# A Novel Forgetting Factor Recursive Least Square Algorithm Applied to the Human Motion Analysis

Hadi Sadoghi Yazdi[1,2], Mehri Sadoghi Yazdi[3], Mohammad Reza Mohammadi[4]

***Abstract*****—**This paper is concerned with studying the forgetting factor of the recursive least square (RLS). A new dynamic forgetting factor (DFF) for RLS algorithm is presented. The proposed DFF-RLS is compared to other methods. Better performance at convergence and tracking of noisy chirp sinusoid is achieved. The control of the forgetting factor at DFF-RLS is based on the gradient of inverse correlation matrix. Compared with the gradient of mean square error algorithm, the proposed approach provides faster tracking and smaller mean square error. In low signal-to-noise ratios, the performance of the proposed method is superior to other approaches.

***Keywords*****—**Forgetting factor, RLS, Inverse correlation matrix, human motion analysis.

## I. Introduction

THE adaptive filtering was studied in several many applications. The recursive least square (RLS) adaptive filter is a well known method for its superiority to the least mean square (LMS) method in both convergence rate and misadjustment [1]. In general, RLS algorithms do not impose any restrictions on the input data structure. As a consequence of this generality, the computational complexity is $O(N^2)$ per time iteration, where $N$ is the size of input data array. Briefly, good convergence and small mean square error (MSE) in stationary environments are typical properties of RLS method. Its good properties have been directed to development of modified or extended RLS algorithms in time-varying systems [2]-[7]. However, RLS using constant forgetting factor (CFF) cannot provide satisfactory performance in time-varying environments.

Many attempts have been made to moderate the RLS algorithm to use in time-varying environments. One modification uses a data weighting window on the input data sequence [12], [13] to adjust the effective memory of the algorithm. However, it is not easy to adjust changes of the window. Another method is to agitate the covariance matrix whenever the change is detected [6], [5]. This approach has the benefit of reenergizing the adaptation gain. However, the algorithm becomes sensitive to disturbance and noise because of the nonzero adaptation gain.

Another method for increasing performance of the RLS algorithms at non-stationary environments is to control the forgetting factor or the effective data window length [8]-[11], [14]. This approach is suitable for achieving better convergence and tracking properties. A new variable forgetting factor, gradient based VFF-RLS algorithm or GVFF-RLS is introduced in [8], [14]. In this approach, control of the forgetting factor (FF) is based on the dynamic evaluation of the gradient of MSE. The idea of the current work is to introduce a new type of gradient based FF. The new dynamic FF (DFF) is based on the gradient of inverse correlation matrix. It is show that the results of DFF-RLS are better than GVFF-RLS. Also the proposed approach is compared to forgetting factor based on the gradient of error, fusion of gradient error and inverse correlation matrix and fusion of gradient MSE and inverse correlation matrix and results show superiority of the propose DFF-RLS algorithm.
The RLS algorithm is explained in Section 2. Section 3 is devoted to proposed approach, DFF-RLS. Experimental results are discussed in section 4 and in final section conclusions are presented.

## II. RLS Algorithm

RLS filter is an adaptive, time-update version of the Wiener filter. This algorithm is used for finding the system transfer function, noise cancellation, finding the inverse system function, and prediction. Its purpose is to minimize the weighted sum of squared error, i.e. the error function in the time domain obtained from Eq. (1)

$$\varepsilon_k = \sum_{i=1}^{k} \lambda^{k-i} e_i^2 \tag{1}$$

Where $e_k$ is the error signal, $e_k = d_k - X_k^T W_k$ and $W_k = [W_1,...W_L]^T$ is the weight vector of the RLS filter with input signal $X_k = [x_1,...,x_L]^T$ and $\lambda$ is the forgetting factor $0 \le \lambda \le 1$. The filter weights are obtained from the following equations,

$$\begin{aligned} R_k W_k &= P_k \\ W_k &= R^{-1}{}_k P_k \end{aligned} \tag{2}$$

[1] Hadi Sadoghi Yazdi is assistant professor in Engineering Department of Tarbiat Moallem University of Sabzevar, Iran and Computer Department of Ferdowsi University of Mashhad, Mashhad, Iran; (e-mail: sadoghi@sttu.ac.ir).

Mehri Sadoghi Yazdi is MSc student in Electrical and Computer Engineering Department of Shahid Beheshti University of Tehran (e-mail: me.sadooghi@mail.sbu.ac.ir).

Mohammad Reza Mohammadi is MSc in Department of physical education and sport science of Tarbiat Moallem University of Sabzevar, Sabzevar, Iran.

