and

$$H(n) = [h_1(0;n), h_1(1;n), ..., h_1(N-1;n), h_{20}(0;n), h_{20}(1;n), ..., h_{2,N-1}(0;n)]^T \quad (6)$$

By modifying the filtered-X LMS algorithm [4], we obtain the filtered-X second-order Volterra LMS algorithm with the coefficient adaptation as follows:

$$H(n+1) = H(n) + \mu e(n) U(n) \quad (7)$$

where μ is a diagonal matrix with μ_L for the first N diagonal entries and μ_Q for the rest of the diagonal entries, and $U(n)$ is the filtered signal vector shown below

$$U(n) = [u(n), u(n-1), ..., u(n-N+1), u_{20}(n), u_{20}(n-1), ..., u_{2,N-1}(n)]^T \quad (8)$$

Note that the dimension of $X(n)$, $H(n)$ and $U(n)$ is $(N(N+3)/2) \times 1$. Based on the multichannel structure, the $N+1$ independent elements of $U(n)$ can be obtained from $\tilde{U}(n) = \hat{s}(n)^* \tilde{X}(n)$, where $\tilde{X}(n) = [x(n), x_{20}(n), x_{21}(n), ..., x_{2,N-1}(n)]^T_{(N+1)\times 1}$, $\tilde{U}(n) = [u(n), u_{20}(n), u_{21}(n), ..., u_{2,N-1}(n)]^T_{(N+1)\times 1}$ and $\hat{s}(n)$ is the impulse response of the secondary path estimate $\hat{S}(z)$. Note that $u(n) = \hat{s}(n)^* x(n)$ and $u_{2j}(n) = \hat{s}(n)^* \{x(n)x(n-j)\}$, $j = 0,1, ..., N-1$. The implementation is also depicted in Fig. 1.

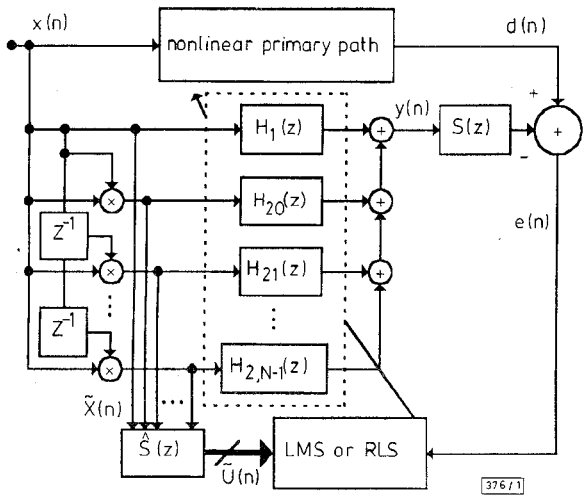

Fig. 1 *Active noise control system using filtered-X second-order Volterra adaptive algorithm*

Similar to the derivation in [4], we propose the filtered-X second-order Volterra RLS algorithm as following

$$e(n) = d(n) - s(n) * y(n) = d(n) - s(n) * \{H^T(n) X(n)\} \quad (9)$$

$$k(n) = \frac{\lambda^{-1} P(n-1) U(n)}{1 + \lambda^{-1} U^T(n) P(n-1) U(n)} \quad (10)$$

$$H(n+1) = H(n) + k(n) e(n) \quad (11)$$

$$P(n) = \lambda^{-1} P(n-1) - \lambda^{-1} k(n) U^T(n) P(n-1) \quad (12)$$

where λ ($0 \ll \lambda \leq 1$) is the forgetting factor, $k(n)$ the gain vector and $U(n)$ defined in eqn. 8. Note that $e(n)$ is the residual noise measured by the error sensor; and the coefficient vector in eqn. 11 is shifted by one sample as compared with the one in the standard RLS since the coefficient vector is required before the next sample arrives in order to generate the adaptive filter output $y(n)$ in eqn. 9.

