TU BERLIN

Adaptive Filter course

Praxis project: Filter implementation

LMS, KLMS & ρRLS Algorithm Analysis

Author:
Thunus Stéphane, 398881

Supervisors:
Elvira FLEIG,
Rolf JONGEBLOED

09. November 2020

Table of Contents

List of figures

1 Introduction

This paper analyses the forgetting factor recursive least squares filter (ρRLS) and the (kernel) least means squares (KLMS and LMS), all being supervised machine learning algorithms and part of the adaptive finite impulse response (FIR) filters (AF) class. They compare one unprocessed and one processed signal (supervised learning) and determine in real time (fitting, adapting) the equivalent FIR system responsible for the perceived changes. The notion of equivalent FIR system is important, as the system to be fitted isn't necessarily a FIR, or even linear, in which case, an FIR creating a similar output least squared-wise is fitted.

Like all machine learning algorithms, the fitting models have (model-) parameters, hyper-parameters and goodness of fit metrics. The only model-parameters of the ρRLS and LMS filters are the weights fitting the desired system, of which there are as many as filter taps, which is the filter's order +1 (Moschytz & Hofbauer, 2000). The KLMS algorithm stores directly a selection of representative inputs as centers and weighting coefficients (Liu, Príncipe, & Haykin, 2010). The number of coefficients is reduced by heuristics but are not directly under control and don't correspond to the impulse response of the equivalent FIR system.

The hyper-parameters, whose optimal settings this paper is concerned about, are different for each filter type and depend strongly on the filter's operation environment, as will be further investigated. Common hyper-parameters of the ρRLS and LMS are the filter order, while the LMS and KLMS both require choosing a step-size. The ρRLS also requires setting the forgetting factor ρ and an initialization value of the scalar matrix. The LMS has no further hyper-parameters, while the KLMS's further parameters depend on its type (Kernel function, dictionary sparsification heuristic, etc).

All three filters can be evaluated with the same goodness of fit measures. Convergence speed and accuracy, as well as oscillation tendencies are analyzed with the mean squared error (MSE) and the momentary absolute prediction error (APE). For further intuition, the filter output will be graphically compared to the desired output. Furthermore, the ρRLS and LMS allow direct comparison to the desired weights if the system to be emulated is known, as is the case for some systems in this test series.

Section 2 comments the implementation of the filters and the related tests, while section 3 discusses properties and parameter choices common to all performed tests. Section 4 and section 5 focus on the fitting of IIR and FIR systems in respectively stationary and non-stationary noisy environments. Section 6 compares the LMS's and KLMS's prediction capacity of single variable time series, while section 7 and section 8 conclude with methodology critique and further work.

2 Implementation

2.1 Grid search

A grid search function was implemented to construct and evaluate mesh-grids of hyper-parameter values and determine the best combination based on the resulting MSE values. In addition, the function plots all resulting MSE and outputs a table containing the test results (Filter-type, Hyper-parameter combinations, final MSE) and a dictionary to be directly unpacked into the constructor of the tested filter class. Furthermore, to perform hyper-parameter searches for prediction filters, a "future" variable allows to arbitrarily delay the signal. To decrease the KLMS's high overfitting risk, grid search offers the possibility of using test-signal prediction MSE values rather than the input or desired signal training ones.

2.2 Filter classes

2.2.1 Adaptive filter superclass

The main content of the adaptive filter library is the abstract super-class "adaptive filter" from which all filter types inherit most of their functionality. In a classical OOP approach, all common procedures like printing the common parameters and getter functions for numbers of taps and weights are handled by the abstract super class. It has no constructor as no variables are common between all filters and some methods are abstract to force all subclasses to overwrite them. It implements the two main filter functions adapt() and filter(), as the general procedures are the same for respectively all and most filters.

The adapt() function, adapting the filter to the given system, while also providing graphical representation and outputting relevant data, differs almost only by subtype-polymorphism of the filter-dependent iteration() function. Furthermore, the MSE computation was implemented manually in a vectorized fashion that outperforms the sklearn library.

The filter() function applies the adapted filters by either calling a built-in LTI-filter application function for the LMS and ρRLS or using the KLMS filter output formula.

2.2.2 LMS

The LMS class implements a constructor, which, after input validity verifications, sets the passed step size μ, being the single hyper-parameter, in addition to initializing the weights vector with zeros. A print function overloads the superclass' print() function and the iteration() function follows the pseudo code delivered by Moschytz and Hofbauer (2000). All other functions are provided by the superclass adaptive filters.

An auxiliary function LMSconverge() makes convergence predictions based on input signal power and filer order, based on Moschytz and Hofbauer (2000). The function is to be executed before constructing the filter or performing the grid-search to gain insight of the maximal converging step-size. If a specific step-size is given, the function predicts if the filter converges and the convergence time.

2.2.3 ρRLS

Similarly to the LMS class, the ρRLS class only implements a constructor and its own iteration() function from Moschytz and Hofbauer (2000) and overloads the print function, while taking all other functionality from the superclass. The constructor sets after verification the two ρRLS hyper-parameter being the forgetting factor ρ and the recursive autocorrelation matrix $\mathcal{R}_k$'s initialization parameter a for the initialization as $\mathcal{R}_0 \coloneqq a \cdot I_N$ with N being the number of filter taps. Similarly to the LMS class, the constructor initializes the weights vector with zeros.

2.2.4 KLMS

The KLMS algorithm with Platt's novelty criterion and gaussian kernel was designed from the pseudo-code provided by Liu, Príncipe and Haykin (2010). As for both previously discussed classes, the KLMS class implements only a constructor and its specific iteration() function and overloads the print function. The filter output function "f()" was separated from the iteration() procedure, being also used by the filter() function.

The high algorithm complexity motivates aggressive optimization, thus all weighted sums were changed to scalar products, as vectorized operations are 3-4 orders of magnitude faster than sequential for-loop iterations. To illustrate, $\sum_{j=0}^{i-1} a_{i-1}[j]\ K(\underline{x}[i], \mathcal{C}_{i-1}[j])$ becomes $\langle a_{i-1}; K(\underline{x}[i], \mathcal{C}_{i-1})\rangle$, were a_{i-1} is the weight vector and $\mathcal{C}_{i-1}$ the center vector at iteration i. The gaussian kernel function was also vectorized, such that matrix-vector operations are performed rather than sequential elementwise application, also accelerating the operation by several magnitude orders.

Platt's novelty criterion was implemented for dictionary size sparsification and reducing model overfitting. Both thresholds (minimum distance and minimum error) were set to be variables passable to the constructor, allowing grid-search optimization.

To reduce the number of necessary procedures, the constructor initializes the centers and coefficient vector with zeros, having no effect in the sum and thus not falsifying the computations. After model fitting, these initialization values are deleted from the center ($\mathcal{C}$)and coefficient (a) lists.

This KLMS implementation has 5 hyper-parameters. Those are the step-size as for the LMS, the kernel size regulating the fitted gaussian distributions' width, the window-size being equivalent to regular FIR filters' tap amount and both novelty criterion thresholds, respectively controlling the minimum distance and error necessary to be added to be dictionary.

3 General tests

This section tests parameters and properties independent of the environment stationarity and the FIR or IIR characteristic of the emulated system.

3.1 LMS: Step-size related to order

To gain insight in the general dependence of the required step-size to the filter order, an arbitrary step-size list was tested for orders 0,1 and 4 (= 1,2,5 filter taps respectively) on different systems and additive noise variances.

Step size µ	1 tap MSE	2 taps MSE	5 taps MSE
1	10.7974	∞	∞
0.5	0.1875	2.1165	∞
0.25	0.1186	0.1333	0.0084
0.1	0.1062	0.1010	**0.0006**
0.05	0.1035	0.0955	0.0008
0.025	**0.1027**	**0.0936**	0.0013
0.01	0.1036	0.0938	0.0031
0.005	0.1059	0.0960	0.0060
0.0025	0.1108	0.1010	0.0120
0.001	0.1253	0.1157	0.0296
0.0005	0.1496	0.1405	0.0592

Table 1: Step-size compared to MSE

The adjacent table shows a test results extract for the mapping of the provided FIR system with different filter orders in a stationary environment with an additive noise variance of 0.001, resulting in a total signal power of 0.986. The step-sizes resulting in the minimal MSE for the respective orders are marked in bold.

The first observation is that step-sizes over a certain value either diverge as seen from the ∞ in the table or result in great MSE values and thus bad approximations of the desired system. Furthermore, increasing the number of taps reduces the maximal step-size as already formalized by Moschytz and Hofbauer (2000) in $0 < \mu < \mu_{\max} = \frac{2}{tr(R)} = \frac{2}{N \cdot \mathbb{E}\{|x[k]|^2\}}$ with $\mathbb{E}\{|x[k]|^2\}$ being the input power, N the number of filter taps and R the signal's autocorrelation matrix.

Furthermore, the step-size µ relation to the MSE behaves in a convex way (Moschytz & Hofbauer, 2000). Too large step-sizes don't attain the minimum of the hyper-parabolic error surface, as the weights keep oscillating around the minimum (ibid.), while too small step-sizes don't converge fast enough to reach a minimum before the training signal ends.

The above Table also hints that greater orders of LMS provide better approximations. The relation with the MSE and the noise influence are respectively discussed in each chapters' noise resistance section.

3.2 LMS: Step-size related to additive noise variance

The grid-searches calculated for the previous section were used to test if the additive noise variance has an influence on the step-size but no such relationship was discovered. If the noise doesn't significantly increase the input power, which requires a reduction of the step-size as noted above, the step-size remains unaffected by this study's findings. Further, more intricate tests could however reveal some relationship, being however outside this paper's scope.

3.3 RLS: Initial scalar matrix value

The ρRLS algorithm requires initializing the recursive autocorrelation matrix $\mathcal{R}_k$. The common choice is a scalar matrix ($\mathcal{R}_0 \coloneqq a \cdot I_N$) having the useful property of being full rank, necessary for its inversion by the algorithm (Moschytz & Hofbauer, 2000) (Haykin, 2014). Moschytz and Hofbauer (2000) propose the use of an arbitrary large number, while Haykin (2014) proposes an initialization based on the signal to noise ratio (SNR). This section proposes a thorough test on the various influences the initial value a has on the convergence behavior of the ρRLS algorithm.

Haykin (2014) proposed that higher SNR required large positive numbers while low SNR data best converges with small initialization values. In this test series, the signal to noise ratio is directly influenced by the amplitude/variance of the noise signal added to the input signal. To illustrate, the signal with the added noise variance of 10 has a very low signal to noise ratio and is expected to require a small-valued initialization.
This study's results confirm Haykin's (2014) findings in general trends since no exact optimum initialization values were computed.

The grid search is composed of a base 2 logarithmically spaced 50 number series in $[e-06; e+05]$ with additionally a linearly spaced 20-number series of ρ values and the four noise levels and four tested systems resulting in $50 \cdot 20 \cdot 4 \cdot 4 = 8'000$ fitted filters. This data amount couldn't be displayed or added to a reasonably sized annex, thus only the results and selected plots will be presented. The exact results are available in the datafiles provided with the paper.

The result showed that the correct initialization has more beneficial effect on the MSE for higher order filters (here 5 taps) than for lower order filters. Furthermore, more difference was observed for better signal to noise ratio or input power, which are indiscernible by this test series, requiring further tests varying the input level in addition to the noise level. A correct initialization also has beneficial influences on the convergence speed as illustrated in the below figure.

The maximal observed MSE values were mostly at the opposite end of the optimum initialization value in the grid search. To illustrate, the high SNR scenario, optimally initialized with a large a has tendentially its minimum at the smallest value tested by the grid search (clipping), while the optimum varies strongly. Upon finding these results, the respective grid searches for all four provided test inputs were enlarged by 3-4 magnitude orders (for example from 0.01 to e-06), which resulted in the same behavior. Thus, one can conclude that while the optimum is asymptotically bounded (see below figure), the worst performance isn't, meaning that the error can become arbitrarily big with a poor initialization.

