SIMULATION OF RLS AND LMS ALGORITHMS FOR ADAPTIVE NOISE CANCELLATION IN MATLAB.

Ondračka J., Oravec R., Kadlec J., Cocherová E.
Department of Radioelectronics, FEI STU Bratislava, Slovak Republic
UTIA, CAS Praha, Czech Republic

***Abstract**:*

The main goal of this article is to describe different algorithms of adaptive filtering, mainly the RLS and LMS algorithm, to perform simulation these algorithms in MATLAB - SIMULINK and finally, compare these algorithms.

DESCRIPTION:

To compare the RLS and LMS algorithms we utilised and improved the existing functional scheme from MATLAB, precisely the scheme of RLS and LMS algorithms for adaptive noise cancellation, as is shown in the Figures 2-4. The subfigure in the Fig.2 stayed without changes, while the internal parts of schemes of RLS adaptive filters (Fig. 4, on the left) and of LMS adaptive filters (Fig. 4, on the left) we changed radically. The all scheme, as is shown in the Fig. 4, is represented by one block, i.e. the block of matlab-function. Since every matlab-function has only one input, we insert a multiplexer, which all the input signals collects to the one vector.

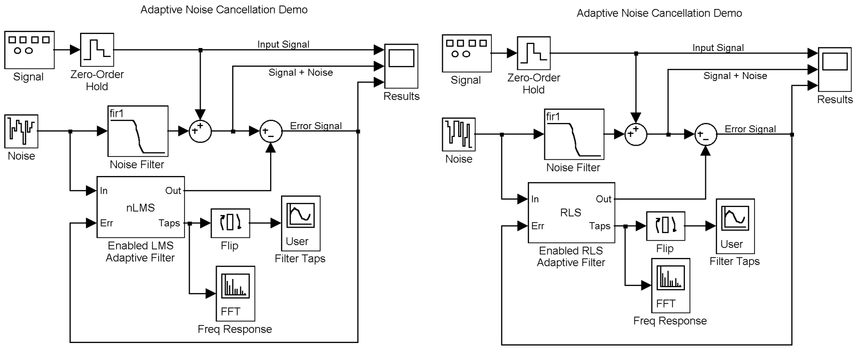

Fig. 1. Block diagrams of noise cancellation
LMS algorithm (on left) and RLS algorithm (on right)

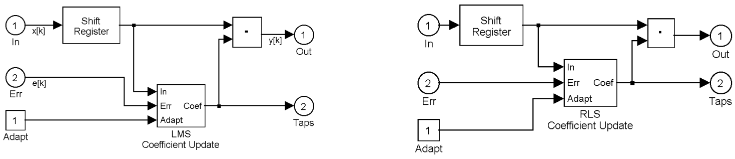

Fig. 2. Sub-scheme of adaptive filters LMS (on left) and RLS (on right)

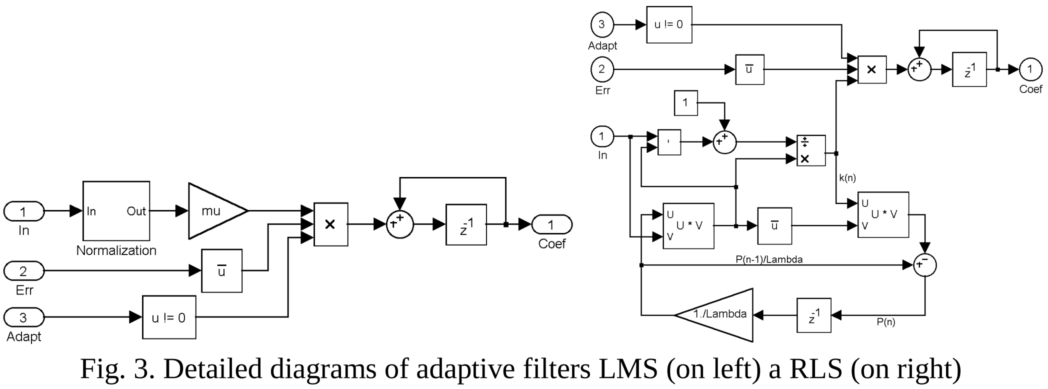

Fig. 3. Detailed diagrams of adaptive filters LMS (on left) a RLS (on right)