Simulation: In the following simulation, we use the zero-mean Gaussian noise with a variance of 1.0 as the input signal and obtain the primary disturbance $d(n)$ with a quadratic nonlinearity by the following procedure. We first filter the input $x(n)$ using a delayed bandpass filter given by $C_1(z) = z^{-2}/(1-0.2z^{-2})$ to generate $p(n)$, where $p(n) = c_1(n)^* x(n)$; we then obtain $q(n)$ using $p(n)$ and a quadratic function in the following expression $q(n) = p(n)+0.5p^2(n)$; we finally filter $q(n)$ via another delayed bandpass filter represented by $C_2(z) = z^{-2}/(1-0.25z^{-2})$ to give the primary disturbance $d(n) = c_2(n)^* q(n)$. Notice that $c_1(n)$ and $c_2(n)$ are the impulse responses of $C_1(z)$ and $C_2(z)$, respectively. The secondary-path transfer function is assumed to be $S(z) = z^{-2}/(1-0.3z^{-2})$ (a delayed bandpass) while the secondary path estimate is $\hat{S}(z) = z^{-2}/(1-0.35z^{-2})$. We choose the memory span of the adaptive second-order Volterra filter as $N = 9$ so that the number of filter coefficients is $54(N(N+3)/2)$. We set the step sizes as $\mu_L = 0.01$ and $\mu_Q = 0.001$ for the filtered-X second-order Volterra LMS, and the forgetting factor as $\lambda = 0.995$ for the filtered-X second-order Volterra RLS. As a comparison, we also use the standard filtered-X LMS algorithm with a filter order of 54 for the adaptive FIR filter, and set the step size as 0.0002. The squared residue errors are plotted in Fig. 2 for 3500 iterations. As illustrated in Fig. 2, the standard filtered-X LMS performs poorly, while both developed algorithms can perform well to reduce the primary disturbance. Specially, the filtered-X second-order Volterra RLS algorithm achieves the significantly improved performance in terms of convergence speed.

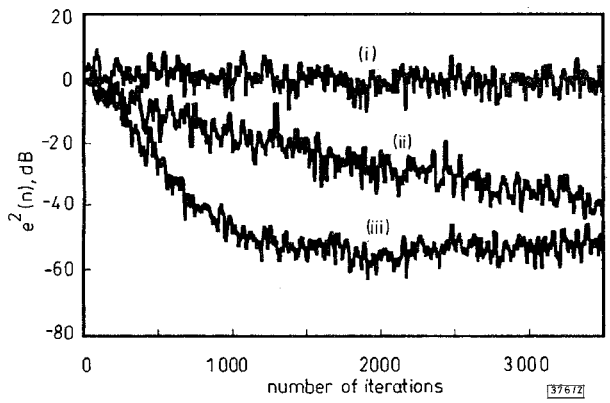

Fig. 2 *Performance comparison for filtered-X adaptive algorithms*

(i) standard filtered-X LMS
(ii) filtered-X second-order Volterra LMS
(iii) filtered-X second-order Volterra RLS

Conclusion: We have developed the filtered-X second-order Volterra adaptive algorithms based on a multichannel structure for active noise control. The simulation results demonstrate that the developed algorithms outperform the standard filtered-X LMS algorithm when the primary path has a quadratic nonlinearity. The filtered-X second-order Volterra RLS algorithm converges very fast at the cost of an increase in the computational complexity.

24 February 1997
Electronics Letters Online No: 19970477

Li-Zhe Tan and Jean Jiang (*Interactive College of Technology, 5303 New Peachtree Road, Atlanta, Georgia 30341, USA*)

References

1 ELLIOTT, S.J., and NELSON, P.A.: 'Active noise control', *IEEE Sig. Process. Mag.*, 1993, pp. 12–35
2 MATHEWS, V.J.: 'Adaptive polynomial filters', *IEEE Sig. Process. Mag.*, 1991, pp. 10–26
3 TAN, L.Z., and JIANG, J.: 'Adaptive second-order Volterra delay filter', *Electron. Lett.*, 1996, **32**, (9), pp. 807–809
4 KUO, S.M., and MORGAN, D.R.: Active noise control systems-algorithms and DSP implementations' (John Wiley and Sons, New York, 1996)

