Kalman filtering & colored noises: the (autoregressive) moving-average case

Matthieu Geist*
*Supélec, IMS Research Group
Metz (France)
matthieu.geist@supelec.fr

Olivier Pietquin*†
†UMI 2958 (GeorgiaTech-CNRS)
Metz (France)
olivier.pietquin@supelec.fr

***Abstract*—The Kalman filter is a well-known and efficient recursive algorithm that estimates the state of a dynamic system from a series of indirect and noisy observations of this state. Its applications range from signal processing to machine learning, through speech processing or computer vision. The underlying model usually assumes white noises. Extensions to colored autoregressive (AR) noise model are classical. However, perhaps because of a lack of applications, moving-average (MA) or autoregressive moving-average (ARMA) noises seem not to have been considered before. Motivated by an application in reinforcement learning, the contribution of this paper is a generic approach to take MA and ARMA noises into account in the Kalman filtering paradigm.**

***Index Terms*—Kalman filtering, colored noises, reinforcement learning**

I. Introduction

The Kalman filter [1] is an efficient recursive algorithm that estimates the state of a dynamic system from a series of indirect and noisy measurements. The corresponding framework, to be briefly presented in section II, assumes white noises. This hypothesis is mandatory to obtain Kalman equations. However, taking a colored noise into account can be of interest, for example for speech processing [2]. Yet, the only type of noise to be considered in the literature is autoregressive (AR), and a classical approach is to extend the process equation [3], as reminded in section III. Motivated by an application in reinforcement learning [4], to be sketched in section VI, this paper introduces a unified and generic approach to take moving-average (MA) and autoregressive moving-average (ARMA) noises into account as well (sections IV and V respectively). For the sake of simplicity, this contribution is presented for linear systems and colored observation noises. Nevertheless it can be easily extended to nonlinear systems (using an extended Kalman filter [5] or a unscented Kalman filter [6]) and to colored state noises (as explained in section III).

II. Kalman filtering

The filtering problem solved by Kalman is usually expressed in a so-called state-space formulation:

$$\begin{cases} \mathbf{x}_i = F_i \mathbf{x}_{i-1} + \mathbf{v}_i & \text{(process equation)} \\ y_i = H_i \mathbf{x}_i + n_i & \text{(observation equation)} \end{cases} \tag{1}$$

The state $\mathbf{x}_i \in \mathbb{R}^m$ has to be estimated from scalar observations $y_{1:i} = y_1, \ldots, y_i$ (the scalar assumption is made for ease of notation, without any loss of generality). State evolves according to the process matrix F_i and to the process noise $\mathbf{v}_i$, which is supposed to be white and independent of other quantities. Measurements y_i are done through the observation matrix H_i corrupted by some additive noise n_i, which is also supposed to be white and independent. Given these assumptions, the optimal estimate (in a minimum mean square sense) can be obtained recursively with the Kalman equations, which we briefly remind here. How to obtain them can be found in reference textbooks such as [7], [8]. The following notations are adopted: $\hat{\mathbf{x}}_{i|i}$ is the estimate at time i and $P_{i|i}$ the associated variance matrix, $\hat{\mathbf{x}}_{i|i-1}$ is the prediction of this estimate (and $P_{i|i-1}$ the associated variance), $P_{\mathbf{v}_i}$ (resp. P_{n_i}) is the variance of the noise $\mathbf{v}_i$ (resp. n_i), and K_i is the Kalman gain. Noises are supposed centered, and a prior $(\hat{\mathbf{x}}_{0|0}, P_{0|0})$ is mandatory.

Prediction equations:

$$\begin{cases} \hat{\mathbf{x}}_{i|i-1} = F_i \hat{\mathbf{x}}_{i-1|i-1} \\ P_{i|i-1} = F_i P_{i-1|i-1} F_i^T + P_{\mathbf{v}_i} \end{cases} \tag{2}$$

Kalman gain:

$$K_i = P_{i|i-1} H_i^T \left(H_i P_{i|i-1} H_i^T + P_{n_i} \right)^{-1} \tag{3}$$

correction equations:

$$\begin{cases} \hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + K_i \left(y_i - H_i \hat{\mathbf{x}}_{i|i-1} \right) \\ P_{i|i} = P_{i|i-1} - K_i \left(H_i P_{i|i-1} H_i^T + P_{n_i} \right) K_i^T \end{cases} \tag{4}$$

