXX family. The values in the last column have been computed using the total cycles needed (column 4 of Table 1), with an additional implementation overhead of 4-10%.

Q	M	D	A	Total cycles	$N \times f_s$ max.
32	841	909	8	$6123 \times N \times f_s$	1570
24	533	570	6	$3894 \times N \times f_s$	2424
16	284	334	4	$2122 \times N \times f_s$	4363
12	195	252	4	$1506 \times N \times f_s$	6123
10	148	191	4	$1163 \times N \times f_s$	8123
8	91	131	2	$719 \times N \times f_s$	12642

Table 1: FD-LMS algorithm Computing Requirements

From the data in Table 1 it becomes obvious that a trade-off between representation accuracy, sampling frequency, and the number of parameters used for identification has to be made. We have to choose between a larger N with few bits of accuracy or better accuracy with a small number of estimated parameters. Similarly, as the sampling frequency increases, both Q and N face limitations.

Although the data in Table 1 allow us to compute the maximum achievable N for a given sampling frequency and a given Q, it is not correct to assume that using that maximum N will yield the best possible results for the given combination of Q and f_s. Indeed, Figure 1 shows the identification error measure of (4) as a function of N for a sampling frequency $f_s = 10 f_{BW}$. The data trends that appear in this Figure are typical of all simulations, and allow us to draw useful conclusions.

As expected, the estimation error is increasing, when for a given N we use fewer bits in the fixed-point representation. When 10 and 8 bits are used, the error measure (4) ranges in the order of magnitude of thousands, so we can claim that no useful identification is possible. Contrary to that, when 32 and 24 bits are used, results are almost identical and are almost as good as the results we obtain using double precision numbers.

In addition, we observe that for a given Q, the optimum N is *not* the largest possible. The error curves for Q= 32, 24, 16 and 12 bits all have a minimum, which is located around $N = 70$ for Q=12 bits, around $N = 100$ for Q=16 bits, and around $N = 130$ for Q=32 and 24 bits. We assume that this is caused by the increased algorithm's complexity that a larger N implies. This complexity augments round-off accumulation, which eventually becomes dominant. This phenomenon is more intense in representations with fewer bits, since these algorithms suffer more from numerical errors, and therefore the error minimum appears for smaller N.

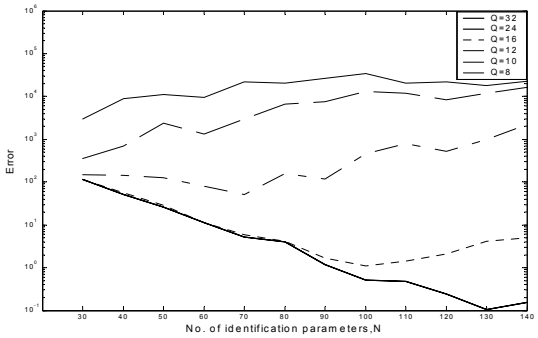

Figure 1: Identification Error vs. size of Identification Vector

From the algorithmic point of view, the fact that for large values of N we obtain worse results than those obtained for smaller values, can be explained if we consider the exact nature of the estimated parameters. We explicitly estimate the N-point DFT of the system's impulse response, B_n, which is directly dependent on the system's impulse response. This means that whatever inaccuracy occurs in the estimation of the DFT coefficients, will be reflected in the samples of the identified system's impulse response, and vice versa. However, we know that the high order samples of the impulse response contain only a small amount of the total energy of the system's impulse response, and therefore contain little information. Therefore when a large N is used, the estimation becomes significantly more vulnerable to round-off errors, and, whatever gain we have from the use of a larger number of parameters to be estimated, is lost due to numerical problems.

From the numerical point of view, this behavior of the FD-LMS algorithm is mainly attributed to the use of the DFT. The DFT is implemented in a recursive way, and therefore round-off errors are multiplied and accumulated, leading the DFT computation to numerical instability [5].