Where $R_k$ is the input autocorrelation function and $P_k$ is the cross correlation vector between the input signal and the desired signal:

$$R_k = \sum_{i=1}^{k} \lambda^{k-i} X_k X_k^T \tag{3}$$

$$P_k = \sum_{i=1}^{k} \lambda^{k-i} X_k d_k \tag{4}$$

It is apparent that $R_k$ and $P_k$ can be written as,

$$\begin{aligned} R_k &= \lambda R_{k-1} + x_k x_k^T \\ P_k &= \lambda P_{k-1} + x_k d_k \end{aligned} \tag{5}$$

For computing $R_k^{-1}$, equation (1-A) of lemma in appendix A is used. With the correlation matrix $R_k$ assumed to be positive definite and therefore nonsingular, we may apply the matrix inversion lemma (appendix A) to the equation (5).

We first use the following definitions:

$$A = R_k,\ B^{-1} = \lambda R_{k-1},\ C = X_k,\ D = 1$$

And substituting in equation 2-A of appendix-A, we obtain:

$$R_k^{-1} = \lambda^{-1} R_{k-1}^{-1} - \frac{\lambda^{-2} R_{k-1}^{-1} X_k X_k^T R_{k-1}^{-1}}{1 + \lambda^{-1} X_k^T R_{k-1}^{-1} X_k} \tag{6}$$

If we define,

$$\mathrm{K}_k = \frac{\lambda^{-1} R_{k-1}^{-1} X_k}{1 + \lambda^{-1} X_k^T R_{k-1}^{-1} X_k} \tag{7}$$

Substituting Eq. (7) in Eq. (6) and with some simplifications,

$$R_k^{-1} = \lambda^{-1} R_{k-1}^{-1} - \lambda^{-1} \mathrm{K}_k X_k^T R_{k-1}^{-1} \tag{8}$$

And after substitution of Eq. (8) in Eq. (2) and using Eq. (5) and $W_{k-1} = R_{k-1}^{-1} P_{k-1}$ we get:

$$\begin{aligned} W_k &= R^{-1}{}_k P_k \\ &= (\lambda^{-1} R_{k-1}^{-1} - \lambda^{-1} \mathrm{K}_k X_k^T R_{k-1}^{-1})(\lambda P_{k-1} + X_k d_k) \\ &= W_{k-1} + \mathrm{K}_k (d_k - X_k W_{k-1}) \\ &= W_{k-1} + \mathrm{K}_k e_k \end{aligned} \tag{9}$$

Eq. (7) can be written as,

$$\begin{aligned} \mathrm{K}_k &= \lambda^{-1} R_{k-1}^{-1} X_k - \lambda^{-1} \mathrm{K}_k X_k^T R_{k-1}^{-1} X_k \\ &= (\lambda^{-1} R_{k-1}^{-1} - \lambda^{-1} \mathrm{K}_k X_k^T R_{k-1}^{-1}) X_k \\ &= R_k^{-1} X_k \end{aligned} \tag{10}$$

also Eq. (9) can be written as,

$$W_{k+1} = W_k + R_k{}^{-1} X_k e_k \tag{11}$$

## III. Dynamic Forgetting Factor, DFF-RLS

The control mechanism is to adjust FF in order to minimize MSE and increase the convergence speed. In [8], [500] FF is calculated recursively as:

$$\lambda_k = \lambda_{k-1} - \eta \nabla_\lambda (k) \tag{12}$$

Where, $\eta$ is the step size and $\nabla_\lambda (k) = \nabla_\lambda (E\{e^2(k)\})$. We propose DFF-RLS with higher capability compared to method of [8] and [500]. The essence of the DFF-RLS algorithm is to use the decreasing effect of noisy sample and increasing effect of old samples. Practically, it uses the dynamic equation to calculate the MSE rather than using the noisy instantaneous estimate. Using the steepest descent method, the FF can be updated recursively as:

$$\lambda_k = \lambda_{k-1} - \eta \nabla tr(R_k^{-1}) \tag{13}$$

Where, $\lambda_{k-1}$ is forgetting factor at time k-1 and $\eta$ is the step size, also

$$\nabla tr(R_k^{-1}) = tr(R_k^{-1}) - tr(R_{k-1}^{-1}) \tag{14}$$

Where $tr(R_k^{-1})$ is trace of inverse correlation matrix at time k or k$^{th}$ sample.

### *A. The Behavior of DFF-RLS Algorithm*

DFF-RLS behaves as an smoother on input data. The degree of non-stationary affects on the calculation of $\lambda_k$. If the degree of non-stationary of environment noise increases, then $tr(R_k^{-1})$ will decrease, therefore $\nabla tr(R_k^{-1})$ will be negative and $\lambda_k$ is decreased. On the other hand, the effect of noisy samples is decreased in calculation of $\varepsilon_k$ of Eq.(1) and the weight of sample $W_{k+1}$ in Eq.(11) or average of square error slake.

Noisy inputs eventuate increasing at eigenvalue spreads, i.e. reduction of effect of noisy samples lead to low eigenvalue spreads and accordingly, convergence speed increases.