Furthermore, an important finding is that the MSE relation to factor a in the tested range of [e-06; e+05] is convex, meaning that there are no non-global optima as illustrated by the below figure. Thus, a grid search can extend its search in the direction reducing MSE values, being guaranteed to reach an optimum and immediately stop when the MSE starts rising. The step size can then be reduced to approach the optimum with desired precision.

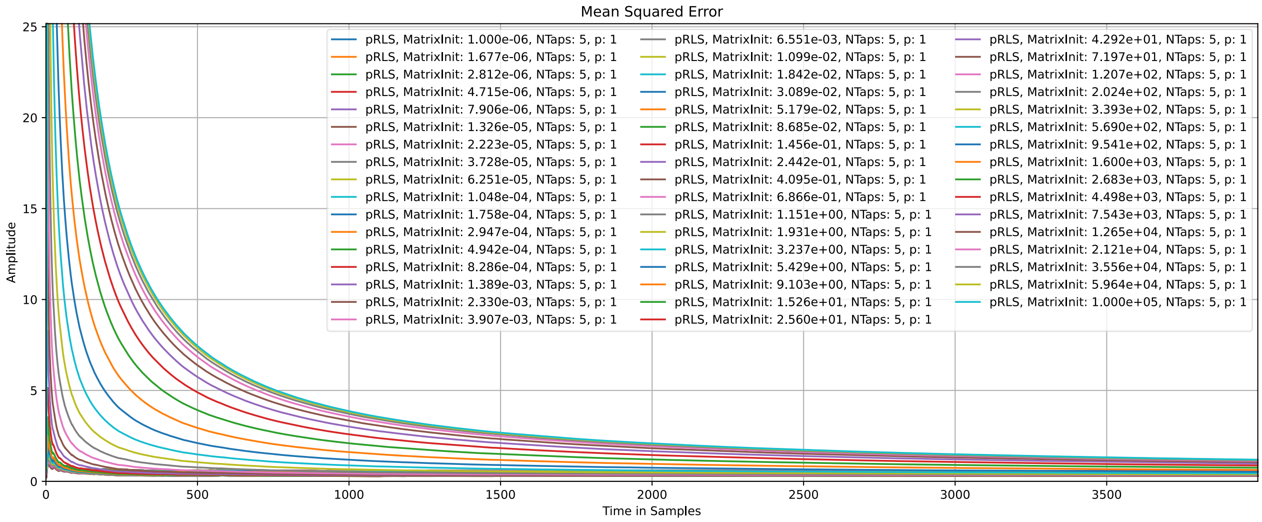

Figure 3.1: ρRLS: scalar matrix initialization vs MSE

The above figure illustrates some of the aforementioned findings, displaying an extract of the static environment FIR's grid search. The filter characteristics are : $\rho = 1$, 5 taps (order=4), additive noise amplitude NA=1, resulting in Input Power =1.99 and the tested values for the initialization parameter a are from e-06 to e+05. The grid-searches of the other systems and noise levels look similar.

This graph shows the initialization factor a's influence on final MSE values and convergence speed. The entries corresponding to the highest a (right legend list entries) give the worst results in their order of magnitude (1e+05 worse than 5.96e+04, etc). This order continues until the red second-column entry 5.179e-02, being the optimum value in this test. Smaller values result in a greater MSE, thus an efficient grid search could stop upon encountering 3.089e-02, which yields a higher MSE, due to the found convexity relation.

4 Stationary signals

The first test series focus on the convergence behavior and estimated filter quality of ρRLS and LMS filters in a stationary environment. Stationary means that the signal's statistical properties are invariant during the entire signal (Moschytz & Hofbauer, 2000). In this study, this means that the additive noise level/variance stays the same and that the system to fitted by the AFs remains static.

4.1 Common aspects

This section describes filter properties common to all applications on stationary signals, irrespective of the system to be approximated. Thus, the results of this section count for the following FIR and IIR sections. The used files respectively are "FIR" and "IIR".

4.1.1 Optimal ρ value for ρRLS

The forgetting factor ρ's optimal value is a compromise between convergence speed and tracking ability (Moschytz & Hofbauer, 2000). In stationary environments, the best value for ρ is 1, meaning that the filter considers all inputs for the autocorrelation matrix estimation $\mathcal{R}_k$, without progressively discarding older inputs (ibid.)

In accordance to Moshytz and Hofbauer (2000) the recursive autocorrelation matrix denoted $\mathcal{R}_k$ approximates the input signal's autocorrelation matrix R, whose precision increases with Dataset size (k): $\mathcal{R}_k \approx R\frac{1-\rho^{k+1}}{1-\rho}$ and $\mathbb{E}\{\mathcal{R}_k\} = R\frac{1-\rho^{k+1}}{1-\rho}$. Thus, having the smallest possible information lost $(\rho = 1)$ increases the filter performance in stationary environments.

Moshytz and Hofbauer (2000) also predict that the fitting error is $M_{RLS} \approx \frac{1-\rho}{2} \cdot N$, meaning that a noise-less environment and a forgetting factor of 1 can lead to an almost zero fitting error.

ρ	MSE	ρ	MSE
1	0.6186	0.9473	0.7007
0.9947	0.6248	0.9421	0.7099
0.9894	0.6325	0.9368	0.7191
0.9842	0.6405	0.9315	0.7285
0.9789	0.6487	0.9263	0.7380
0.9736	0.6571	0.9210	0.7476
0.9684	0.6655	0.9157	0.7573
0.9631	0.6741	0.9105	0.7672
0.9578	0.6829	0.9052	0.7771
0.9526	0.6917	0.9	0.7873

Table 2: ρ-value compared to MSE

A grid search was conducted over the usual noise variances (0.001, 0.1, 1, 10) and number of filter taps (1,2,5) with 20 linearly spaced values between 0.9 and 1 for the stationary FIR and IIR systems, resulting in $4 \cdot 3 \ \cdot 20 \cdot 2 \ = 480$ tested parameter combinations. The results confirmed the theoretical assumptions, as every time $\rho = 1$ was selected, delivering the lowest MSE.

The adjacent table is one of the many generated by the grid search, which all display a common trend. The tested ρRLS has 5 taps (order 4) and the maximum noise variance of 10 was selected with $\mathcal{R}_k = 10k \cdot I_n$. The MSE increases by about 0.006 at every 0.005 decrease in ρ in the beginning and about 0.01 towards the end. Thus, the MSE increase grows more than proportionally to the ρ-value decrease. Moreover, an expanded grid-search range for ρ indicated that the MSE decreases monotonically, indicating a monotonous relationship between ρ and the MSE.

4.2 FIR adaption test

This section describes the test procedure and results of fitting an FIR system in a stationary environment. All procedures are described in detail and the corresponding theorems are provided, while their relation to the test data is explained in great detail to set the basics for the identical further tests.

4.2.1 Filter order impact

4.2.1.1 LMS

This section confirms the expectation that a higher order filter being a fitting model with more degrees of freedom approximates the given system with higher precision. As noted in section 3, the optimal step-size μ depends amongst other factors on the filter order. Thus, to avoid favorizing a particular order with an arbitrarily chosen step-size, a grid-search decided of the respective optimal step-size. The grid-search consist of 60 logarithmically (base 2) spaced values in the order dependent interval [MaxMu/2 ; MaxMu/2048]. No noise was added to the input signal to measure the actual unhindered performance of the respective filter orders.

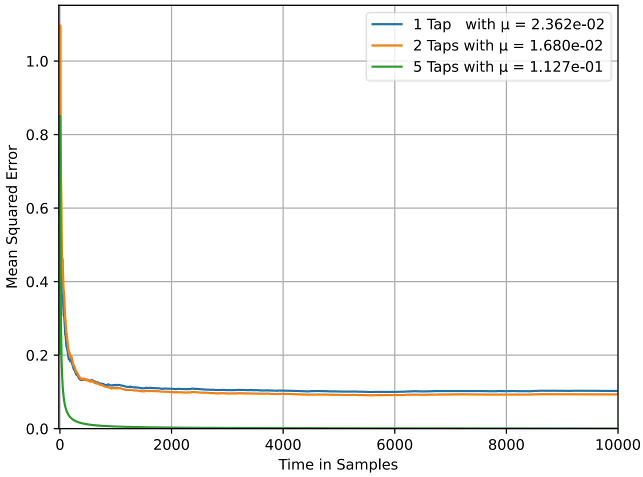

Figure 4.1: LMS order comparison

The above figure displays the asymptotic MSE convergence curves of the LMS Filters of order 0, 1 and 4 ranked by their descending order. Higher order filters thus perform better (lower MSE) than their low order counterparts on noiseless signals since more taps are available to estimate the system. The MSE values of the filters in ascending order are $\{0.102, 0.093, 5.8e-04\}$.

Comparing the resulting filter weights of the respective orders ([0.7, 0.1, -0.03, 0.18, -0.24], [0.70216226, 0.07588638] and [0.70905876]) shows that the fitted coefficients become gradually noisier as the order diminishes. The 4th order filter maps the weights with zero error, while the first and zeroth order filter respectively present an estimation error of 0.002 and 0.009.

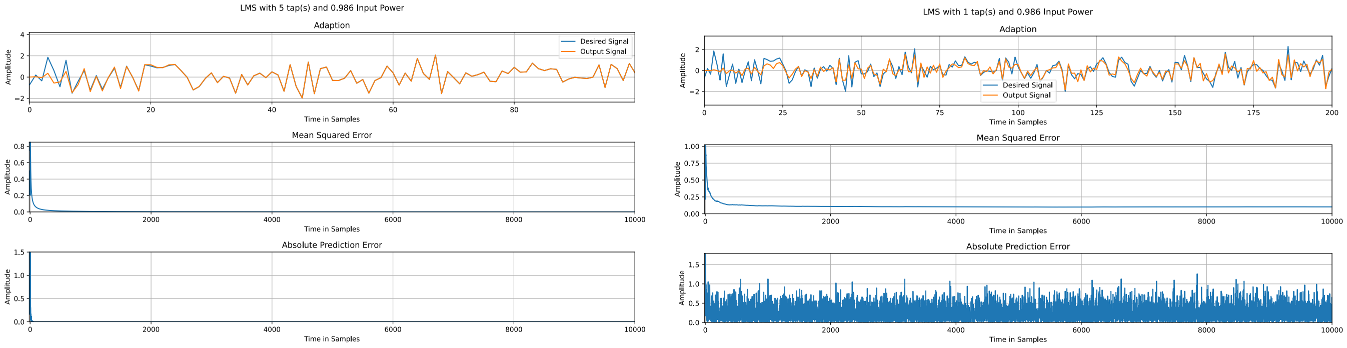

Figure 4.2: LMS: 5 tap vs 2-tap adaption

The left subplot in the above figure displays the 5-tap filter's adaption process converging after about 10-15 iterations. The top subplot compares the filter output to the desired signal, being the unknown filters' output signal when fed the same input as the adaptive filter. Both following subplots display the same information, with the middle one being the average of the squares of the bottom one (MSE), being thus less precise. The bottom plot displays well the transient nature of the error and thus the convergence speed, while the MSE smears the transient to an asymptotical behavior. The absolute prediction error directly influences the weights change rate, as large errors lead to greater weight changes and no error leaves the weights untouched.

The right subplot of the above figure displays the single tap filter adaption process (resembling the 2-tap filter), which doesn't converge to a zero-error state, as the desired filter is of higher order. The top subplot displays the still surprisingly good original signal approximation, which might be due to the desired system further coefficients being relatively small compared to the first one. The subgraphs below display the MSE, which can also be seen in the above grid search in Figure 4.1, following the usual asymptotical convergence trend up to 0.1027. The bottom-most subgraph displays the absolute prediction error at every iteration, which doesn't change much after the first 400 iterations. The prediction being different than zero means that the filter weight vector oscillates around its current value. The slower convergence is partially due to the smaller step-size.