Modified Moose estimator for tracking highly manoeuvring targets

Chi-Min Liu and Kuo-Guan Wu

Indexing terms: Target tracking, Kalman filters

The Moose's adaptive state estimator has proved successful in tracking a manoeuvring submarine [1]. However, its application to track highly manoeuvring aircraft will encounter problems in complexity and tracking accuracy [2, 3]. The authors present a new algorithm that extends the Moose estimator to solve its problems. The new algorithm is verified through Monte-Carlo simulation and shows a better performance than the original Moose algorithm.

Introduction: The major aspects of designing an algorithm for tracking manoeuvring targets includes the modelling of the input exercised by the pilot that causes manoeuvres, and the combination of the manoeuvre model with the Kalman filter to form the tracking algorithm. In the Moose's adaptive state estimator [1], the possible manoeuvre inputs are discretised into a finite set of values and the transition among those discrete values is modelled as a semi-Markov process. The Moose estimator consists of a Baysian method for manoeuvre estimation and a single Kalman filter for state estimation. The Baysian method estimates a target's manoeuvre value by a weighted combination of those discrete values; the single Kalman filter estimates a target's state by taking the manoeuvre estimate as the true manoeuvre value in computing the state prediction. When the Moose's algorithm is used to track highly manoeuvring targets, the following difficulties will arise [2, 3]. First, its complexity increases with the large set of discrete values for covering the wide range of manoeuvres. Secondly, the measurement noise affects the evaluation of the weighting probabilities of Baysian manoeuvre estimation, and hence degrades the accuracy of manoeuvre and state estimation results.

To solve these difficulties, we modify the Moose's algorithm with the following extensions. First, the manoeuvre model is extended to consider the second-order statistic of the manoeuvre variable. Secondly, the measurement noise is smoothed by a scheme that explores the correlation between successive manoeuvre values. Thirdly, the manoeuvre estimation error is compensated in the Kalman filter to improve its convergence behaviour. Simulation results demonstrate that the resultant new algorithm can track highly manoeuvring targets with a better performance than the original Moose algorithm.

Algorithm derivation: The dynamic and measurement of a manoeuvring target can be modelled by the following equations:

$$x(k) = \Phi \cdot x(k-1) + B \cdot U(k-1) + v(k-1) \quad (1)$$

$$z(k) = H \cdot x(k) + w(k) \quad (2)$$

where $x(k)$ is the state vector and $z(k)$ is the measurement vector. $U(k-1)$ denotes the manoeuvre input vector. $v(k)$ is a zero-mean white Gaussian process with covariance matrix Q; $w(k)$ denotes the measurement noise, assumed to be a zero-mean white Gaussian process with a non-zero covariance R. Considering eqns. 1 and 2, if $U(k-1)$ is known, the Kalman filter can be used to track the target. In practice, however, $U(k-1)$ is unknown. In this Letter, we consider $U(k-1)$ as an unknown variable and model its probability density function as a mixture of multiple Gaussian density functions as

$$f(U(k-1)|Z^k) = \sum_{i=1}^{N} w_i(k-1) \cdot N(U(k-1); U_i, C_i) \quad (3)$$

where Z^k denotes the cumulative measurement sequence $z(1)$, ..., $z(k)$; $N(U(k-1); U_i, C_i)$ is a Gaussian density function of mean U_i and covariance C_i; $w_i(k-1)$ denotes the mixture weight. The mean and covariance of $U(k-1)$ can be obtained from those of the mixture components by the following two equations:

$$\hat{U}(k-1) = \sum_{i=1}^{N} w_i(k-1) \cdot U_i \quad (4)$$

$$C(k-1) = \sum_{i=1}^{N} w_i(k-1) \cdot [C_i + (\hat{U}(k-1) - U_i) \cdot (\hat{U}(k-1) - U_i)^t] \quad (5)$$

We take the mean value in eqn. 4 as the estimate of $U(k-1)$ and use the covariance in eqn. 5 to evaluate the estimation error of eqn. 4.