Our experiments show that DFF-RLS have smaller MSE and higher speed than the conventional RLS. This is discussed in the next section.

## IV. Experimental Results

In this section, the convergence property of DFF-RLS is evaluated then tracking of noisy chirp sinusoid is tested. Finally, DFF-RLS algorithm is used in human motion analysis and results are discussed.

### *A. The Convergence Behavior*

Computer simulations are accomplished to verify the convergence time of DFF-RLS relative other approachs. Figure (1) shows a plant identification problem. The plant noise $b_k$ is a stationary white zero mean Gaussian sequence with variance $\sigma_b^2$. A broadband signal $x_k$ is applied to the input of adaptive filter and unknown plant. After adaptation the plant is identified.

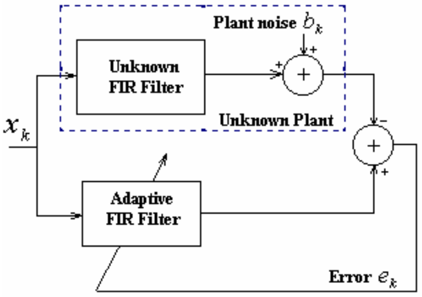


Fig.1. Adaptive plant identification

Figure (2) shows variations of MSE versus iteration number. This figure shows an increase in speed of convergence to optimum weight, i.e. the rate of identification of unknown FIR filter is increased.

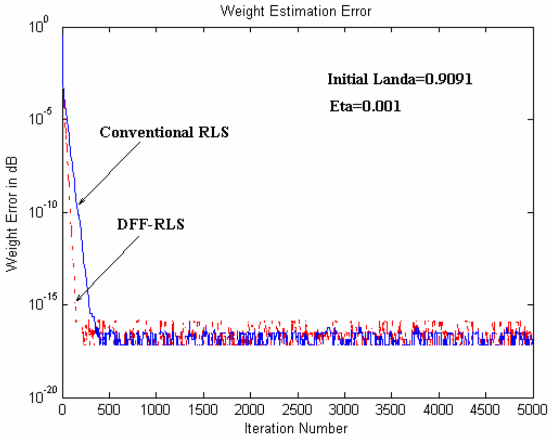


Fig.2. Comparison of convergence speed of DFF-RLS and conventional RLS

In Fig 3, DFF-RLS and GVFF-RLS [8], [500] are compared fir the identification problem. As is seen DFF-RLS have better convergence speed compared to GVFF-RLS algorithm.

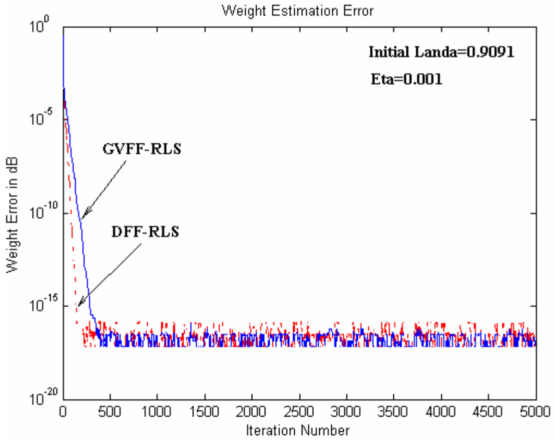


Fig.3. Comparison of convergence speed of DFF-RLS and GVFF-RLS

*B. The Tracking of Noisy Chirp Behavior*

In this experiment, we consider the tracking of a chirped sinusoid. The chirped input signal is given by:

$$S(k)=\sqrt{P_s}e^{j[(2\pi f_c+\psi k/2)k+\varphi]} \tag{15}$$

Where, $\sqrt{P_s}$ denotes the signal amplitude, $f_c$ is the center frequency, $\psi$ is the chirp rate and $\varphi$ is an arbitrary phase shift. The signal $S(k)$ is deterministic but nonstationary because of the chirping. If $S(k)$ is added with noise $n(k)$, then tracking of noisy chirp would be a good benchmark for testing DFF-RLS and GVFF-RLS an other methods. The signal-to-noise ratio (SNR) is denoted as:

$$SNR=10\log\left(\frac{\sqrt{P_s}}{A_{n(k)}}\right) \tag{16}$$

Where, $A_{n(k)}$ is the noise amplitude.

In our experiment, noisy chirp with center frequency 150Hz and 1 KHz sample rate is utilized. The dynamic FF, $\lambda_k$, achieved by Eq.(13) with SNR=16dB is depicted into Fig 4 .