4.2.1.2 ρRLS

Similarly to the previous LMS test, this section verifies that higher order ρRLS filters create better approximations. As before, to analyze the true fitting capacities of the filter, no additive noise was added to the input signal. Since the filter operates in a stationary environment the forgetting factor ρ was set to 1. The factor of the scalar matrix being independent of the order and the signal to noise ration being practically infinite (no noise), 10k was taken.

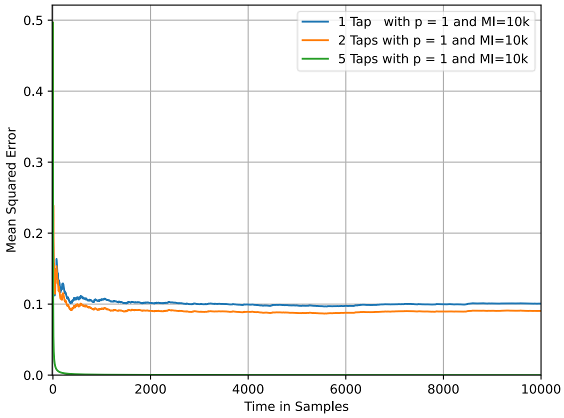

Figure 4.3: ρRLS order comparison

Moschytz and Hofbauer (2000) propose

$$M_{RLS} \approx \frac{\mu_0}{2} \cdot N \cdot \text{Input power} \; = \frac{1-\rho}{2} \cdot N$$

as fitting error approximation for an ρRLS filter. Thus, in stationary environments where $\rho = 1$ is the optimal forgetting factor, the filter error should be around zero. This study's author assumes that Moschytz and Hofbauer restrict their approximation to FIR filters of order equal and smaller than the used adaptive filter used in noise-free environments as none of these variables are contained in the approximation.

As expected, the 4th order filter converges very quickly (5 iterations) and fits the unknown system with almost no error, while the lower filter orders take longer to converge and don't achieve an error free approximation. Similarly to the LMS test, the 4th order filter weights remain static after convergence, while the lower order filter's weight oscillate. The resulting MSE values are ranked in descending tap order, with the highest MSE corresponding to the lowest order. The 5-tap filter achieves an MSE of 5.987e-05, while both other filters achieve respectively 0.1 and 0.09.

The resulting filter coefficients are respectively [0.7, 0.1, -0.03, 0.18, -0.24], [0.7002266 0.1022911] and [0.6997884]. The 4th order filter approximation is error free, while the lower orders have mostly errors in the e-04 region.

4.2.2 Noise resistance

4.2.2.1 LMS

This section compares the effect the addition of uncorrelated data to the input has on the LMS algorithm's fitting capacity. The added noise variances/amplitudes are $\{0.001, 0.1, 1, 10\}$ for an input peaking at about 1.2. To mitigate the step-size's (hyper-parameter µ) effect as covariate, a grid-search chose the optimal value for each noise level and filter order. Thus, the range between the respective MaxMu/2 and MaxMu/2048 was logarithmically (base 2) divided into 50 steps. The following plots display the MSE evolution of respectively order 0, 1 and 4 filters with different noise variances.

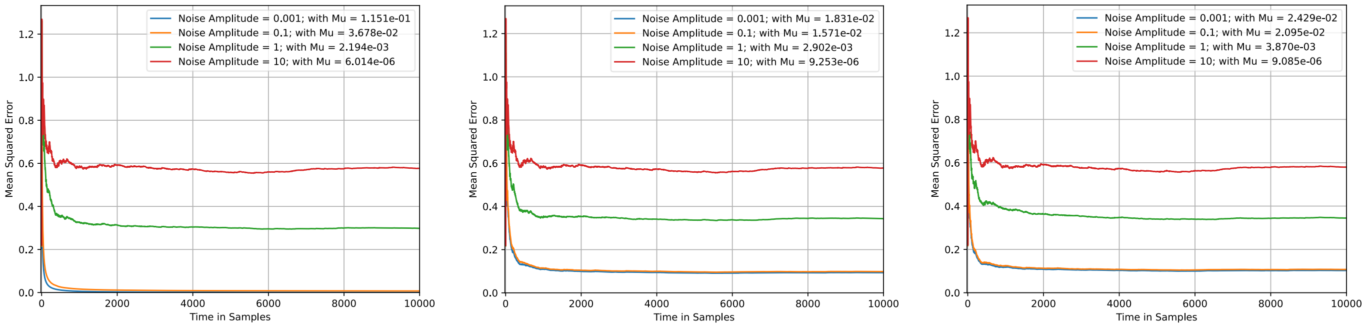

Figure 4.4: LMS: noise amplitude vs filter order 4, 1 and 0

Unsurprisingly, the best MSE values are ranked in ascending noise-amplitude, meaning that the noise amplitude negatively affects the overall fitler performance. The additive noise variance also strongly reduces the chosen optimal step-size as for all filter orders, which as displayed in the below table, leaves the weights around zero.

The above illustrations reveals that above a certain noise level, the data is so corrupted that having the minimum necessary order to fit the filter makes no difference anymore, as all red curves (noise amplitude NA=10) are identical. Those signals don't converge, which is not surprising considering that the noise signal has an amplitude 10 times higher than the input signal, drowning it out. The 0.1 and 0.001 noise signals converge to almost the same MSE, while the NA=1 signal converges to a higher level. This means that the prediction error isn't zero and that thus the filter weights oscillate around their value rather than converging.

Taps	Noise	W0	W1	W2	W3	W4	MSE
		0.7	0.1	-0.03	0.18	-0.24	
1	0.001	0.709134	/	/	/	/	0.10273
1	0.1	0.696418	/	/	/	/	0.10273
1	1	0.696418	/	/	/	/	0.10273
1	10	0.005726	/	/	/	/	0.58
2	0.001	0.703267	0.073795	/	/	/	0.09334
2	0.1	0.688233	0.089681	/	/	/	0.09824
2	1	0.354273	0.097480	/	/	/	0.34299
2	10	0.007252	0.000102	/	/	/	0.57776
5	0.001	0.700126	0.099799	-0.029894	0.17998	-0.240032	0.00058
5	0.1	0.687523	0.1	-0.038086	0.188539	-0.22779	0.00738
5	1	0.352613	0.058993	-0.007846	0.096573	-0.121376	0.29306
5	10	0.006727	0.000905	-0.001825	0.002357	-0.002848	0.57948

Table 3: static LMS: noise amplitude vs weights and MSE

The above table displays the final weight value for each filter order at different noise variances and their final MSE, while the top row displays the desired filter values. This table confirms the intuition that higher noise levels result in poorer fitting capacities. As noted in previous paragraphs, all filter orders perform equally as poorly at very high noise levels, yielding weights being two magnitude orders too small and similar MSE values around 0.58. The NA=1 weights are half their correct amplitude for orders 2 and 4.

As Moschytz and Hofbauer (2000) predicted, higher order filters are more sensitive to noise, having more parameters affected by it, as seen from the last column. The 1st order LMS's MSE remains at 0.103 for the three lowest noise levels, while the 4th order LMS jumps from 5.8e-04 to 0.29, being a 505-fold increase. The same effect is visible from the weights themselves, as the 1st order LMS remains within e-04 of the minimum-noise fitting while the 4th order LMS quickly degenerates from the almost perfect estimation of 0.7001 to 0.3526.

4.2.2.2 ρRLS

The noise resistance tests for the ρRLS were performed with $\rho = 1$, being in a stationary environment. Similarly to the LMS, the covariate effect of the scalar matrix initialization was contained by performing a logarithmically (base 2) spaced grid-search of 60 parameters from e-06 to e+05. Only the results of the 5-tap filter are plotted, as similarly to the LMS, the graphs are identical with slight vertical translations of the lower orders and a static NV=10 curve.

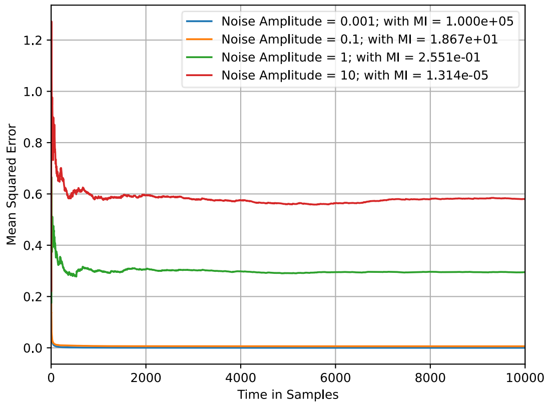

Figure 4.5: ρRLS: Noise Amplitude vs MSE

As expected, noisier signals yield higher MSE values and exhibit a more spread out asymptotical transient before stabilizing their trajectories. The filter fitted on the NA=10 signal doesn't converge but fluctuates strongly despite the strong smoothing effect of the MSE's average. This means that the absolute prediction error (APE) constantly changes and thus the weights fluctuate around their value. The signals with the least difference in noise amplitude are close to each other, while the MSE with greater amplitude difference are further away, illustrating the noise influence on MSE values.

Furthermore, the scalar matrix initialization's value diminishes with every noise addition amplitude increase, confirming the results of section 3, linking it to the signal to noise ratio.

Taps	Noise	W0	W1	W2	W3	W4	MSE
		0.7	0.1	-0.03	0.18	-0.24	
1	0.001	0.69979	/	/	/	/	0.10066
1	0.1	0.69264	/	/	/	/	0.10605
1	1	0.35362	/	/	/	/	0.33862
1	10	0.00661	/	/	/	/	0.57928
2	0.001	0.70023	0.10229	/	/	/	0.09044
2	0.1	0.69388	0.10087	/	/	/	0.09503
2	1	0.69388	0.04760	/	/	/	0.34319
2	10	0.00681	0.00170	/	/	/	0.57887
5	0.001	0.69984	0.09997	-0.02988	0.17992	-0.2400	0.00599
5	0.1	0.69358	0.09120	-0.02930	0.19677	-0.2399	0.01965
5	1	0.36222	0.04678	-0.03783	0.14539	-0.1370	0.40805
5	10	0.01463	0.01277	-0.01774	0.01564	-0.0041	0.69108

Table 4: Noise vs weights and MSE

The above table displays the ρRLS noise resistance depending on its order and the resulting weights and MSE values. As expected from the previous plot, the MSE and difference between desired and fitted weight values increase with noise amplitude. Noise amplitude 0.001 and 0.1 yield almost perfect results for each order, while NA=1 predicts the weights only correctly for the 1st order ρRLS, while both other orders predict half the correct weight. The highest order filter shows a 4-fold MSE comparing NA=0.1 to NA=0.001 despite yielding similar weights, which is partially an artefact of the MSE computation as explained in section 7.

4.2.2.3 RLS vs LMS

This section compares both filters performance with each filter having their respective optimal hyper-parameter values as explained in both previous sections. Only the 5-tap filter evolution comparison is plotted below, since, as noted in the previous sections, their evolutions are similar but with vertically translated low-noise curves and with more stretched out asymptotical behavior.

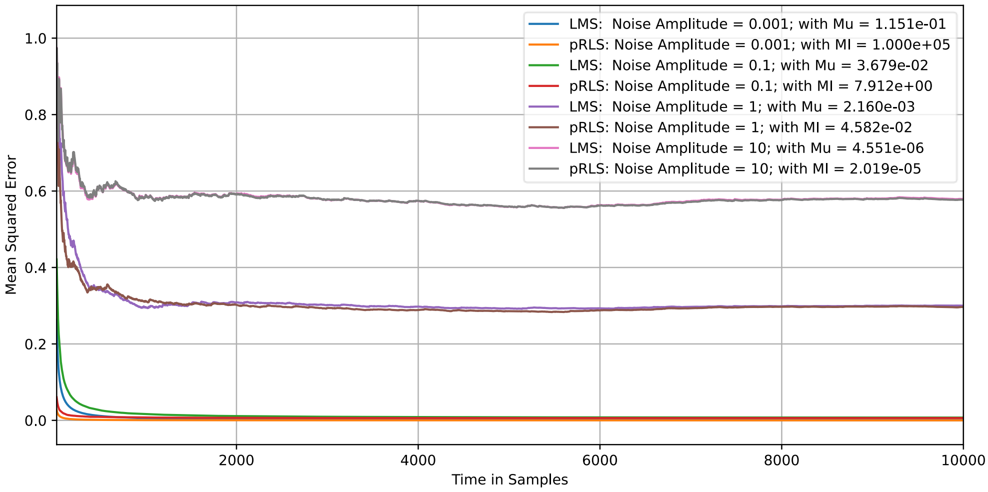

Figure 4.6: LMS vs ρRLS: noise amplitude vs MSE

As illustrated above, the ρRLS filters converge or stabilize faster than their LMS counterparts for every order. The NA=0.1 ρRLS converges even faster than the NA=0.01 LMS, while both filters fluctuate in an almost identical manner for NA=10 and NA=1 indicating that the system can't properly be approximated by neither filter.