From the above discussion it becomes obvious that we have to be careful in the selection of the number of parameters we seek to identify, when we are working with limited precision numbers, and that the selection of the optimal combination of Q, f_s and N in the case of limited computational resources, requires that we first investigate the unconstrained problem.

We now examine how the identification error varies for a fixed word length, but for varying sampling rate and number of estimated parameters. Figure 2 shows the error as a function of N and f_s for Q=12, while Figure 3 presents the

same function for Q=16. In these Figures we observe that the overall minimum of the error occurs at the lowest sampling frequency, and for a moderate number of estimated parameters. Specifically, for Q=12 the minimum occurs at $f_s = 6f_{BW}$ and for $N = 50$, while for Q=16 we have minimum error for $f_s = 6f_{BW}$ and $N = 120$. It is worth mentioning that although both surfaces follow the same trend as functions of frequency (we have smaller error at low frequencies for both graphs), they exhibit a different behavior for large values of N. While for Q=12 we observe an increase in the error as N increases, for Q=16 the error is almost monotonously decreasing. This behavior verifies the trends shown in Figure 1. We may also note, that for Q=16, the estimation error increases as the sampling frequency increases. This should be expected, since when sampling at a higher sampling rate, a larger N is required in order for the estimated impulse response to contain a sufficient amount of the real system's impulse response energy.

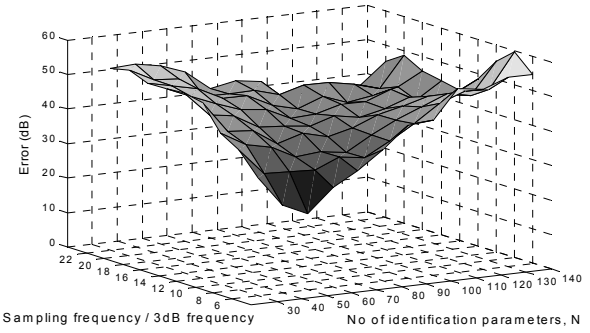

Figure 2: Identification error vs. f_s and N, w/ Q=12

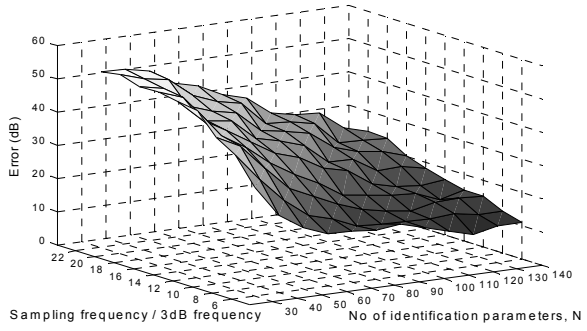

Figure 3: Identification error vs. f_s and N, w/ Q=16

Next we examine what the globally best achievable error is, for all possible values of the algorithm's parameters (namely Q, f_s and N). Figure 4 presents the best possible error for all combinations of f_s and Q. We observe that the optimal error we could obtain, for the given identification task, occurs for Q=32 or 24 bits, a sampling frequency $f_s = 6f_{BW}$, and a number of estimated parameters equal to 140. It is worth noticing in Figure 4 that the error has a minimum for $f_s = 6f_{BW}$, for all the word lengths we have used. This indicates that at higher sampling rates, we also have to use a larger *N*, in order for the estimated impulse response to contain enough energy (that is, to cover enough duration) of the impulse response of the real system, and this effectively degrades the results, through the increased accumulation of round-off errors.

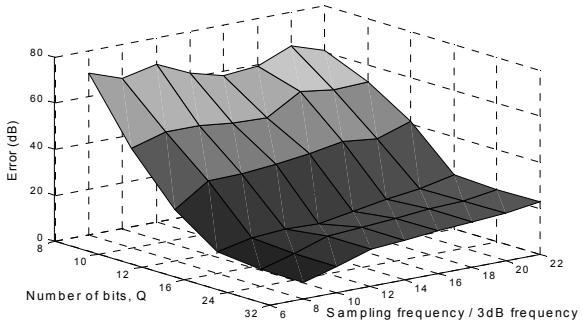

Figure 4: Achievable error vs. Q and f_s

After having examined the unconstrained problem, we may now describe a method for selecting the optimal set of algorithm parameters, when our computational resources do not allow for the use of the globally best parameters.