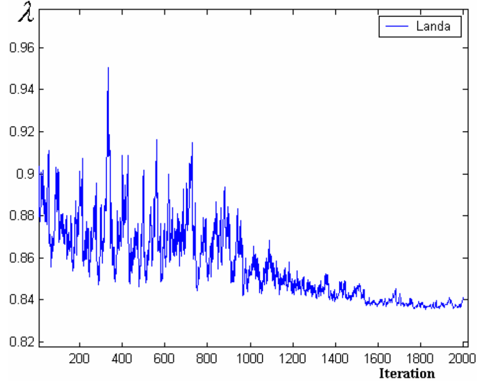


Fig.4. $\lambda_k$ with SNR=16dB by DFF-RLS

MSE variations of DFF-RLS, GVFF-RLS and RLS with $\lambda=0.909$ are illustrated in Fig 5.

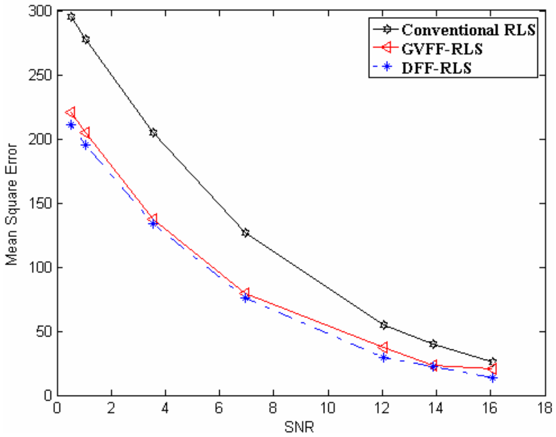


Fig.5. MSE versus SNR smaller than 16dB

The mean of MSE for SNR's smaller than 16dB for DFF-RLS, GVFF-RLS and RLS with $\lambda = 0.909$ and 2001 samples, are 97.09, 103.36 and 146.44 respectively. Also, the mean MSE for SNR's ranging from 16dB to 70dB for DFF-RLS, GVFF-RLS and RLS are 4.34, 5.44 and 6.62 respectively.

The above discussions and results show that DFF-RLS and GVFF-RLS have better operation in low SNR; also DFF-RLS has 6.2dB lower MSE than GVFF-RLS in this range. But in high SNR, three algorithms have approximately tha same results. Howbeit, in each SNR DFF-RLS obtains better performance.

### *C. The Tracking of Body Parts Using DFF-RLS Algorithm*

Human motion analysis from video streams has a wide range of applications such as human-machine interaction, sports, security surveillance, content-based retrieval, etc. In this paper, we focus on biomechanical application of human motion analysis based on memory based tracker. The main contribution of this paper is a novel predictor/smoother for tracking of wearing reflective marker. DFF-RLS helps to smooth trajectories while predicting position for reduction of search area. In this sub-section, firstly we survey previous work on vision-based human motion analysis to explaining the importance of our work and then results of body part tracking are presented.

An intensive survey on the different methodologies for visual analysis of human movement can be found in [15]. In this paper Gavrila groups all approaches into 2D approaches with or without explicit shape models and 3D approaches. Human motion estimation approaches are either based on information about the object shape, namely, model based approaches or methods that do not rely on a priori shape model. 2D approaches without explicit shape models are based on describing human movement in terms of simple low-level 2D features instead of recovering the pose. The second approach, which is a view-based approach, uses explicit shape models to segment, track, and label body parts. 3D approaches attempt to recover the 3D poses over time.

In [16] Moeslund and Granum surveyed various computer vision-based human motion captures. They elaborate various categories of human motion capture, namely, initialization, tracking, pose estimation, and recognition.

In [17] Wang & Baciu proposed an approach for human motion estimation from monocular image sequences. A model based on the notion of relative deformation was presented that introduced a way for anthropomorphic body locomotion analysis including clinical gait analysis and robots motion analysis.

Wang & Baciu grouped motion of parts of a human body. Nine points: head, right shoulder, left and right knee, right elbow, right hip, right wrist, and right ankle were used as feature points. The movement of the three feature points located at shoulder, elbow and wrist represented the movement of the arm. Similarly, another group of points located at hip, knee and ankle represent the movement of the leg. But for practical tracking of all feature points, they grouped markers and in each snap shot there exits only one group.

This approach helped them to avoid occlusion of markers, and false attribution. Simple background and glare of markers and darkness of dress were simplifying assumptions in their work. But in normal conditions one can not easily collect these assumptions. We try the same markers for tracking in practical conditions but track all markers simultaneously.

As Wang, Hu and Tan have surveyed recent developments in human motion analysis in [18], prediction algorithms are from conventional method as Kalman filter, optical flow and types of least square which we reviewed them in introduction. In [19] an optical flow process allows the prediction of the position of the model from its position at the previous time. Marzani in [19] paper tried tracking the leg of a human body.