The results tables 3 and 4 show that both filter's MSE values are very similar to each other for both lowest orders. However, the 4th order LMS's MSE values beat the 4th order ρRLS, even by an entire magnitude order for the lowest noise amplitude (NA=0.01).

Despite the single tap ρRLS's MSE being higher than its LMS counterpart the final weight's difference to the reference is one magnitude order higher. This result disparity is commented in section 7. The 5 taps (order 4) filter weights are generally more precise than their lower order counterparts for both filters. NV=10 yields weights that are 2 magnitude orders away from the correct ones (for example 0.0064 for the RLS and 0.0051 for the LMS instead of 0.7), with the RLS slightly outperforming the LMS. NV = 1 results in both filters predicting weights that are half the magnitude of the desired ones for all orders, with the RLS often being better with a difference in e-03 or e-02. For NV=0.1 both filters approximate the filter coefficients with an error in the e-03 range, while the error is mostly in the e-04 range for NV=0.001.

In conclusion, both filters only produce reliable results for noises being at least a magnitude order lower than the signal containing information about the system to be approximated. Furthermore, the ρRLS outperforms the LMS in most cases, which might be due to the faster convergence, which, however, becomes marginal for longer signals.

4.3 IIR adaption test

This section makes direct filter order and noise resistance comparison between the LMS and ρRLS, since the results are almost identical, and the theoretical basis as already presented in the FIR section adaption section.

4.3.1 IIR system

Finite impulse response (FIR) filters can per definition only approximate a limited segment of the IIR's impulse response, having a limited number of taps. The lower order filters used for this test series are thus not expect to fit the system without error.

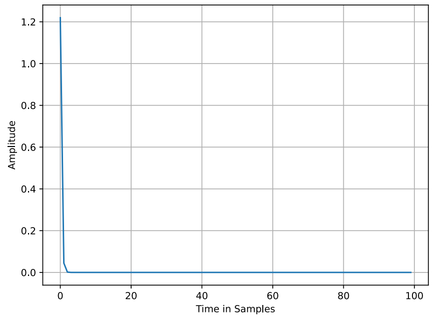

Figure 4.7: IIR impulse responce

The usual test revealing if a fixed-size FIR approximation is fruitful, takes the amplitude decay speed of the IIR system into consideration. If the impulse response envelope loses in amplitude, the approximation FIR order can be chosen to take the first N samples, where the envelope is above a threshold value. This method allows to truncate the IR, while keeping only the most relevant samples for the system output.

The displayed system impulse response is: [1.22, 4.461e-02, 1.632e-03, 5.971e-05, 2.184e-06, 7.993e-08, 2.924e-09, 1.069e-10].

The test IIR system's impulse response, as displayed in the adjacent figure and the vector above, drops very quickly towards zero, while the first sample is about 1.2. Thus, an FIR filter of a limited order could approximate it well. However, as mentioned in the introductory section, the adaptive filter don't fit the impulse response of the system but create an equivalent FIR system producing the same outputs.

4.3.2 Filter order impact

The observations formulated in the general section about the influences of LMS order and noise variance on the step-size and the ρRLS initialization matrix remain identical, being independent of the system to be fitted. As in section 4.2.1.1, a 60-point grid-search based on the maximum μ theorem is performed to avoid favoring a particular filter order with an arbitrary step-size. $\rho = 1$ and MI=10k were used, being the optimal values for noiseless stationary environments.

As for the identical FIR test, this MSE comparison is made with no additional noise in the input signal to test the true fitting capacity of the respective filter and their orders without interference. It is expected that higher filter orders approximate unknown system better, especially for IIR systems, having per definition an infinite impulse response, thus theoretically requiring an infinite amount of FIR filter taps for an error-free approximation.

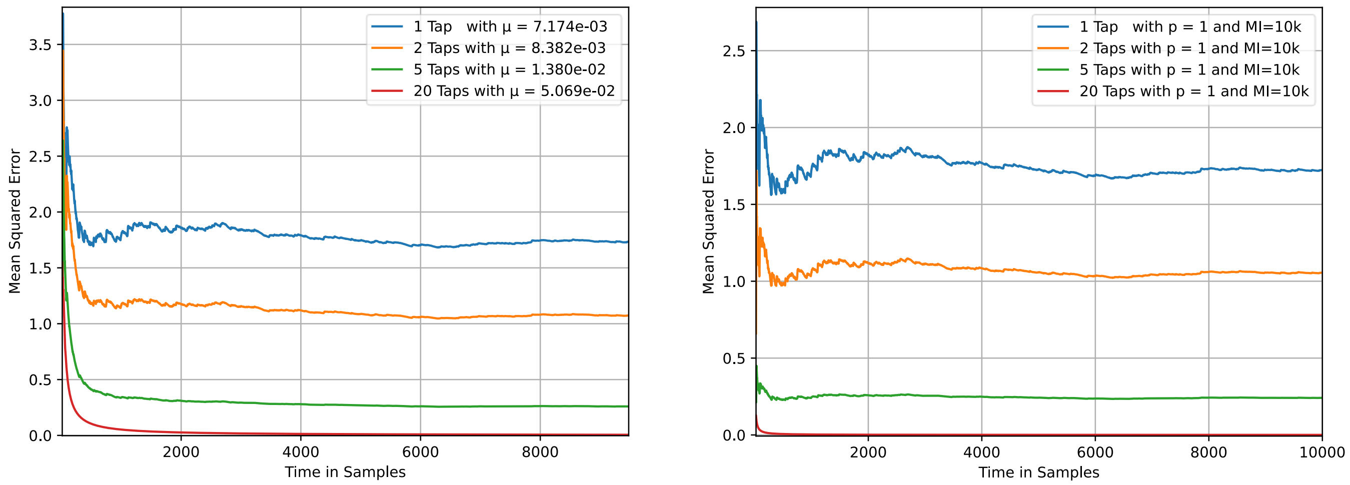

Figure 4.8: LMS and RLS: order vs MSE

As illustrated above, about 20 taps are required to arrive at an acceptable MSE of about 5.5e-03 after 10k iterations, which also converge the fastest, being the highest order filter. The lower order filters fluctuate more strongly than their FIR counterparts, while the 5 taps filter stabilizes only after 6k iterations. The 0th and 1st order filters show such strong variations that the averaging effect of the MSE doesn't smooth it out, meaning that the prediction error fluctuates strongly leading the filter weights to oscillate strongly. Furthermore, the asymptotic behavior flattens out with increasing order, as was already the case in the FIR study.

The fitting procedures of the filter aren't displayed, as they are only distinguishable from Figure 4.2 by their respectively higher absolute prediction error and MSE values.

Taking the first weights provided by the 20-tap approximation as the correct FIR weights, the value of the first common weights are pretty well approximated with differences only in the e-03 range.

4.3.3 Noise resistance

As for the FIR test, this IIR fitting MSE comparison is performed with optimal LMS step-size and ρRLS scalar-matrix initialization to limit their respective covariate effects on the tests of the respective orders' fitting capacity. The environment being static, the optimal forgetting factor of $\rho = 1$ was taken.

The below figure displays the ρRLS's MSE evolution in descending filter order with 5 taps being the left-most plot. The LMS's MSE curves weren't displayed, having identical evolution and elevation but with slightly more flattened initial asymptotic behavior. A direct comparison of the 4th order curves of both algorithms is represented by the below figure.

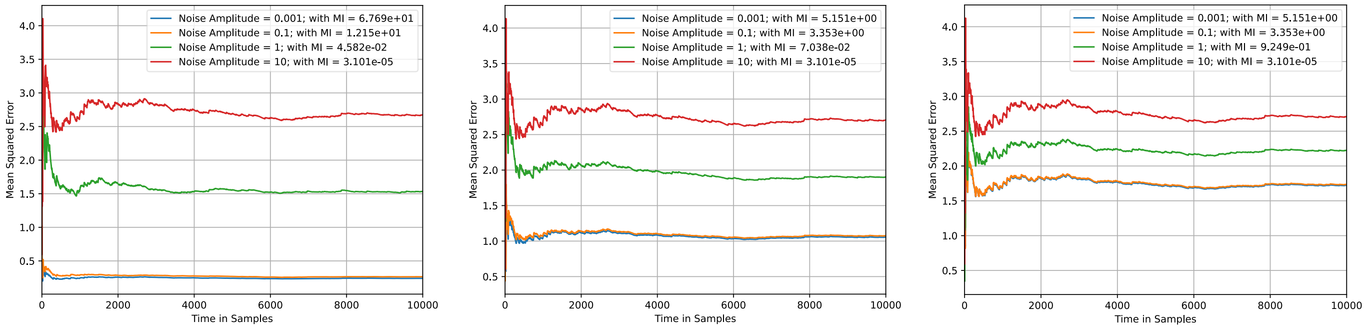

Figure 4.9: LMS and RLS: noise amplitude vs order

As for the FIR-fitting example, all NA=10 MSE curves oscillate strongly despite the MSE's average smoothing effect, indicating irregular weight updates instead of convergence and remain at the same height irrespective of the filter order. The lowest noise amplitude's 5-tap MSE curves no longer approach zero but converge to 0.25, while all other curves oscillate more strongly when gaining MSE. Unlike the FIR's MSE curves, the lower noise amplitude curves aren't only translated vertically but also progressively take the NA=10's shape, which might be due to their reduced distance to it compared to the FIR case.

The following figure compares the noise resistance of a 5-tap ρRLS and LMS filter with optimal hyper parameters and increasing additive noise amplitude.

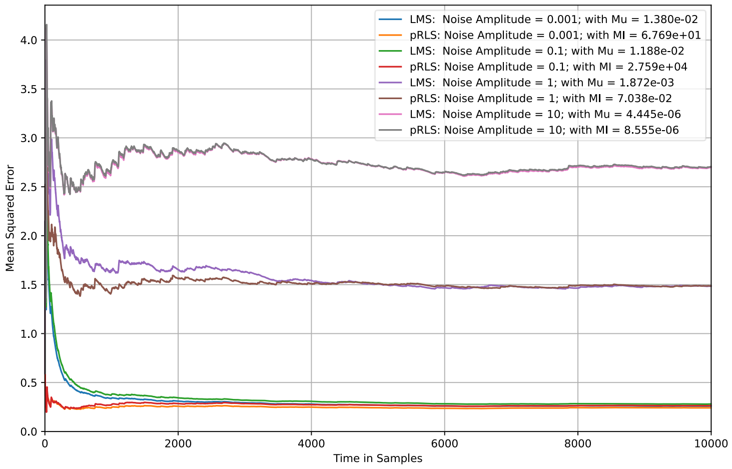

Figure 4.10: LMS and RLS: noise amplitude vs MSE

As expected from section 3 and (Haykin, 2014), the scalar matrix initialization values decrease with increasing SNR, while the optimal converging LMS step-size μ decreases with input power (Moschytz & Hofbauer, 2000). As for the FIR tests, Both NA=10 MSE curves are almost identical, landing around 2.7 for all orders and filters. The NA=1 RLS curve converges faster than the LMS's, however, both become similar after 4k iterations landing at 1.46 for the 5-tap filters. Also similarly to the FIR example, the ρRLS's NA=0.1 MSE curve converges faster than the LMS's both lowest noise levels, all ending up in the 0.25 region.