The effect of having limited computational resources is examined, by considering the case of a CPU with a clock that is limited to a maximum operating frequency. The clock frequency, f_{ck}, is set to successively smaller values, and the effects of this in the selection of algorithm parameters is presented. At this point we would like to point out that the following analysis does not only concern the case of a varying f_{ck}, but that it is absolutely analogous to the problem of a fixed f_{ck}, but with a system to be identified, whose natural frequency is set to successively larger values. Therefore, we can think of the ratio $f_{ck}/\omega_n = f_{ck}/2\pi f_n$ as the normalized CPU clock frequency.

Given the frequency of our CPU's clock, f_{ck} (expressed in MHz), we can compute the maximum value for the product $N \times f_s$, from Table 1. Since the values of the fifth column of the table correspond to a CPU clock $f_{\max} = 10\text{MHz}$, the maximum $N \times f_s$ product for the given f_{ck} is acquired by multiplying the elements of the last column of Table 1 by $f_{ck}/f_{\max}$. Then, for every Q, and for every possible sample rate within out set search space, we determine the maximum achievable N and compare that with the optimal N we have defined from our previous analysis. The smaller value of the two is selected as the *optimal achievable* N for that given combination of f_s and Q, and we repeat the same process for all values of Q and f_s. After we have defined the optimal achievable *N* for the entire search space, we plot the optimal identification error as a function of Q and f_s, and we create a graph similar to that of Figure 4.

The difference now is that due to the limited computational resources, for some combinations of high f_s and a large wordlength, the maximum achievable N is smaller than 30. In that case, we consider this set of

parameters as not satisfying, and we omit it from the plot entirely. For the points for which the value of N falls between the data points for which we have measurements, an interpolation is carried out.

We assume a nominal CPU clock frequency of 10MHz, and carry out the above procedure for 10 different frequency division factors, ranging form 1 to 10. This could be otherwise interpreted as considering a system to be identified, whose natural frequency ranges from 1 to 10 times the system we have been working with so far. Figures 5 and 6 present the error plots for all the achievable combinations of Q and f_s, for frequency division factors of 1 and 5 respectively. The reduction of the allowable search space is obvious, and has the effect that the minimum achievable identification error increases, as the division factor increases. We can also observe that the minimum achievable error values for some of the (Q, f_s) pairs differ from those in Figure 4, due to the fact that the optimum N for that set of parameters can no longer be used, because it requires more computational resources than those available.

Table 2 contains the values for the selected algorithm parameters, and the best achievable error given the available resources (which is the minimum of each of the surfaces as those plotted in Figures 5 and 6). Note that for a division factor of 7 and 8 (a CPU clock of 1.43 and 1.25MHz respectively), the smallest achievable error remains constant. This is caused by the fact that, for these two values of the CPU clock frequency, the optimal error is obtained using 12 bits of accuracy. We have seen however, that for Q=12 and $f_s = 6f_{BW}$ the error as a function of N has a minimum for $N = 50$. This means that although our resources may allow us to use a larger N, it is to our best interest to use only 50 parameters for system identification. This is what happens in this case, and accounts for the 'odd' behavior of the error. We should point out that the optimal sampling frequency is always $f_s = 6f_{BW}$, and that the error is generally kept down to a satisfactory level, at least until we are forced to use 12 bits of accuracy.

C. System Identification under combined computational and time constraints

In this section, an additional algorithm characteristic, the rate of convergence, is taken into account. We are interested in selecting Q, f_s and N, when time constrains are forced, that is when we only have a finite time interval during which the algorithm must estimate the system's parameters.