In [20], Lu used first-order Kalman filter for prediction of drowning incidents in swimming pools. In this paper, the vision component used a first-order Kalman filter to track each detected swimmer over frames. During the tracking, the position of the swimmer in every subsequent frame can first be predicted by the filter so that a local search window can be specified for swimmer segmentation. Furthermore, a best-fit ellipse is obtained to approximate the size of each tracked swimmer. Also Kalman filter was used by Kim in [23] in pose estimation of head, hand and feet independently. Also, Roy in [22] used a statistical model for reducing noise in gate database generation.

The Kalman filter is an optimum filter which is model based and minimizes the variance of the estimator error. In practice, the motion model of objects inside the scene does not exist and it is possible to be suggested either fixed or variable. The need of the Kalman filter to a model is one of problems when using this filter [21].

### *D. The Survey of Tracking in Human Motion Analysis*

The simulation and analysis of human motion is a popular issue in simulation area. It is highly valuable in sports motion training and analysis. A method based on computer vision technologies is presented to achieve the structure of motion in human. It is very helpful for training athletes. Of course structures observed in humans are being progressively applied to the theoretical approaches developed in many applications like robotics. From these observations, they are able to estimate the forces and moments at each joint [24]-[26].

The captured structure from motion highly depends on noise reduction algorithm in tracking system. In this paper, we provide a suitable tracking algorithm and apply it for analyzing human motion.

The tracking algorithm has become a significant tool in computer vision. A variety of methods there exist for tracking objects in outdoor scenes. Many types of unsupervised object tracking methods are presented which popular methods include feature based, region based and model based tracking algorithms [5], [27]-[30].

### *E. The proposed algorithm in marker tracking*

Marker Tracking is one of the biomechanical motion analysis methods. In this method, markers are installed on specific points of a human body. Many problems there exist in tracking markers as such as,

a) Occlusion conditions can cause loss of markers in some durations of time. We used the DFF-RLS algorithm which predicts marker position in a few frames.
b) Reflection of light from peripheral equipments and non-uniformity of gray valued markers. We encountered to this problem after data collecting. In our experiments, 18 markers are used which some of markers have different light. Of course, light of shirt of samples were another problem.

The above problems are different types of noises which are observed in marker tracking which are incentives for using adaptive filter for noise reduction. The tracking algorithm has following steps,

a) Capturing markers positions in the first frame using suitable GUI (Graphic User Interface).
b) Matching 7*7 buffered sub-image in consecutive frames using correlation method. Best search area is 7*7 windows.
c) After convergence of the DFF-RLS for each marker, the predictor/smoother trajectory is used for prediction of the position and smoothing of the trajectory for finding correct velocity and acceleration information's. The DFF-RLS algorithm helps for prediction of loosed markers in 5-20 frames depend on the loosed position relative to turning point.
d) The GUI software provides for user which guides tracking or it provides HCI (Human Computer Interaction) for user in the markers tracking.

As it is seen eighteen markers are shown in Fig 6. We have used negative image for suitable showing of markers positions. Tracking results for marker no.13 is shown in Fig 7, and Fig 8 shows 71 tracked frames. Effect of smoothing is seen in Fig 9. Smoothed trajectory is important for finding position without shaking and correct velocity and higher order of derivation of position. For finding the performance of the smoother, we collected a data set of manual main positions and mean square error is obtained for calculation of smoother effect. The performance is obtained from (17),

$$P = \frac{\left|\xi_{IS} - \xi_{InS}\right|}{\xi_{IS}} \tag{17}$$

Where,

$$\xi_{IS} = \sum_{i=1}^{N}\left(\left\|\begin{pmatrix} x_i \\ y_i \end{pmatrix}_I - \begin{pmatrix} x_i \\ y_i \end{pmatrix}_S\right\|\right), \tag{18}$$

$$\xi_{InS} = \sum_{i=1}^{N}\left(\left\|\begin{pmatrix} x_i \\ y_i \end{pmatrix}_I - \begin{pmatrix} x_i \\ y_i \end{pmatrix}_{nS}\right\|\right) \tag{19}$$

Where, index I stands for ideal, index S is smooth version and index nS is no smoothed trajectories. N is number of captured samples. The performance is obtained over collected data set. The specifications of data set are,

- 12 sport men moved over captured tool
- The camera with 350 FPS (Frame per second) with gray images
- Each man moved in 6 different velocities

For measuring robustness of the proposed tracker we added different level uniform noise to x and y positions. As shown in Fig 10, the proposed DFF-RLS algorithm obtains better result for low SNR relative to GVFF-RLS algorithm.

One of result of designed software is film of motion skeleton and combination of negative film and skeleton which depicted in Fig 11, 12 and 13 numbers of constructed images have been shown.