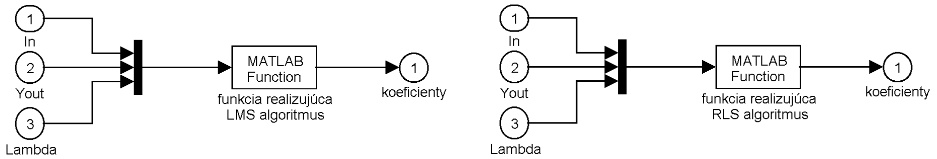

Fig. 4. Adaptive LMS filter (on left), adaptive RLS filter (on right)

We based the algorithms development on the equations (1) and (2), which describe the evaluation of filter coefficients in following way:

$$[X_{uu}(k+1)]^{-1} = [X_{uu}(k)]^{-1} - \frac{[X_{uu}(k)]^{-1}\mathbf{u}_{k+1}\mathbf{u}_{k+1}^{T}[X_{uu}(k)]^{-1}}{1+\mathbf{u}_{k+1}^{T}[X_{uu}(k)]^{-1}\mathbf{u}_{k+1}} \tag{1}$$

$$\mathbf{w}_{LS}(k+1) = \mathbf{w}_{LS}(k) + \underbrace{[X_{uu}(k+1)]^{-1}\mathbf{u}_{k+1}}_{Kalman\ vektor\ \mathbf{k}_{k+1}} \cdot \underbrace{\left(d_{k+1} - \mathbf{u}_{k+1}^{T}\mathbf{w}_{LS}(k)\right)}_{\mathrm{Pr}\,ioritn\acute{y}\ zvy\check{s}ok\ \varepsilon_{k+1}} \tag{2}$$

These relations we rewrite to the form of m-file as a file *rls1.m* :

```
1  function y=rls1 (u);
2  global Wo Wn Wm Po Pn Pm;
3  [m,n]=size(u);
4  uin=u(1:m-2,1);
5  yout=u(m-1,1);
6  lambda=u(m,1);
7  Wo=Wm;
8  e= yout-uin'*Wo;
9  Po=(1/lambda)*Pm;
10 Pn=Po - ((Po*uin)*(uin'*Po))/(1 + uin'*Po*uin);
11 Wn=Wo+((Pn*uin)*e);
12 y=Wn;
13 Pm=Pn;
14 Wm=Wn;
```

In the first line we define the *rls1* function, which has one input *u* and one output *y*. In the second line we declare auxiliary matrixes *P* and vectors *W*. In the third line we ascertain size of input data. In the lines 4, 5 and 6 we divide input data to appropriate vectors *uin* = $\mathbf{u}_{k+1}$, *yout* = d_{k+1} and *lambda*. The constant *lambda* is auxiliary constant (neglecting factor). In the line 7 we write to the auxiliary vector of weight Wo the weights from former time step. In the following line we evaluate inaccuracy of prediction, i.e., as it is termed in Eq. 2, the prior remainder, according the relation: $(d_{k+1}–\mathbf{u}_{k+1}^{T}\mathbf{w}_{LS}(k))$. In line 9 we set down covariance matrix from previous time step into matrix *P*o. Next line represents the actualisation of covariance matrix according Eq. 1. In the line 11 we actualise the weights of adaptive filter. Actualised weights are inscribed to the output of function in the line 12. In the last two lines we take down the covariance matrix and weight vector for the use in the following time step.

An establishing of m-file for LMS algorithms arises from following equation 3, which describes actualisation of filter coefficients:

$$\mathbf{w}_{LMS}(k+1) = \mathbf{w}_{LMS}(k) + \mu \cdot \mathbf{u}_{k+1} \cdot \left(d_{k+1} - \mathbf{u}_{k+1}^{T}\mathbf{w}_{LMS}(k)\right) \tag{3}$$

This relation we rewrite to the file *lms1.m*:

```
1  function y=lms1(u);
2  global Vo Vn Vm;
3  [m,n]=size(u);
4  uin=u(1:m-2,1);
5  yout=u(m-1,1);
6  lambda=u(m,1);
7  Vo=Vm;
8  e= yout-uin'*Vo;
9  mi=0.001*lambda;
10 Vn=Vo+(mi*uin*e);
11 y=Vn;
12 Vm=Vn;
```