It is quite straightforward that the shorter the period of time we allow for the algorithm to operate, the less iterations it performs, and the larger the estimation error is. Based on this observation, one might expect that when time constraints become very strict, a higher sampling rate would be preferable, since we would allow more algorithm iterations. However, this hypothesis is not validated by the simulation study. Figure 7 shows the estimation error as a function of the estimated parameters and the sampling rate, for Q=16 and $t = 20\,\text{sec}$. We find that the error becomes minimum for a low sampling frequency $f_s = 6\times f_{BW}$, and for $N = 60$. This indicates that the improvement in performance, gained by the increased number of iterations at high sampling rates, is lost, mainly due to two factors. Firstly, when using a high sampling frequency, we need to estimate a large number of samples of the system's impulse response, and this causes increased round-off accumulation, as we have shown. Secondly, a larger number of estimated parameters has the effect of slowing down the algorithm's convergence. As a result, convergence is not yet complete for small values of t.

We use the methodology described in the previous section, to select the optimal set of Q, f_s and N, given a limited amount of time for the algorithm to operate, and a limited amount of computing power. The results for Q and f_s are not presented here, but are almost identical to those of Table 2, for each given value of t. That is, the optimal sampling rate is invariably found to be $6\times f_{BW}$, and the selection of the optimal Q follows the same trend as that of Table 2, with more precise representations chosen, when computing resources are abundant, and a decrease in the optimal choice of the finite wordlength (FWL) as f_{ck} decreases.

Figure 8 shows the best achievable estimation error as a function of the CPU clock frequency division factor, and the time we allow for estimation. As we might expect, we obtain better results when increased computing resources and longer time intervals are available, and performance improves more with the increase of t, when a low sampling frequency is used, since convergence of the algorithm is slower.

Figure 9 shows the optimal selection of the number of the estimated parameters, as a function of the time and computational constraints. Note that the axes in Figures 8 and 9 do not have the same orientation, but this is necessary for appropriate visualization of the desired characteristics. From this Figure we may conclude, that for short identification intervals, a smaller value of N yields better results. The 'step'

f_{ck}	1	2	3	4	5	6	7	8	9	10
Error	2.36	2.94	4.93	5.78	6.28	9.12	12.8	12.8	15.8	18.3
Q	24	16	16	16	16	16	12	12	12	12
N	123	120	98	74	59	49	50	50	46	41
f_s	$6\times f_{BW}$	$6\times f_{BW}$	$6\times f_{BW}$	$6\times f_{BW}$	$6\times f_{BW}$	$6\times f_{BW}$	$6\times f_{BW}$	$6\times f_{BW}$	$6\times f_{BW}$	$6\times f_{BW}$

Table 2: Optimum attainable error (and associated parameters) subject to computing constraints

that appears in the bottom right part of the plot corresponds to the point where the optimal selection of Q changes from Q=16 to Q=12.