The final weights' results display identical trends to those of the FIR section. This section was therefore strongly abbreviated to avoid unnecessary redundancy and the comparison tables were omitted.

The two lowest noise variances provide the best fittings, yielding the test series' lowest MSE's and the closest weights to the 20-tap noiseless fitting from previous section, taken as solution for the fitting. The best approximations are performed by the highest order filters with the ρRLS mostly outperforming the LMS. The weights shrink with increasing noise levels, with NA=1 halving the weight amplitude and NA=10 resulting in weights two orders smaller than the correct values.

In conclusion, identical rules apply to FIR fitting than to IIR fitting: the noise level must be a magnitude order lower than the information containing input signal. Furthermore, the filter requires sufficient taps to perform proper fitting, making it more sensitive to noise, which can reveal problematic as potentially high orders are required to approximate an IIR adequately.

5 Non-stationary signals

This section analyzes the ρRLS's and LMS's adaption capacity to an abrupt system change after fitting for 5k samples. Identical tests to the previous sections are performed after some required clarifications about the goodness of fit metric.

5.1 Common aspects

5.1.1 MSE vs Absolute Prediction Error

In the following test series, the mean squared error (MSE), taken as metric until now must be carefully interpreted. As the name suggests, it is an average of **all** previous values, which for long time series has a very strong smoothing effect. Thus, even abrupt changes in prediction error and thus filter weights will not necessarily be visible in the MSE development. Furthermore, this test series takes 10k samples, which if used as audio would represent a 4.41th of a second. Thus, real word examples feature lengths superior by orders of magnitude resulting in a stronger smoothing.

This effect is illustrated by the two following figures:

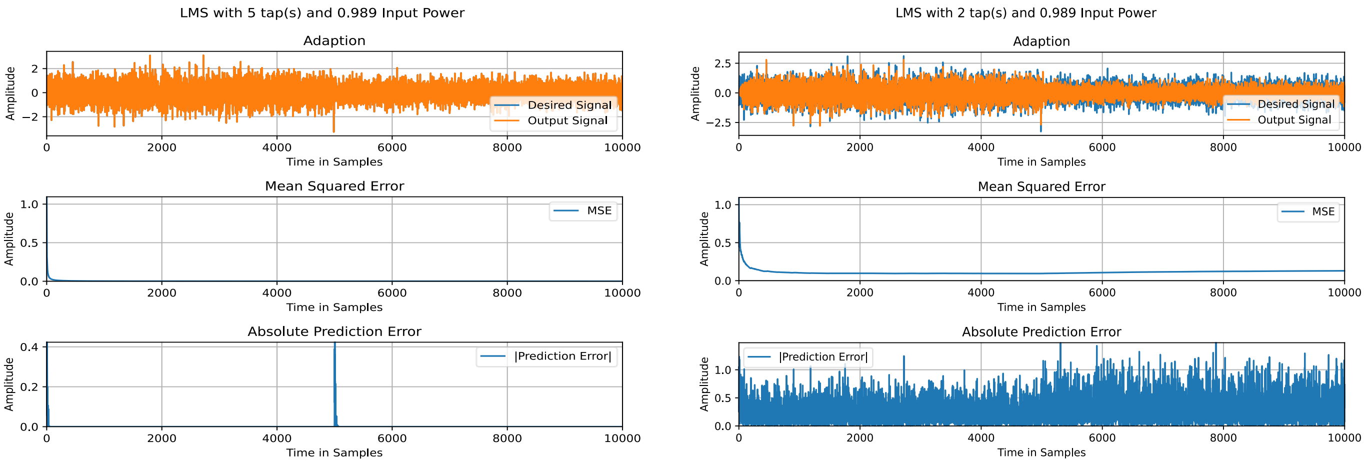

Figure 5.1: Mean Squared Error vs Absolute Prediction Error

Both figures represent the adaption process of a 4th order FIR filter, whose target system (4th order FIR) abruptly switches at the 5kth sample. As expected for a noiseless signal and optimal hyper parameters, the left filter having the right order achieves an almost zero (3.3e-04) MSE, while the first order FIR performs poorly (MSE=0.13).

Comparing the absolute prediction error to the MSE graphs, one notices that the spike in the middle of the left figure at the system change time isn't visible in the above MSE. This is due to the aforementioned smoothing effect of long averages, as the spike representing 50 values of 5050 gets flattened by the averaging process.

Likewise, in the right figure, the system change increases the absolute prediction error in a more visible manner than the MSE.

In further MSE plots, a vertical bar will visually signify the system change at 5k samples and the y-axis range will be reduced to increase MSE changes' visibility. This chops off the initial MSE peak height, which isn't of particular interest.

5.1.2 ρRLS: ρ value

This section requires the RLS to use the forgetting factor to react to the environment change. As seen in stationary signal section, where the grid-seach systematically chose the $\rho = 1$, reactivity to environment change comes at the cost of convergence and low MSE values.

The first grid-search tests all 4 noise levels on the FIR and IIR systems for 40 values linearly chosen between ρ = 0.8 to ρ = 1.0 with the default initialization of 1000 for the SkalarMatrix. The most used values were the two highest options (1.0 and 0.995).

Furthermore, since previous tests proved the importance of correctly initializing the scalar matrix, a further grid-search was performed to find a possible relation between the optimal forgetting parameter and the scalar matrix. The grid-search tested a logarithmic (base 2) range creating 30 values from e-04 to e+05 for the matrix initialization against a linear value series of 30 values between 0.9 and 1, resulting in 900 combinations per noise level for all of the 3 orders, thus 10 800 parameter combinations.
The results show that an optimal scalar Matrix initialization reduces the MSE as described in section 3.3 but has no effect on the forgetting factor, as the optimal ρ remains in {0.993, 0.996, 1.0} with the vast majority of cases being 0.996 and rare exceptions in 0.94. However, choosing a small ρ (around 0.94) reduces the MSE after a bad scalar matrix initialization.

Furthermore, the best scalar matrix initialization value exhibits convex behavior in the tested ranges (e-04 to e+05), meaning that a single (global) optimum seems to exist. This is ideal for optimization processes and grid-searches since the hyper-parameter search can follow the MSE reduction direction until the MSE rises again and be guaranteed that the optimal choice was reached.

5.2 FIR System change

The second system is an 4th order FIR with coefficients [0.4, -0.01, 0.3, -0.18, -0.2] replacing the initial of [0.7, 0.1 -0.03, 0.18, -0.24] after 5k iterations. Since the filters have half the time to converge, final MSE value is expected to be higher than in previous test.

5.2.1 Filter order impact

This section's tests series are identical to those in section 4.2, thus no additional noise is added, the LMS's step-size is defined by the same grid-search and higher order filter are expected to provide better approximations. The filter order's influence on the MSE evolution is tested with a supplementary system change at 5k samples, visually denoted with a vertical blue line. Thus, the ρRLS requires a variable forgetting factor ρ, since the environment is no longer static. To prove this effect, ρRLS order test is performed with $\rho = 1$ and with a grid-search choosing ρ from a linear 40 values series between 0.90 and 1. The scalar matrix is initialized with 10k as previously to allow a direct comparison.

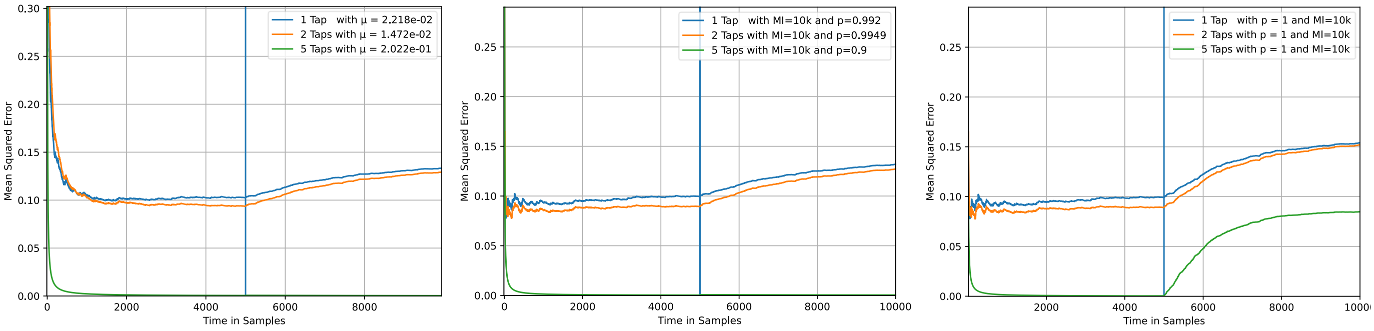

Figure 5.2: LMS and ρRLS and RLS: FIlter order vs MSE

As expected from previous tests, the MSE values are ranked in descending order for all filters. The 4th order LMS and ρRLS with $\rho \neq 1$ fit both systems with almost no error, while the $\rho = 1$ ρRLS rapidly gains in MSE. The MSE-jump of the $\rho = 1$ ρRLS's filters is due to the sudden system change destabilizing the filter, which tries to find a compromise between the old and new environment leading the very poor performance.

The three filters' lower orders all exhibit a similar MSE-jump at system change. The shallowest jump is performed by the ρ ≠ 1 ρRLS filter, followed by the LMS then the $\rho = 1$ ρRLS.

The 4th order $\rho \neq 1$ ρRLS filter and LMS yield the correct weights and respectively exhibit an MSE of 4e-04 and 3e-04, while the ρ = 1 ρRLS results in 0.085 MSE with deviations from the correct weights in the 0.1 magnitude order, for all orders. The first order LMS and variable ρ RLS filters divergence from the desired weights is in the e-03 range respectively yielding an MSE of 0.127 and 0.13. The 0th order variable ρ ρRLS outperforms the LMS by offering a divergence in the e-03 order compared to e-02 for the LMS.

Furthermore, the grid search chose different ρ values for the ρRLS depending on filter order. The 5-tap is optimal with the lowest possible ρ, forgetting the most possible data, while lower orders retain more data to optimize MSE. The two following plots illustrate the two different behaviors.

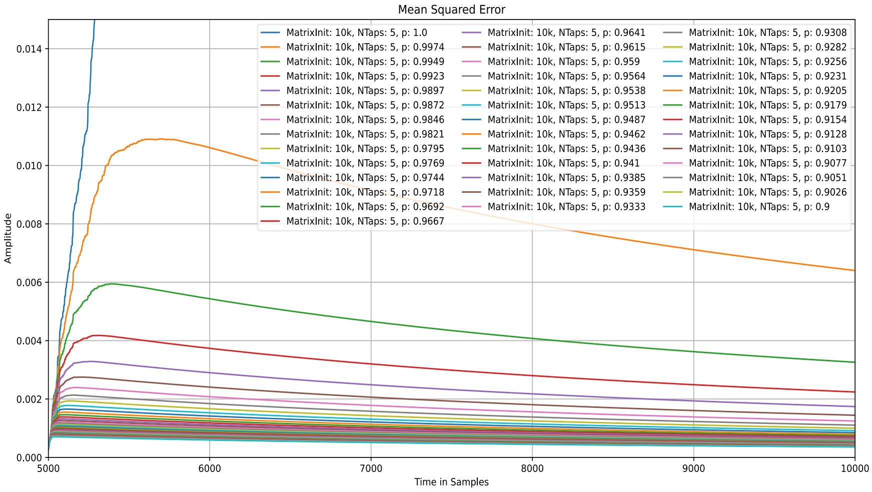

Figure 5.3: ρRLS: Minimal ρ choice

The 5 taps filter is the only one capable of approximating both given systems in an error-free way. As previous tests revealed, the ρRLS requires about 5 iterations to fit the unknown system of same order starting with a scalar matrix as autocorrelation matrix. The recursive autocorrelation matrix, however, once filled with values requires many samples to be completely overwritten by new data. Thus, as displayed by the above figure, lower data retention with low ρ values allow a faster adaption to the new system, as the MSE curves are classed in ascending ρ values.