Resemble to the previous *rls1* file, we define the *lms1* function in the first line. This function has one input *u* and one output *y*. In the second line we declare auxiliary vectors *V* and in the third line we find out size of input data. In the lines 4, 5 and 6 we divide input data to appropriate vectors *uin* = $\mathbf{u}_{k+1}$, *yout* = d_{k+1} and *lambda* (*lambda* is auxiliary constant for external regulation of the step parameter μ). In the line 7 we write the weights from former time step to the auxiliary vector of weights Vo. In the following line we evaluate inaccuracy of prediction according appropriate part of Eq. 3, what is: (d_{k+1}–$\mathbf{u}_{k+1}^{T}\mathbf{w}_{LS}(k)$). In line 9 we make product of step parameter $\mu = 0.005$ and auxiliary constant *lambda* (there is no external influence, as *lambda* = 1). Next line represents the actualisation of covariance matrix according Eq. 3. In the line 11 actualised weights of adaptive filter are inscribed to the output of function. In the last line we take down the weights vector for the use in the following time step.

To work properly from the first time step, we must initialise all used matrixes and vectors. For this goal we created next matlab-file *init.m*, which we must run every time before first running of simulation, to set all appropriate matrixes and vectors to their starting value.

```
function init;
global Vo Vn Vm Wo Wn Wm Po Pn Pm;
k=32;
d=1000;
Po=d*eye(k,k);
Pn=d*eye(k,k);
Pm=d*eye(k,k);
Wo=zeros(k,1);
Wn=zeros(k,1);
Wm=zeros(k,1);
Vo=zeros(k,1);
Vn=zeros(k,1);
Vm=zeros(k,1);
```

RESULTS

In this section, we compare RLS and LMS adaptive filters. These comparisons are worked out in the form of figures, which show input and output signals from filters, next we compare amplitudes of actualisation coefficients and transmission characteristics of RLS and LMS adaptive filters.