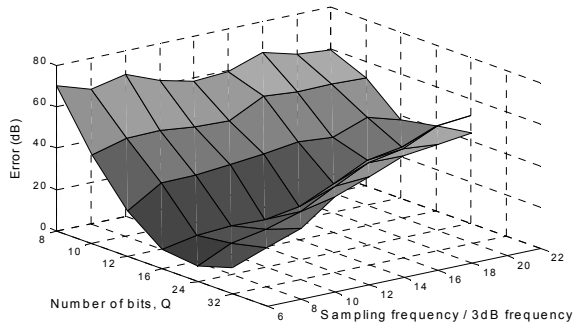

Figure 5: Attainable Identification Error for $f_{ck} = f_{\max}$

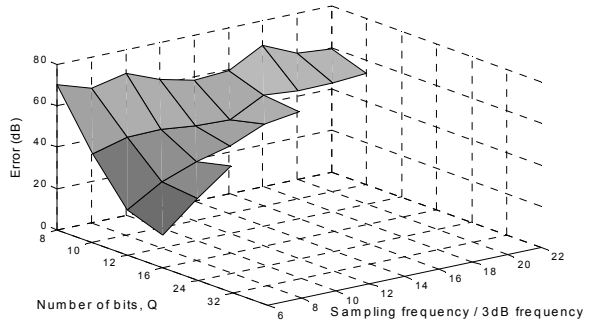

Figure 6: Attainable Identification Error for $f_{ck} = f_{\max}/5$

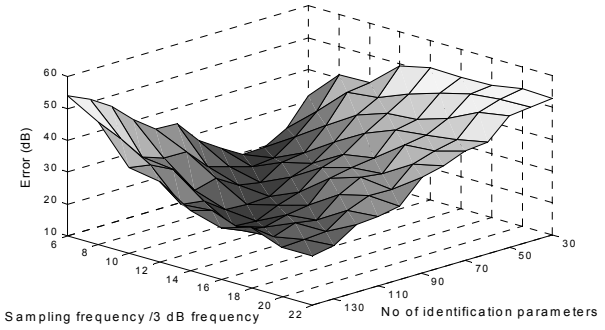

Figure 7: Identification performance vs. f_s and N, for t=20sec.

IV. CONCLUSIONS

In this article, it is shown that certain parameters related to the implementation of the identification algorithm need to be properly selected. Under reduced computational constraints, the sampling rate needs to be kept as low as possible. The lower limit for the sampling rate is the point at which the effects of aliasing begin to affect the identification of the system's response. The number of estimated system parameters is strongly dependent on the accuracy of the arithmetic, while the wordlength is predominantly determined by the computing resources available. In general, given the sampling frequency and the CPU clock frequency it is preferable to use more bits of representation accuracy with a smaller number of estimated parameters.

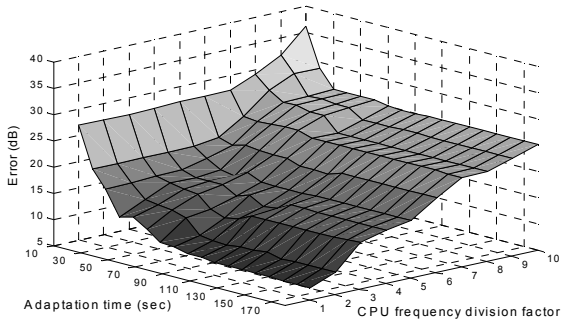

Figure 8: Lowest attainable error vs. t and f_{ck}

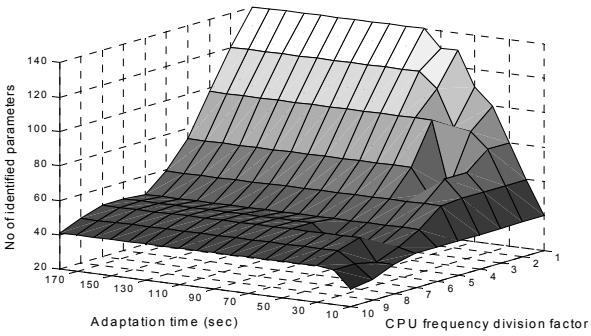

Figure 9: Optimal selection of N vs. t and f_{ck}

REFERENCES

[1] [1] S.S. Narayan, A.M. Peterson, M.J. Narashima, "Transform Domain LMS Algorithm", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-32, no. 3, pp. 609-615, Jun. 1983

[2] [2] P.E Wellstead, M.B Zarrop, "Self-Tuning Systems: Control and Signal Processing", Wiley Publishers, 1991

[3] [3] K.J. Åström, B. Wittenmark, "Adaptive Control", Addison-Wesley Publshing, 1995

[4] [4] F.J. Testa, AN617 Application Notes: Fixed Point Routines for the PICmicro Microcontroller Families, www.microchip.com

[5] [5] J.H Kim. T.G Chang, "Analytic derivation of the finite wordlength effect of the twiddle factors in recursive implementation of the sliding-DFT", IEEE Transactions on Signal Processing, vol. 48, no. 5, pp. 1485-1488, 2000