The expressions for evaluating $w_i(k-1)$ are summarised as follows. Denote the event that $U(k-1)$ belongs to the ith mixture component by $U_i(k-1)$ and assume that the transition among the mixture components satisfies the first-order Markov property, then according to the Bayes rule, $w_i(k-1)$ can be evaluated by

$$w_i(k-1) = \Pr(U_i(k-1)|Z^k) = \frac{\Pr(U_i(k-1)|Z^{k-1}) \Pr(z(k)|U_i(k-1), Z^{k-1})}{\Pr(z(k)|Z^{k-1})} \quad (6)$$

where $\Pr(U_i(k-1)|Z^{k-1})$ is recursively computed by

$$\Pr(U_i(k-1)|Z^{k-1}) = \sum_{j=1}^{N} p_{ji} \cdot \Pr(U_j(k-2)|Z^{k-1}) \quad (7)$$

and $\Pr(z(k)|U_i(k-1), Z^{k-1})$ is evaluated by the Gaussian function

$$\Pr(z(k)|U_i(k-1), Z^{k-1}) = N(z(k); H(\Phi \cdot x(k-1|k-1) + B \cdot U_i), S_i(k)) \quad (8)$$

$$S_i(k) = H \cdot \Phi \cdot P(k-1|k-1) \cdot \Phi^t \cdot H^t + H \cdot Q \cdot H^t + R + H \cdot B \cdot C_i \cdot B^t \cdot H^t \quad (9)$$

The p_{ji} in eqn. 7, which equals $\Pr(U_i(k-1)|U_j(k-2))$, denotes the Markov transition probability; the $P(k-1|k-1)$ in eqn. 9 denotes the covariance of the state estimate at the $(k-1)$th sampling instant.

To reduce the effect of the measurement noise on the accuracy of eqn. 8, the following scheme will be used to smooth the noisy measurement $z(k)$ before computing eqn. 8. The preprocessing scheme consists of the state update equations of Kalman filtering, with the approximation of $U(k-1)$ by the manoeuvre estimate $\hat{U}(k-2)$ of the previous sampling instant, and with the reduction of the Kalman gain matrix by a scaling matrix $\alpha(k)$. As a result, $S_i(k)$ is modified to be

$$S_i'(k) = H \cdot \alpha(k) \cdot K(k) \cdot S(k) \cdot K^t(k) \cdot \alpha^t(k) \cdot H^t + H \cdot B \cdot C_i \cdot B^t \cdot H^t \quad (10)$$

$\alpha(k)$ is designed to reduce $S_i'(k)$ to be of the same order as its final term. In this way, it will become possible to discriminate the correctness of each $U_i(k-1)$ from the likelihood function, and thus to obtain accurate weighting probability $w_i(k-1)$ and accurate manoeuvre estimate $\hat{U}(k-1)$.

The manoeuvre estimation results $\hat{U}(k-1)$ and $C(k-1)$ are incorporated into the prediction equations of Kalman filtering by

$$\hat{x}(k|k-1) = \Phi \cdot \hat{x}(k-1|k-1) + B \cdot \hat{U}(k-1) \quad (11)$$

$$P(k|k-1) = \Phi \cdot P(k-1|k-1) \cdot \Phi^t + Q + B \cdot C(k-1) \cdot B^t \quad (12)$$

Then, a target's current state estimate $\hat{x}(k|k)$ can be obtained by performing the update equations of standard Kalman filtering.