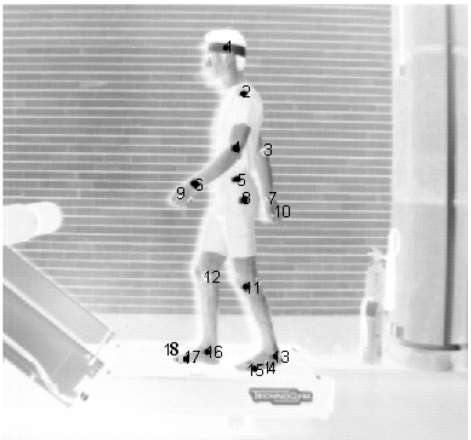
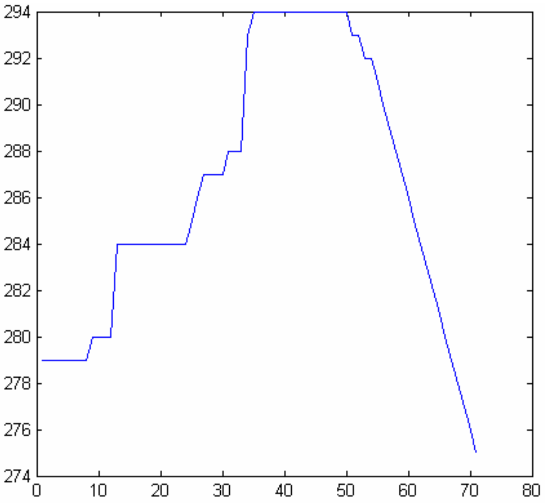
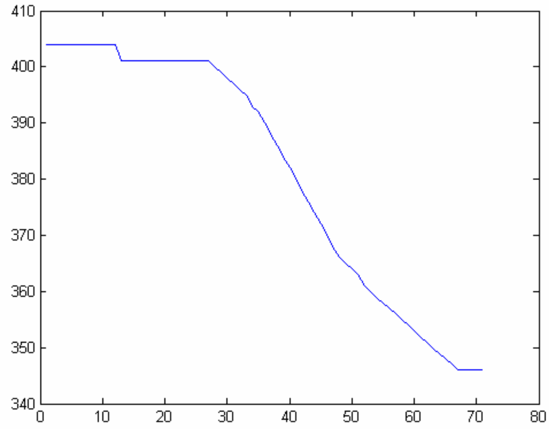


Fig.6: Eighteen markers in negative image



(a)



(b)

Fig.7: The tracked position of 14th marker, a) X, b)Y positions

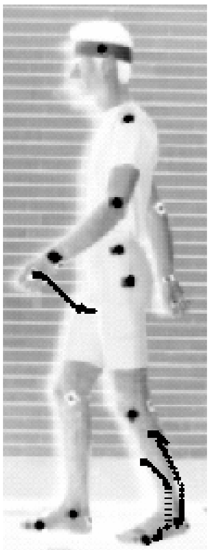
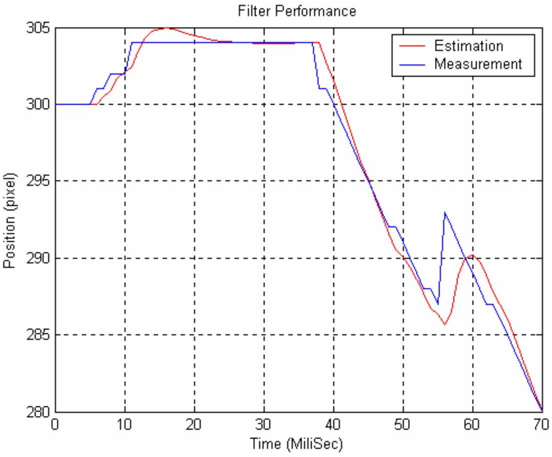
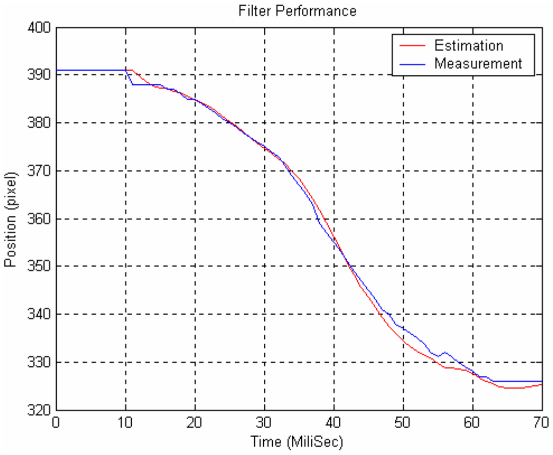


Fig.8: Trajectory of 9, 13,14th marker in 71 frames



(a)



(b)

Fig.9: Smoothing of 13th marker a) X position using the DFF-RLS algorithm, b) Y position