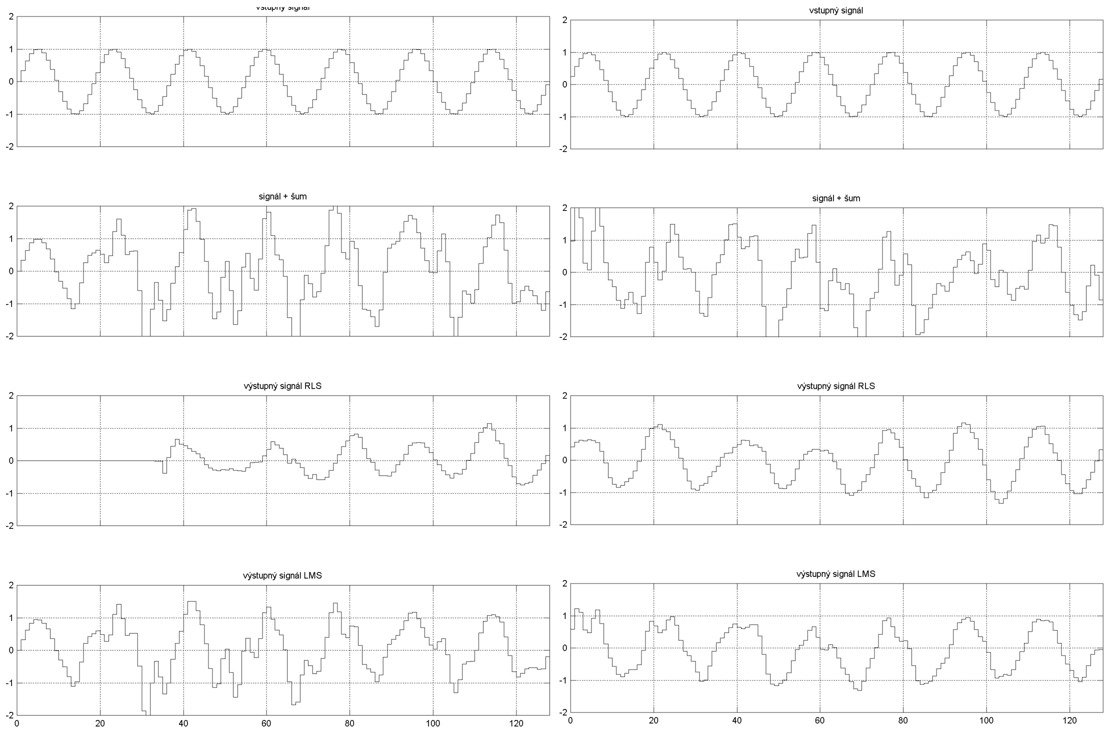

Fig. 5. Time courses of simulated output signal (two lower traces) for sinusoidal input signal (upper trace) after 128 (on left) and 256 (on right) evaluation steps, are shown in the figure on the left or right side, respectively. (There are shown only the final 128 evaluation steps.) The second trace from the top is the input signal with added noise.

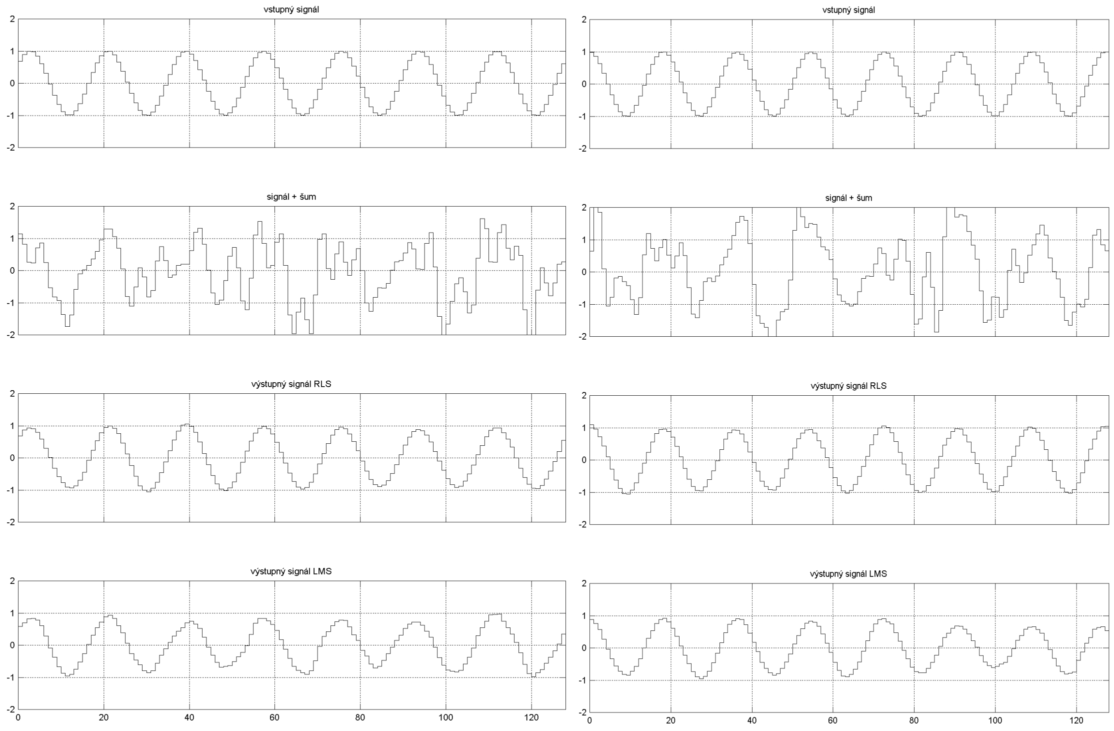

Fig. 6. Time courses of simulated output signal (two lower traces) for sinusoidal input signal (upper trace) after 512 (on left) and 1024 (on right) evaluation steps, are shown in the figure on the left or right side, respectively. (There are shown only the final 128 evaluation steps.) The second trace from the top is the input signal with added noise.