Fig. 1 *Accelerations of target in x direction*

Results and discussion: To compare the performance of the proposed algorithm with the original Moose estimator, a 2D target scenario was generated according to eqns. 1 and 2 with Φ, B, H being

$$\Phi = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

T was set as 1 s. $v(k)$ was set as a zero process and $w(k)$ was set to have the covariance matrix

$$\begin{bmatrix} 100m^2 & 0 \\ 0 & 100m^2 \end{bmatrix}$$

The target scenario was designed to contain five different manoeuvring periods, with acceleration values ranging from –15 to 15g. The x-acceleration process of the target is shown in Fig. 1. A 50-run Monte Carlo simulation was conducted, with the x- and y-movements being tracked independently.

The parameters of the proposed algorithm are summarised as follows. The probability density function of the manoeuvre variable was designed to consist of 18 Gaussian functions. The means of the first 17 functions were selected as: –160, –140, –120, –100, –80, –60, –40, –20, 0, 20, 40, 60, 80, 100, 120, 140, 160m/s^2; the variances were all selected as 50 (m/s^2)^2. The 18th Gaussian function was designed to have mean 0 m/s^2 and variance 1 (m/s^2)^2. The Markov transition probability was set as $p_{ii} = 0.8$, and $p_{ij} = 0.2/17$ $\forall\, i \neq j$. As to the implementation of the Moose algorithm, 17 discrete values: –160, –140, –120, –100, –80, –60, –40, –20, 0, 20, 40, 60, 80, 100, 120, 140, 160m/s^2 were selected as the mean acceleration values. The Markov transition probability was designed as p_{ii} = 0.96, and $p_{ij} = 0.0025$ $\forall i \neq j$.

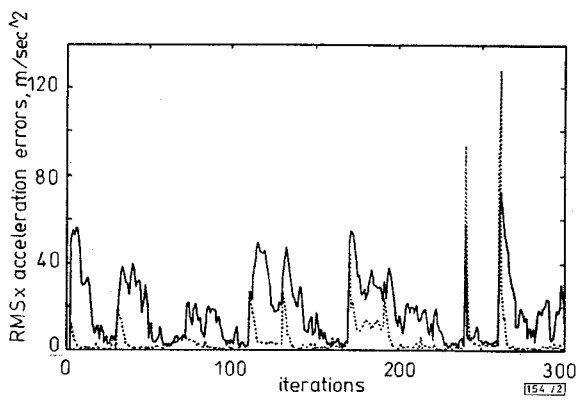

Fig. 2 *RMS acceleration estimation errors of both original Moose algorithm and proposed algorithm in x direction*

——— Moose algorithm
– – – – proposed algorithm

Fig. 2 shows the acceleration estimation errors of both algorithms. From Fig. 2 we can find that the acceleration errors of the proposed algorithm converge faster to smaller values than the Moose algorithm during both the non-manoeuvring and the manoeuvring periods. The magnitudes of the transient errors of the proposed algorithm at the beginnings of manoeuvres depend on the acceleration values. Those transient errors can be attributed to the measurement preprocessing scheme, which increases the effect of the previous acceleration estimate and decreases the effect of the current measurement on the current acceleration estimation. However, during the constant-acceleration periods, the transient errors can converge fast, to be as large as those during the non-manoeuvring periods.

3 February 1997

Electronics Letters Online No: 19970478

Chi-Min Liu and Kuo-Guan Wu (*Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsinchu, Taiwan, Republic of China*)

References

1 GHOLSON, N.H., and MOOSE, R.L.: 'Maneuvering target tracking using adaptive state estimation', *IEEE Trans.*, 1977, **AES-13**, pp. 310–317

2 LIM, C.H., and LEE, H.S.: 'Modified moose estimator with plant-filter mismatch compensation', *Electron. Lett.*, 1993, **29**, pp. 65–67

3 BOGLER, P.L.: 'Tracking a manoeuvring target using input estimation', *IEEE Trans*, 1987, **AES-23**, pp. 298–310

Selective SC-filters with low passive sensitivity

M.D. Lutovac, D. Novaković and I. Markoski

Indexing terms: Elliptic filters, Switched capacitor filters

A standard mode of operation of the universal switch-capacitor (SC) filter is modified so that very selective elliptic-function filters can be realised with a reduced number of external resistors and very low passive sensitivity.