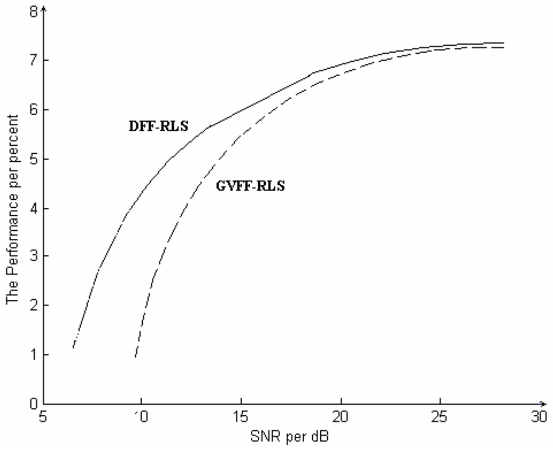


Fig.10: The performance per SNR

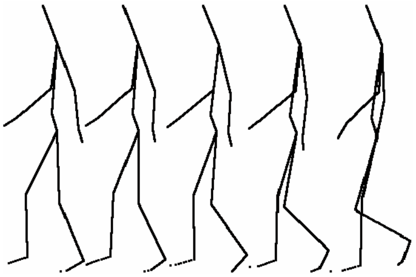


Fig.11: Motion skeleton in 5 frames from 50 tracked frames.

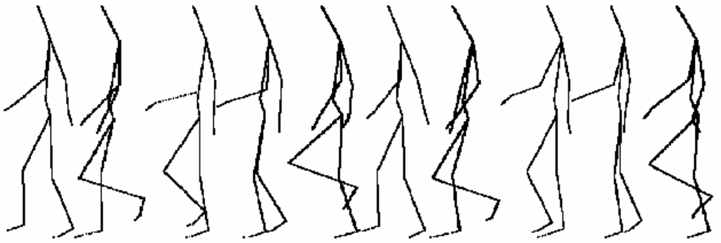


Fig.12: Motion skeleton in 10 frames from 500 tracked frames

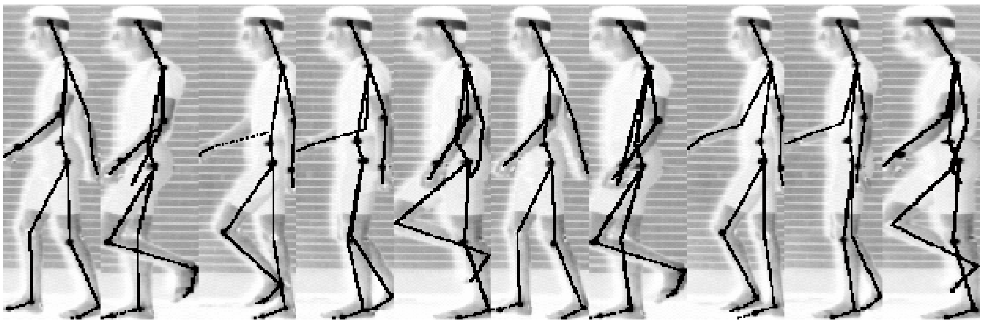


Fig.13: Motion combined skeleton and negative frames in 10 frames from 500 tracked frames

## V.CONCLUSION

This paper is concerned with studying the forgetting factor of the RLS. A new dynamic forgetting factor, DFF, for RLS algorithm was presented. The DFF-RLS compared to GVFF-RLS and conventional RLS of convergence speed and tracking of noisy chirp. Experimental results show DFF-RLS has higher speed convergence relative GVFF-RLS and conventional RLS. Also, results show DFF-RLS and GVFF-RLS have better operation in low SNR; also DFF-RLS reduced MSE 6.2dB relative GVFF-RLS in this duration. But in high SNR, operation of three algorithms is equal approximately. Howbeit, in each SNR DFF-RLS obtain better performance.

## REFERENCES

[1] S.Haykin, Adaptive Filter Theory, 3rd-ed, Printice Hall,1996.

[2] S.Haykin, A.H.Sayed, J.Zeidler, P.Yee, P.Wei, “Tracking of linear Time-Variant Systems,” Proc. MILCOM, pp.602-606, San Diego, Nov. 1995.

[3] H.Sadoghi Yazdi, M.Lotfizad, E.Kabir , M.Fathi "Application of trajectory learning in tracking vehicles in the traffic scene" 9th Iranian computer conference vol.1, pp.180-187, Feb 2004. (In Persian)

[4] H.Sadoghi Yazdi, M.Lotfizad "A new approach for tracking objects using combination of RLS and MAP algorithms, 11th,Iranian Electronic Eng. Conference vol.3, pp.258-266, 2003. (In Persian)

[5] J. Jiang, R. Cook, "Fast Parameter Tracking RLS Algorithm with High Noise Immunity," Electronic Letters 28, pp. 2043–2045, October 1992.