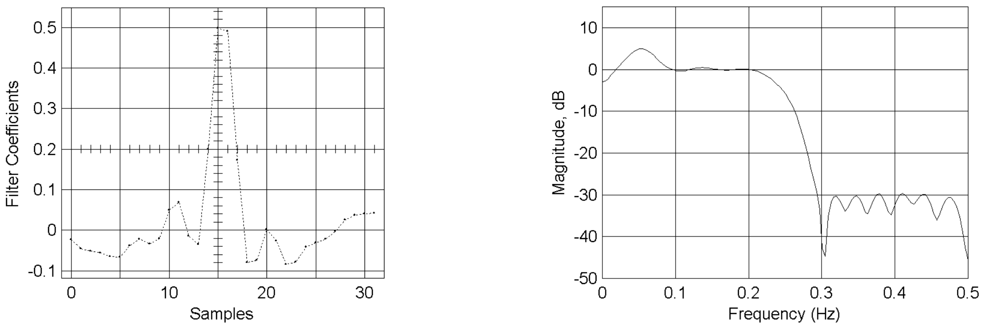

Fig. 7. Amplitudes of actualised filter coefficients and magnitude of transmission characteristic of LMS adaptive filter after 1024 evaluation steps

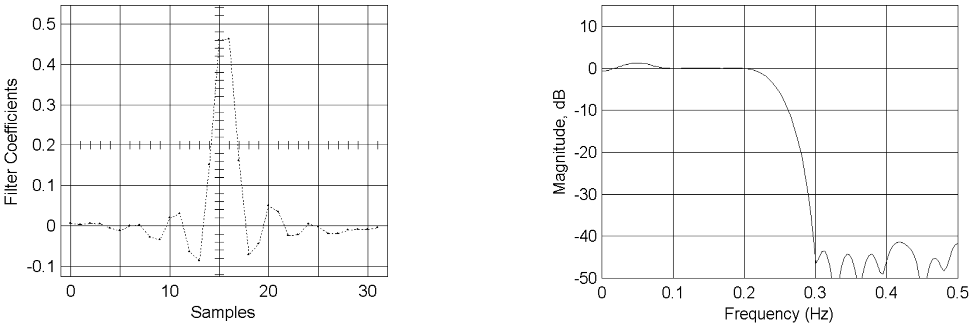

Fig. 8. Amplitudes of actualised filter coefficients and magnitude of transmission characteristic of RLS adaptive filter after 1024 evaluation steps

CONCLUSION

From the simulation of RLS and LMS filters we have found, that the adaptation rate of both filters was nearly equal; these algorithms have adapted approximately after 200 evaluation steps for the sinusoidal harmonic input signal. The quality of disturbance and noise cancellation is more evident after 1000 actualised steps, when we can observe also some differences in transmission characteristics. From observations implies, that RLS adaptive filters give higher quality in the disturbance and noise cancellation.

From mentioned comparisons implies that LMS algorithms are simpler in evaluation process, but they attain lower quality in the cancellation of disturbing signals. Contrary, RLS algorithms achieve higher quality in the disturbing signal cancellation, but they have large numerical claims for RLS filter coefficient evaluation.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Education of the Slovak Republic under Grant VTP 102 “Digital Processing of Audio, Video and Biomedical Signals“, the Texas Instruments company and partially supported by project OK317 of the Ministry of Education and Sports of the Czech republic.

REFERENCES

[1] Moonen, M.: Introduction to adaptive signal processing; Leuven, Belgium; 1995
[2] Uhlíř, J. - Sovka, P.: Číslicové spracováni signálu, ČVUT Praha, 1993
[3] Rozinaj, G. a kol: Číslicové spracovanie signálov, Košice, 1990
[4] Ondráček, O. a kol: Signály a sústavy, STU Bratislava, 1998
[5] IEEE Signal processing magazine, July, 1999
[6] International journal of adaptive control and signal processing, Vol. 13, pp. 487 – 505, 1999

Department of Radioelectronics, FEI STU, Ilkovičova 3, 812 19 Bratislava, Slovak Republic
Tel: +421 7 6029 1532, Fax: +421 7 6542 9683
E-mail: cocherov@elf.stuba.sk, kadlec@utia.cas.cz, oravecra@hotmail.com, ondracka@pobox.sk