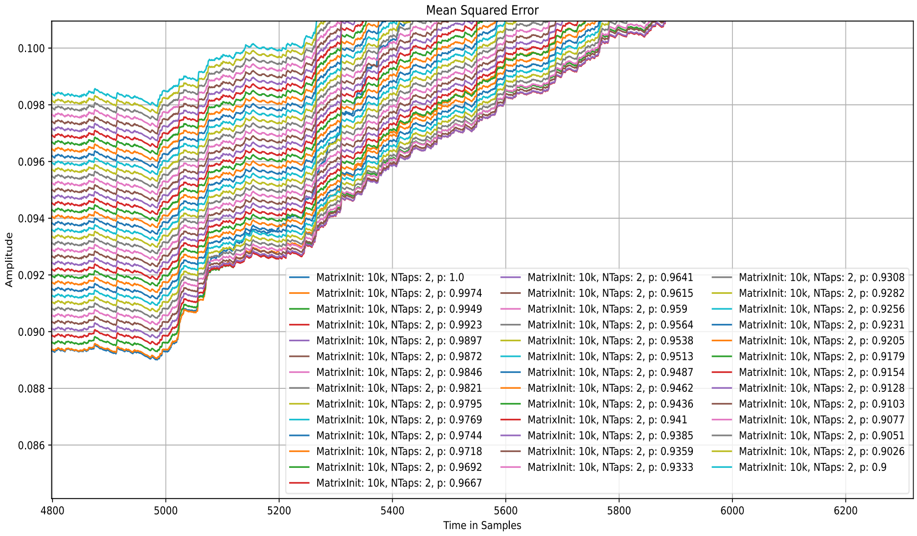

Figure 5.4: ρRLS: Maximal ρ choice

As illustrated above, all different ρ values' MSE curves remain parallel under 5k with the highest ρ values having the lowest MSE. Upon system change, the two highest ρ values, being the two lowest (blue and orange) MSE curves in the above figure, fail to adapt to the new system and rise quickly, while the other MSE lines remain relatively parallel to each other. The previously best forgetting factors thus become the two worst. The third initially best MSE also slightly rises but then remains parallel to the others and thus remains lower than them. Thus, due to their respective failure to adapt, the two initially best MSE curves leave their place to the third best, which adapts to the environment change.

5.2.2 Noise resistance

This section analyzes the influence of the usual amplitudes of additive noise $\{0.01, 0.1, 1, 10\}$ on diverse ρRLS and LMS filter orders. To reduce the covariate effect of the LMS's step-size μ, the ρRLS's forgetting factor ρ and scalar matrix initialization value, these were selected by grid-searches with respectively identical ranges as described in the stationary FIR test section, allowing direct comparison.

As in all previous sections, the higher filter orders (here 4th) are expected to outperform their lower order counterparts, as fitting systems with more degrees of freedom perform better approximations. Considering previous sections, filters are expected to become more noise sensitive while increasing their order, while the lowest noise levels are expected to yield better fittings.

The two following figures display respectively the 0th (left) and 4th (right) order ρRLS and LMS MSE evolution with different noise levels. The 1st order filters weren't plotted, exhibiting the identical rising MSE behavior following the system change as the 0th order filters.

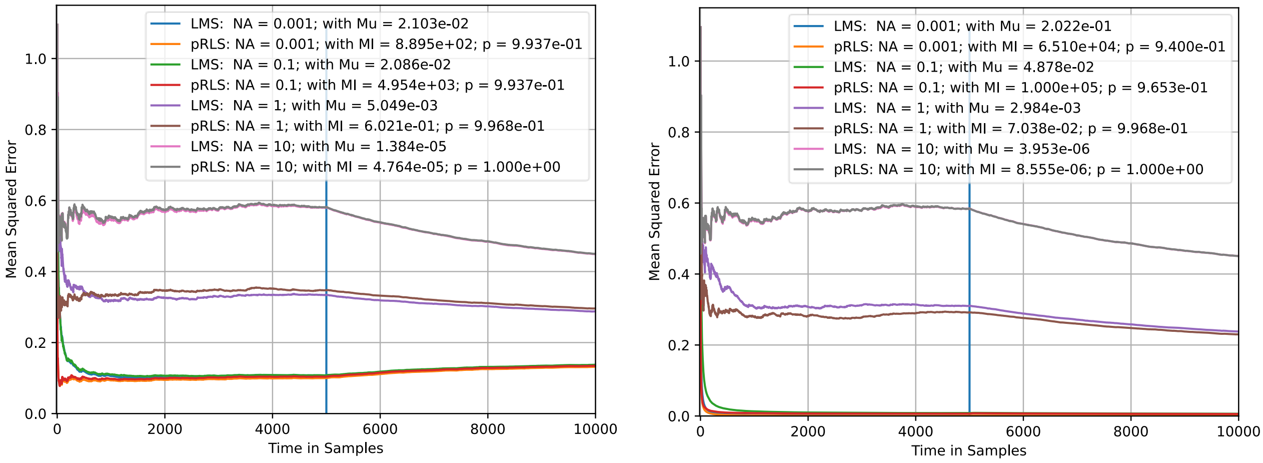

Figure 5.5: LMS and ρRLS order 0 and order 4: noise amplitude vs MSE

The above figure displays MSE reductions after system change for all MSE curves except the 0th order 0.001 and 0.1 noise levels. Thus, for the dropping curves, the second system might be more noise resistant or easier to approximate due to other reasons than the first one. A second possible explanations is that the respective hyper-parameters are selected based on the final MSE and thus favor fitting on the second system. Either way, the respective MSE values drop indicating that the absolute prediction error drops and thus also the amplitude of the changes in weight updates.

The rising MSE curves after system changes are known from previous section's zero-noise tests. Possible explanations are that the portion of the second filter's output amplitude is smaller in the first coefficients being [0.4, -0.01, 0.3, -0.18, -0.2] rather than [0.7, 0.1 -0.03, 0.18, -0.24]. Thus, both lower order filters generate more error, despite estimating the coefficient correctly.

Similarly to both stationary data tests (FIR and IIR), the maximum additive noise amplitude of the first system doesn't converge but fluctuates around for both algorithms. Upon system change, however, the initial asymptotical part of the convergence behavior starts but not enough data is available to allow it to flatten out.

Higher noise levels flatten the initial asymptotical behavior of convergence and as before, the ρRLS converges faster than the LMS at the same noise orders, except here for the NA=1 of the 0th order. As usual the NA=0.1 ρRLS converges faster than its LMS counterpart but doesn't yield a better final MSE. Otherwise all final MSE values of all noise amplitudes and orders are classed in increasing noise order with the ρRLS outperforming the respective LMS counterparts.

Identically to previous tests, the LMS step-size μ and the scalar matrix initial value decrease with increasing additive noise, respectively because of the input power and the signal to noise ratio. The forgetting factor ρ also rises with increasing noise levels and arrives at 1 for the maximal noise addition. This could be explained by the system "noticing" less and less the system change being progressively drowned in more noise.

The analysis of the final MSE and filter weights is similar to the stationary FIR and IIR sections and is therefore abbreviated to the core findings and the table is omitted presenting no new information.

The highest order filters and the lowest noise levels result in the best approximations with the ρRLS mostly outperforming the LMS. The weights shrink with increasing noise levels, with NA=1 halving the weight amplitude and NA=10 resulting in weights two orders smaller than the correct values.

In conclusion, both LMS et ρRLS with $\rho < 1$ respond adequately to abrupt and therefore progressive system changes if the noise level remains a magnitude order lower than the input signal's information containing part.

5.3 IIR System change

5.3.1 Filter order impact

This section's test procedures are identical to those presented in the stationary IIR and this FIR section on the system change data: The LMS's step-size and the ρRLS's forgetting factor are determined by the same grid searches and no noise is added. The scalar matrix is as usual initialized with 10k, allowing a direct comparison with all previous sections.

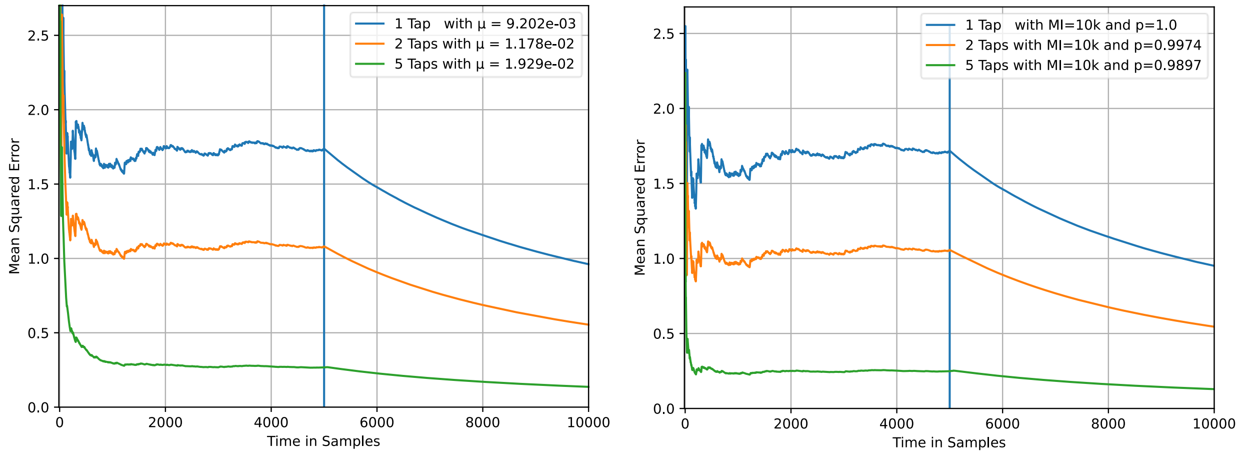

Figure 5.6: LMS and ρRLS Filter order vs MSE

As expected from previous tests, the MSE values appear in ascending filter order for both adaptive algorithms. In both cases, the second system seems easier to approximate, since the MSE falls asymptotically after the system change. Both plots are very similar, with the ρRLS having 0.01 lower MSE values than the LMS for every order.

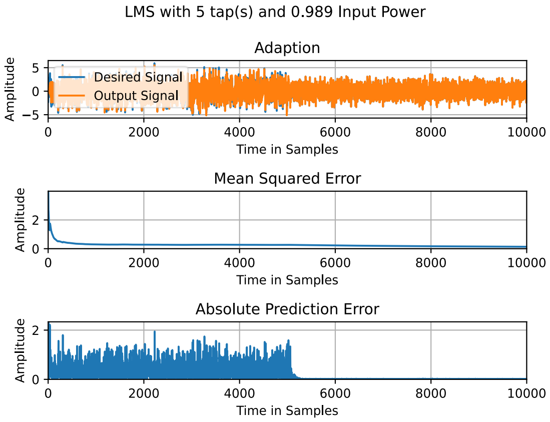

Figure 5.7: LMS: 5-tap adaption

A fitting process inspection gives further insight on the filters' convergence behavior. As seen in the drastic change in absolute prediction error mirrored in the decay of the MSE, the second system is approximated with almost no error by a 4th order FIR. Thus, either the unknown system is a FIR of same or smaller order or all other FIR and IIR weights are very small compared to the first five FIR coefficients. This system change illustrates well how the MSE's smoothing effect strongly falsifies the perceived convergence behavior (see conclusion).

The absolute prediction error of all adapted filters (other orders and ρRLS) display the same abrupt change, but the absolute prediction error of the second system is higher for the lower orders of both algorithms. From the absolute prediction error, one deduces that the weights fluctuate strongly rather than converge during the first system's adaption but converge very quickly upon system change.