Introduction: This Letter presents a modification to one mode of operation of the cascade connection of second-order SC-filters with reduced passive sensitivity. A selective SC-filter with a very high Q-factor (Q > 20) requires tiny external resistor tolerances. In Table 1 the sensitivities of the best active RC filters [1], mode-3a SC-filters [2] and modified SC-filter, called mode-1d, (in [2] there are modes 1a, 1b and 1c) are summarised. When the Q-factors are very high (e.g. Q = 20), the passive sensitivity of an SC-filter in mode-3a is half of the sensitivity of active RC filters. The passive sensitivity of an SC-filter in mode-1d is several times lower than the sensitivity of mode-3a (10 times in Table 1). The low sensitivity is due to the zero passive pole-frequency sensitivity, which means that the pole-frequency depends only on the clock-frequency.

For attenuation poles very close to the passband edge frequency, the passive sensitivity of RC and SC-filters mode-3a could be very high. For SC-filters in mode-1d, the passive sensitivity is always < 0.5.

Table 1: Approximate value of maximal passive sensitivities

		$\lvert S_R^x \rvert$			max ($\lvert S_R^{\lvert H \rvert} \rvert$)		
x	max ($\lvert S_x^{\lvert H \rvert} \rvert$)	RC	SC-3a	SC-1d	RC	SC-3a	SC-1d
Q	1	Q	1	1	Q	1	1
ω_0	Q	1/2	1/2	0	$Q/2$	$Q/2$	0
ω_n	$1/(\omega_n^2-1)$	1/2	1/2	$(\omega_n^2-1)/2\omega_n^2$	$1/2(\omega_n^2-1)$	$1/2(\omega_n^2-1)$	1/2
for $Q=20$, $\omega_n=1.078$ rad/s		max($\Sigma\lvert S_R^{\lvert H \rvert} \rvert$)			33.1	14.1	1.5

$Q \gg 1$, passband edge frequency ω_p = 1, notch frequency $\omega_n > 1$, pole-frequency $\omega_0 > 1$, R: resistor, H: transfer function, RC: second order active RC filter, SC: switched-capacitor filter in mode 3a or 1d

Filter transfer function: Nevertheless, despite the superiority of mode-1 over other modes and active RC filters, mode-1 is used more theoretically, rather than in practical realisations of selective filters. The main reason is that, usually, a different clock frequency must be implemented for two or more biquads.

To use a single clock-frequency, the pole-frequency of all biquads must be identical. Although the Butterworth filter has all poles on a circle, the order is usually extremely high for very selective applications. Recently proposed elliptic-function filters [3] have all poles on a circle and very selective specifications could be fulfilled with the order slightly higher than the theoretically minimal order. The transfer function H_n of the even-order filter is

$$H_n(s) = \prod_{i=1}^{n/2} H_i(s) = \prod_{i=1}^{n/2} \frac{s^2 + \omega_i^2}{s^2 + \frac{\omega_0}{Q_i} s + \omega_0^2} \qquad (1)$$

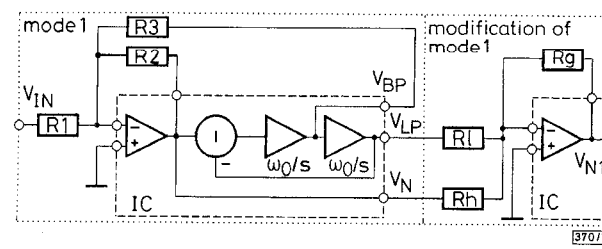

Fig. 1 *Mode-1 and mode-1d of second-order SC-filter*

Mode 1 of SC universal filters: In Fig. 1, a second-order SC filter in mode-1, and its modification is shown. The filter consists of an integrated circuit (IC) and four resistors R_2, R_3, R_h and R_l. The