[6] D.J. Park, B.E. Jun, "Self-Perturbing RLS Algorithm with Fast Tracking Capability," Electronics Letters 28, pp. 558–559, March 1992.

[7] J.M. Cio, T. Kailath, "Fast Fixed-Order, Least Squares Algorithms for Adaptive Filtering," ICASSP 83, Boston, 1983.

[8] C.F. So, S.C. Ng, S.H. Leung, "Gradient Based Variable Forgetting Factor RLS Algorithm," Signal Processing 83, pp. 1163 – 1175, 2003.

[9] [9] T.R. Fortescue, L.S. Kershenbaum, B.E. Ydstie, Implementation of Self-Tuning Regulators with Variable Forgetting Factors," Automatica 17, pp. 831–835, 1981.

[10] [10] D.J. Park, et al., "Fast Tracking RLS Algorithm Using Novel Variable Forgetting Factor with Unity Zone, Electron. Letters 27, pp. 2150–2151, November 1991.

[11] [11] S. Song, et al., "Gauss Newton Variable Forgetting Factor Recursive Least Squares for Time Varying Parameter Tracking," Electron. Letters 36, pp. 988–990, May 2000.

[12] [12] D. T. M. Slock and T. Kailath, “Fast transversal filters with data sequence weighting,” IEEE Trans. Acoust., Speech, Signal Process., vol. 33, no. 3, pp. 346–359, Mar. 1989.

[13] [13] B. Toplis and S. Pasupathy, “Tracking improvements in fast RLS algorithms using a variable forgetting factor,” IEEE Trans. Acoust., Speech, Signal Process., vol. 36, no. 2, pp. 206–227, Feb. 1988.

[14] [14] Shu-Hung Leung, C. F. So, "Gradient-Based Variable Forgetting Factor RLS Algorithm in Time-Varying Environments," IEEE Trans. On Signal Processing, vol. 53, no. 8, pp.3141-3150, Aug. 2005.

[15] [15] D.M. Gavrila, “The Visual Analysis of Human Movement: A Survey,” Computer Vision and Image Understanding, vol. 73, no. 1, pp. 82-98, 1999.

[16] [16] T.B. Moeslund and E. Granum, “A Survey of Computer Vision-Based Human Motion Capture,” Computer Vision and Image Understanding, vol. 81, no. 3, pp. 231-268, Mar. 2001.

[17] [17] Y. Wang, G. Baciu, "Human Motion Estimation from Monocular Image Sequence Based on Cross-Entropy Regularization," Pattern Recognition Letters, vol.24, pp.315–325, 2003.

[18] [18] L. Wang, W. Hu, T. Tan, "Recent Developments in Human Motion Analysis," Pattern Recognition, vol. 36, pp. 585-601, 2003.N. Kawasaki, “Parametric study of thermal and chemical nonequilibrium nozzle flow,” M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[19] [19] F. Marzani, E. Calais, L. Legrand, "A 3-D Marker-Free System for the analysis of Movement Disabilities-An Application to the Legs," IEEE Trans. On Information Technology in Biomedicine, Vol. 5, No.1, March 2001.

[20] [20] W. Lu, Y-P. Tan, "A Vision-Based Approach to Early Detection of Drowning Incidents in Swimming Pools," IEEE Trans. On Circuits and Systems for Video Technology, Vol. 14, No.2, pp. 159-178, Feb. 2004.

[21] [21] L.Zhao, C.Thorpe, “ Qualitative and Quantitative Car Tracking from a Range Image Sequence,” Proc. CVPR, Santa Barbara, CA, June 23-25, pp. 496-501,1998.

[22] [22] A. K. Roy-Chowdhury, "A measure of deformability of shapes, with applications to human motion analysis," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, Vol.1, pp.398-404, June 2005.

[23] [23] S-E. Kim, C-J. Park, I-H. Lee, "A Tracking Method of End-Effectors in a Vision-Based Marker-Free Motion Capture System," Proceedings of the 2004 IEEE conf. on Cybernetics and Intelligent Systems, pp. 129-134, Dec. 2004.

[24] [24] Q. Xian-Jie, W. Zhao-Qi, X. Shi-Hong, W. Yong-Dong, "A Virtual-Real Comparison Technique Used on Sport Simulation and Analysis," IEEE

[25] [25] L. Molina-Tanco, J. P. Bandera, R. Marfil, F. Sandoval, "Real-time Human Motion Analysis for Human-Robot Interaction," IEEE/RSJ International Conf. on Intelligent Robots and Systems, IROS 2005, pp. 1808-1813, Aug. 2005.

[26] [26] B. Dariush, H. Hemami, M. Parnianpour, "Analysis and Synthesis of Human Motion From External Measurements," IEEE Int. Conf. on Robotics & Automation, pp. 4015-4020, April 2000.