The final weights given by both algorithms diverge from each other by an order of e-04 for the 5-tap filter. The ρRLS's lower order instances predicts weights very similar to the optimal ordered filter, while the LMS prediction worsens more strongly. Thus, the ρRLS estimation is more stable for lower orders in this example than the LMS's.

5.3.2 Noise Resistance

This section analyzes the influence of the usual amplitudes of additive noise $\{0.01, 0.1, 1, 10\}$ on diverse orders of the ρRLS and LMS algorithms fitting an IIR system abruptly changing at 5k samples. To reduce the covariate effect of the LMS's step-size μ, the ρRLS's forgetting factor ρ and scalar matrix initialization value, these were selected by grid-searches with respectively identical ranges as described in the stationary FIR test section, allowing direct comparison.

Based on previous sections, higher filter order and low noise environments are expected to be positively correlated with fitting performance, yielding MSE curves classed in ascending additive noise amplitudes. The below figure confirms those expectations by plotting the MSE curves of LMS and ρRLS filters with increasing added noise amplitude. The lower order filter graphs weren't displayed, being similar to this one, while showing the distortion type described in section 4.3.3 when reducing filter order.

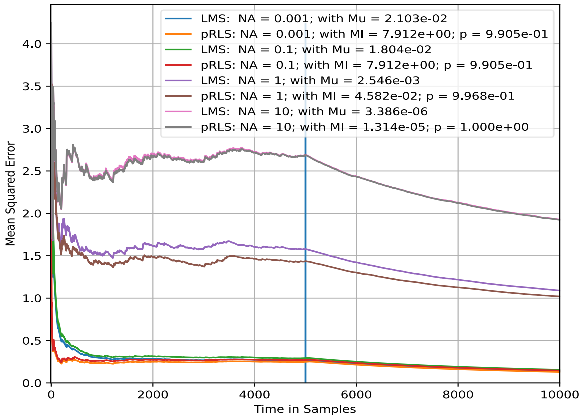

Figure 5.8: LMS and RLS noise amplitude vs MSE

Similarly to all previous sections, MSE curves over NA=0.1 don't converge but oscillate indicating that no convergence takes place but that the filter weights fluctuate around a particular value for the first system. The second system, which, as noted in the previous section, is either a FIR or a FIR-like IIR is fitted with marginal error by 5-tap filters. Thus, the slow value decline is due to the MSE's smearing effect and the correct MSE value would appear after many more iterations. Thus, there is a drop in absolute prediction error resulting in lower amplitude weight updates.

As in all previous tests, both lowest ρRLS filters converge faster than both equivalent LMS filters. In this test the NA=1 ρRLS clearly outperforms its LMS counterpart, while both NA=10 filters perform similarly.

The LMS's step-size and the ρRLS scalar matrix initialization value decrease with higher noise amplitudes as in previous tests, due to respectively the input power rising and the SNR decreasing. As aforementioned, the forgetting factor ρ also rises with noise, as the filters probably notice the system change less and use more input values to stabilize against noise.

The analysis of the final MSE and filter weights is similar to all previous sections and is therefore abbreviated to the core findings and the table is omitted presenting no new information.

Increasing the noise levels decreases the weight's values, with as usual the NA=1 weights being half the amplitude of the desired value and the NA=10 weights being two magnitude orders too small. Final MSE value-wise, higher filter orders are more sensitive to noise (Moschytz & Hofbauer, 2000), while being more precise for low noise environments. The highest order filters and the lowest noise levels result in the best approximations with the ρRLS mostly outperforming the LMS.

The conclusions of the stationary IIR and non-stationary FIR apply to this section.

6 Prediction

This section compares the LMS algorithm's and its kernelized version the KLMS's (kernel least mean squares) performance in future sample prediction. Using a delayed version of the input signal as desired signal allows to transform the filter adaption into prediction training. Both filters are fitted on a 500 sample training set to estimate the system to predict its future output based on previous outputs.

To prevent the KLMS from overfitting despite the use of Platt's novelty criterion, the grid search compares the filter performance on the validation set rather than the training set. This gaussian distribution-based KLMS implementation requires setting 5 hyper parameters as explained in section 2.2.4, one of which is the window-size filling this paper's two subsequent sections. The other four parameters are determined by a grid-search to find their respective optimal value. The step-size search is a linear spacing of 20 values in the [0.01; 2] interval, similarly to the kernel-size search, whose upper bound is however 20. Platt's novelty criterion's minimal distance and minimal error optimal values were respectively searched in [1e-07, 5e-07, 1e-06, 1e-05, 1e-04, 1e-03] and [1e-06, 1e-05, 5e-05, 1e-04, 5e-04]. The proposed grid-search is relatively coarse but contains already $20 \times 20 \times 6 \times 5 = 12\text{k}$ (times 2 accounting for both window-sizes) values, each taking 1:40h to compute on a 8core 4.3GHz machine despite all mentioned optimizations. Furthermore, a previous broader and coarser grid-search was performed, which allowed to tighten the respective ranges, leading to the current ones.

The LMS's step-size grid-search is a 40 values logarithmical series from the computed maximum μ to a 10th of it. This range was also deduced from previous coarser and broader grids.

The 1-sample-ahead prediction is performed by delaying the input signal by one sample and performing the usual filter application. For the LMS, this is a convolution with the input signal, while the KLMS applies its output formula $\langle a_{i-1}; K(\underline{x}[i], \mathcal{C}_{i-1})\rangle$ sample-wise to the input.

For unknown reasons, the LMS prediction requires a delay compensation of NTaps/5, while the KLMS requires no compensation at all.

6.1 Order 5

This section compares the prediction capacity of a 5-tap LMS with a KLMS using a window-size of 5, which in both cases corresponds to the number of previous values considered for the prediction.

The following figure displays both filter's adaption process on the provided training data with a direct output comparison with the resulting MSE and APE evolution.

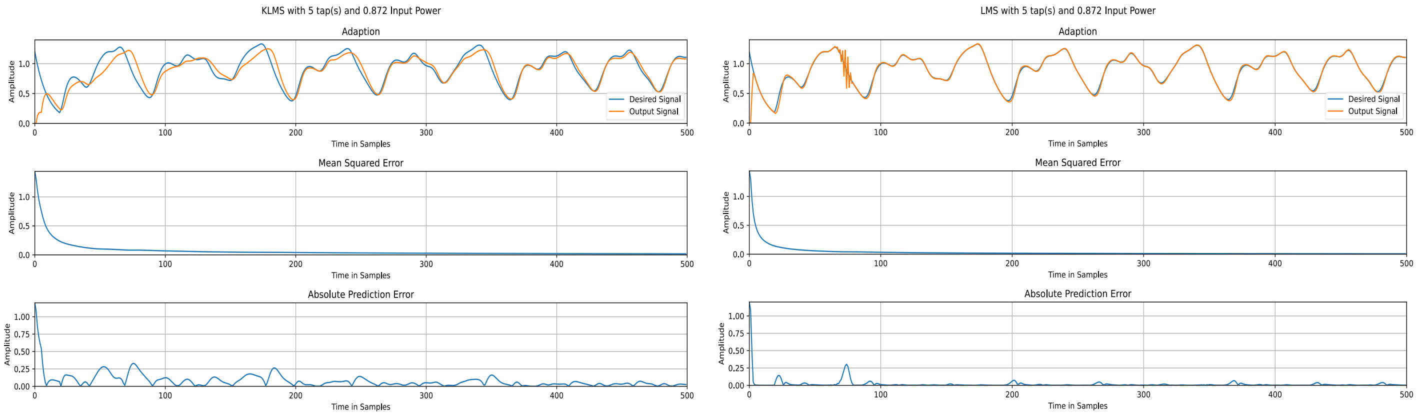

Figure 6.1: KLMS and LMS: 5 previous values adaption

The first KLMS prediction is zero, due to the implementation mimicking an empty sum for the first iteration. Similarly, the LMS's weight vector is initialized as zero vector also yielding a zero output at the first iteration. The KLMS fitting process is slow, as satisfying results are only obtained after about 380 iterations, while the LMS yields already good results after the first strong function decline at 90 samples. This observation is mirrored in the respective MSE's, as the LMS's displays a sharper asymptotical transient as the KLMS. The LMS training-set MSE is 3.55e-03, while the KLMS validation-set MSE is 2.95e-03.

The absolute prediction error (APE) is for this use a better metric than the MSE as the exact unpredictable spots are marked by peaks. The KLMS prediction error shows strong peaks throughout the training phase, which progressively decrease in amplitude, displaying the fitting process. The LMS's APE peaks when the desired output reaches its lowest point and rises again, denoting that the filter systematically mis-predicts those regions and that weight updates are performed.

The optimal step-size for the LMS is about 0.3414, while being about 0.2194 for the KLMS, both quite large compared to the previous system identification examples. The grid-search chose a Kernel-size of 1.0621, a minimal Euclidean distance of e-07 and a minimum error of e-04 for the novelty criterion. e-07 is the smallest available value, already resulting from previous grid-search, meaning that the KLMS essentially tries to deactivate this criterion by clipping it to the minimum at each search interval enlargement.

The following figure compares the LMS and KLMS predictions to the desired values. The range was restricted to thousand samples allowing a better visual prediction comparison. Considering that the signal is fairly periodic, not much information is lost.

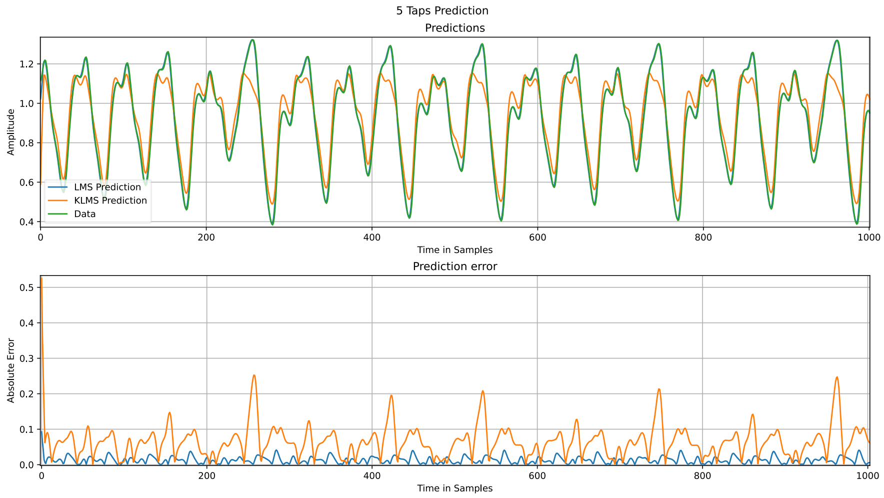

Figure 6.2: LMS and KLMS: 5 previous value prediction comparison

As illustrated above, the LMS's prediction is almost undistinguishable from the desired data with an APE peaking around 0.02. The exact total APE is 131.62 for 10k samples yielding an average of 0.013, representing about 1.46% of the desired signal's amplitude. The weights are [0.67722, 0.22991, 0.02313, 0.0185, 0.0532].

The KLMS, however, produces visibly inaccurate predictions with the APE peaking around 0.25, as it tends to chops off peaks. The total APE is 672.05, which is 5 times superior to the LMS's APE, yielding an average of 0.0672 APE representing 7.4% average prediction error.

In conclusion, not only is the LMS more efficient to fit, having a lower algorithmic complexity and fewer hyper-parameters to search (here 40 against 12k), it produces more accurate predictions for this system. Thus, for this use case, the LMS should be preferred, however, some strongly non-linear systems might require using the KLMS.

6.2 Order 10

This section compares the prediction capacity of a 10-tap LMS with a KLMS using a window-size of 10, corresponding to the number of previous values considered for the prediction.

Following figure displays the KLMS's and LMS's adaption process, providing insight in their respective output signal, MSE and APE.

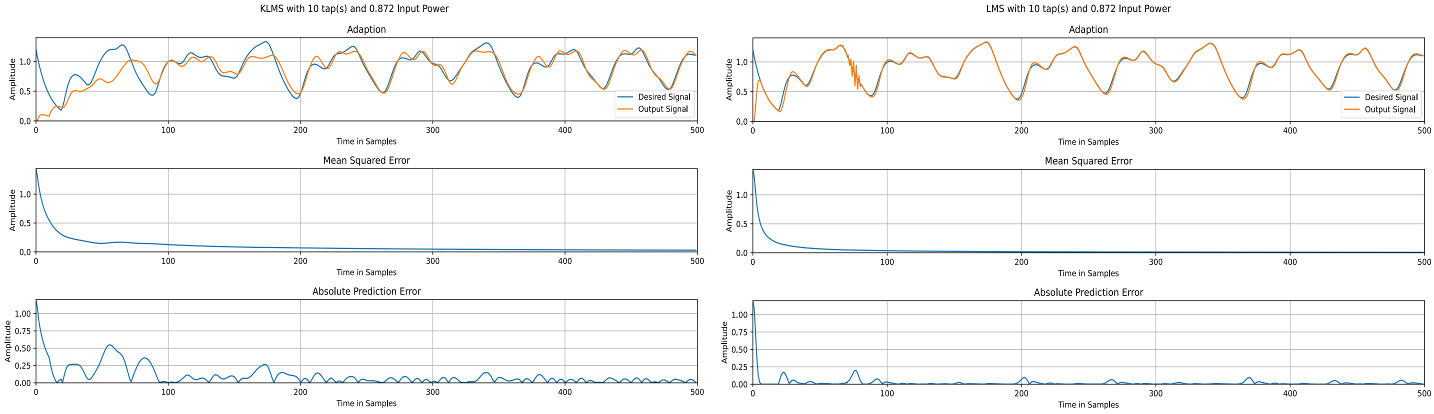

Figure 6.3: KLMS and LMS: 10 previous values adaption

Similarly to the previous 5-tap test, the first predictions of both filters is zero, then progressively rises to fit the input signal. The KLMS requires again most of the training samples to provide a reliable fitting, while the LMS converges after 90 samples. Both fittings resemble strongly their 5-tap counterparts.

The LMS's irregularity is slightly higher on the curve (= earlier) and the smaller peaks are lower in amplitude, while the filter generally converges marginally faster than the 5-tap counterpart. At each APE peak the LMS's weights are updated, resulting in a final weight vector of [0.4636, 0.23386, 0.07334, -0.0047, -0.0227, -0.00933, 0.01782, 0.05098, 0.08643, 0.1217], which surprisingly doesn't resemble the 5-tap LMS weight coefficient, except for the second coefficient.

The 10-tap KLMS, however converges slower and yields more APE and MSE error in general. Its APE displays more and smaller peaks than its 5-tap counterpart. The LMS's MSE converges faster than the KLMS's, each respectively landing at 4.8e-03 (LMS training-set error) and 3.53e-03 (KLMS validation-set error).

The optimal step-size for the LMS is 0.171, while being 0.115 for the KLMS, being respectively half the values of their 5-tap counterparts. The grid-search chose the same kernel-size (1.062), minimal Euclidean distance (e-07) as before, while reducing the minimum error from e-04 to e-06. Since both those values are at their minimum, one can deduce that the grid-search tries to deactivate the novelty criterion to add more samples to the KLMS's dictionary to increase fitting precision.

The following figure compares the LMS and KLMS predictions to the desired values in the limited range of thousand samples.

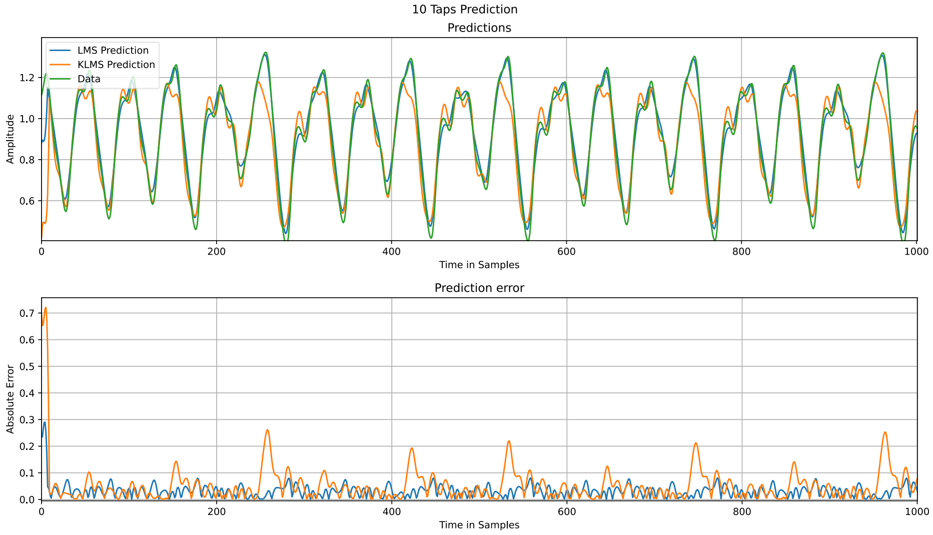

Figure 6.4: LMS and KLMS: 10 previous value prediction comparison

As illustrated above, the LMS once more outperforms the KLMS, while yielding less accurate predictions than the 5-tap LMS. The LMS's APE peaks often around 0.07, while being on average 0.031 representing 3.4% of the signal amplitude, which is more than double the 5-tap APE. The KLMS's APE peaks often around 0.2, with 0.051 average representing 5.7%, which is an improvement over the 5-tap KLMS but is still worse than both LMS filters.

Similarly to the previous conclusion, the LMS should be preferred to the KLMS being more efficient to predict, having fewer hyper-parameters and yielding better predictions. However, increasing the LMS order doesn't necessarily improves the prediction, while the contrary holds for the KLMS.

7 Methodology critique and further work

The first generalized critique on this paper is that each filter is fitted with a single noise addition. It would have been more representative to generate hundreds or thousands of different noise signals and to fit filters on them after a grid-search. A single grid-search would be sufficient, as hyper-parameters should stay identical being independent of the exact random series if the characteristics like amplitude and mean remain unchanged. Limited computing power prohibited this, as some of the tests already required 1:40h to proceed with a single noise addition.

Furthermore, the LMS's and ρRLS's weight vectors, as recommended by Moschytz and Hofbauer (2000), get initialized as zero vector. Further test could determine better initialization methods like for example random numbers with the input power as variance.

7.1 LMS

Further modifications and optimizations of the LMS algorithm exists, which can be implemented with minor modifications like the NLMS and the Newton-LMS (Moschytz & Hofbauer, 2000).

Furthermore, the step-size should be made variable of two components, the input level (as done in the LMSconverge function and in the NLMS algorithm) and also become smaller when converging, such that the filter approaches rapidly the optimal coefficients then slows down to allow more precise convergence. The step-size reduction criterion shouldn't be based on the iteration number, as systems could change over time or involve noise levels slowing down conversion. It should instead be based on the summed or averaged absolute prediction error (APE), directly measuring the error and the weight changes. Thus, when the weights start stabilizing, the step-size should decrease to increase convergence precision.

7.2 ρRLS

One drawback of the ρRLS is the complexity in $O(n^2)$, which can be overcome by using the F-RLS which is $O(n)$, while delivering similar results (Moschytz & Hofbauer, 2000).

7.3 KLMS

The KLMS algorithm has the highest complexity of the three tested algorithms despite all mentioned optimizations. The minimum distance threshold of Platt's novelty criterion requiring the computation of the distance to all stored points could be deleted, as all of this paper's grid-searches essentially cancelled it. This would reduce the algorithms complexity by $O(i)$, with i being the iteration number. Alternatively, other dictionary sparsification methods can be tried.

Furthermore, the filter() function can be vectorized, by constructing a matrix out of the to be processed input windows and performing matrix-vector multiplications rather than scalar products in a sequential for loop. This can only be done with the $f(\cdot)$ function in filter, since the weight vector mustn't be updated, forcing sequential computations.

7.4 Grid-search

The grid-search function should be extended such that upon finding the best hyper-parameter combination it triggers a second grid-search in the vicinity of the best hyper-parameters. The grid-size is thus strongly reduced while keeping the same points amount, leading to a more refined search. To illustrate, upon finding the best hyper-parameter value, the two surrounding grid values can be taken as upper and lower bound for the new grid search, which is done in every parameter space dimension. This procedure's repeated application guarantees convergence towards the found hyper-parameter space local minimum, with no guarantee of it being a global minimum.

Furthermore, the grid-search should take a boolean list as input informing about hyper-parameter convexity, allowing it to extend the range until the error metric (average APE or MSE) stops decreasing.

Finally the KLMS would benefit from a real n-fold cross-validation, splitting the training data into n subsets and using one of them as test data and use the test-set as final validation to prevent overfitting on the validation set.

7.5 MSE

The mean squared error, as mentioned numerous times throughout the paper, strongly falsifies the error metric and is therefore prone to amelioration.

Firstly, squaring the error unnecessarily exaggerates the metric, as values < 1 get marginalized, while errors > 1 get exaggerated. This is problematic, since audio data is per convention in the $[-1; 1]$ range with most of the waveform samples smaller than the signal's amplitude. To illustrate, a amplitude 1 sinus has at most one sample respectively at 1 and -1 pro period, while arbitrarily many, depending on sampling and frequency, are smaller. Thus, the size 2 range has less than 50% chances to yield a difference greater than 1, skewing the metric downwards.

As aforementioned, MSE's smoothing effect due to averaging prohibits detailed analysis, glossing over peaks and smearing out sudden prediction error changes as illustrated in Figure 5.7. The MSE's long averaging process also makes it inadequate for grid-searches, favoring a quickly converging filter landing at higher prediction errors than a very slowly converging filter with a smaller step size. The latter will have higher prediction values on average, even if lower values are reached towards the end. Furthermore, this paper's test series were relatively short in audio standards (10k samples correspond to 0.23 seconds at 44.1kHz) and that thus the smoothing effects are tremendously stronger than analyzed here.

The proposed solution is a moving average or moving median-based absolute prediction error (MAAPE). The absolute function replacing the square doesn't exaggerate the error in any direction, while the moving average or median allows limited data smoothing, generating readable graphs while preserving the temporal structure to account for sudden changes. The MSE is easily computed in real time during the iteration, being a multiply-add procedure. If real time moving average computation is necessary, a round-robin array of the desired averaging length can track the sample arrival order to subtract the oldest sample from the sum and add the new arrival after scaling it. This efficiently updates the average without requiring a scalar product (weighted sum) to be unnecessarily recomputed at each iteration.

8 Conclusion

The LMS and ρRLS algorithms adequately fit filters if enough taps are provided (especially important for fitting IIR's) and if the additive noise is lower in amplitude than the signal containing the system to be identified. Environment variations can be accounted for by the LMS and ρRLS, while the KLMS presumably requires specific clearing mechanisms outside of this paper's scope. Higher order filters perform better fitting, having more degrees of freedom, while setting the unnecessary weights to zero if required.

The LMS’s prediction capacity outperforms the KLMS’s in the tested example, but the KLMS’s higher complexity might be necessary for strongly non-linear systems. Contrary to system fitting, the use of higher orders don’t necessarily result in better predictions for the LMS while the KLMS benefits from larger window-sizes.

All this paper’s results confirm the importance of the right hyper-parameter choice, which should ideally be automatized by grid-searches.

9 Bibliography

Haykin, S. (2014). *Adaptive Filter Theory, Fith edition.* Essex: Pearson.

Liu, W., Príncipe, J. C., & Haykin, S. (2010). *Kernel Adaptive Filtering A Comprehensive Introduction (Adaptive and Learning Systems for Signal Processing, Communications and Control Series).* (S. Haykin, Hrsg.) Hoboken, New Jersey: JOHN WILEY & SONS, INC., PUBLICATION.

Moschytz, G., & Hofbauer, M. (2000). *Adaptive Filter: Eine Einführung in die Theorie mit Aufgaben.* Springer, 2000.