III. Auto-Regressive noise

The white noise assumption is sometimes too strong. A classical approach to take an AR noise into account is to extend the state with the noise [3]. Let u_i be a white noise, the AR noise n_i is defined as:

$$n_i + a_1 n_{i-1} + \cdots + a_p n_{i-p} = u_i \tag{5}$$

Let $\mathbf{a}$ be the set of AR parameters and $\mathbf{n}_i$ the set of AR noises from time i back to $i - p + 1$, that is two $p \times 1$ vectors defined as:

$$\mathbf{a} = (a_1, \ldots, a_p)^T \tag{6}$$

$$\mathbf{n}_i = (n_i, \ldots, n_{i-p+1})^T \tag{7}$$

Equation (5) can be written as:

$$n_i + \mathbf{a}^T \mathbf{n}_{i-1} = u_i \tag{8}$$

Let also $\mathbf{e}_i$ be a unitary column vector, that is $\mathbf{e}_i$ is zero everywhere except in its i^{th} component which is equal to 1. Let A be a $p \times p$ matrix defined as:

$$A = [\mathbf{a}, \mathbf{e}_1, \ldots, \mathbf{e}_{p-1}]^T \tag{9}$$

Equation (5) can also be written as:

$$\mathbf{n}_i = -A\mathbf{n}_{i-1} + \mathbf{e}_1 u_i \tag{10}$$

We define $\mathbf{0}_{p,q}$ the zero $p \times q$ matrix. State-space (1) with observation noise (5) is equivalent to the following state-space model, which can be solved with classic Kalman equations:

$$\begin{cases} \begin{pmatrix} \mathbf{x}_i \\ \mathbf{n}_i \end{pmatrix} = \begin{pmatrix} F_i & \mathbf{0}_{m,p} \\ \mathbf{0}_{p,m} & -A \end{pmatrix} \begin{pmatrix} \mathbf{x}_{i-1} \\ \mathbf{n}_{i-1} \end{pmatrix} + \begin{pmatrix} \mathbf{v}_i \\ u_i \mathbf{e}_1 \end{pmatrix} \\ y_i = \begin{pmatrix} H_i & \mathbf{e}_1^T \end{pmatrix} \begin{pmatrix} \mathbf{x}_i \\ \mathbf{n}_i \end{pmatrix} \end{cases} \tag{11}$$

This adds p components to the state. A very similar method (that is based on an extension of the state) exists to take into account an AR process noise (*e.g.*, see [8, chapter 7.2]). As the principle of our approach for MA (section IV) and ARMA (section V) noises is also to extend the state, it can be adapted the same way to colored (MA and ARMA) process noises.

IV. Moving-Average noise

Let the observation noise be an MA noise, u_i being still white:

$$n_i = u_i + b_1 u_{i-1} + \cdots + b_q u_{i-q} \tag{12}$$

Actually, the key fact to extend the state in the case of an AR observation noise is the possibility to express it recursively, see equation (10). Unfortunately, this is not (directly) possible for an MA noise, see equation (12). This is the main difficulty to consider an MA observation noise in the Kalman filtering framework. The key idea of this contribution is to introduce an auxiliary random process which purpose is to memorize the white noise u_i. This way, the MA noise can be formulated as a vectorial AR noise, and the technique described in section III can be used.

Let w_i be this auxiliary random process which aims at memorizing u_i, and let $\mathbf{w}_i$ the set of "memory" noises from time i back to $i-q+1$, that is the $q \times 1$ vector defined as:

$$\mathbf{w}_i = (w_i, \ldots, w_{i-q+1})^T \tag{13}$$

Let also $\mathbf{b}$ be the set of MA parameters (another $q \times 1$ vector):

$$\mathbf{b} = (b_1, \ldots, b_q)^T \tag{14}$$

We define W the $q \times q$ matrix as:

$$W = [\mathbf{0}_{q,1}, \mathbf{e}_1, \ldots, \mathbf{e}_{q-1}]^T \tag{15}$$

The MA noise defined in equation (12) is equivalent to the following vectorial AR noise:

$$\begin{pmatrix} n_i \\ \mathbf{w}_i \end{pmatrix} = \begin{pmatrix} 0 & \mathbf{b}^T \\ \mathbf{0}_{q,1} & W \end{pmatrix} \begin{pmatrix} n_{i-1} \\ \mathbf{w}_{i-1} \end{pmatrix} + \begin{pmatrix} 1 \\ \mathbf{e}_1 \end{pmatrix} u_i \tag{16}$$

Actually, from the upper bloc we have that $n_i = \mathbf{b}^T \mathbf{w}_{i-1} + u_i$ and from the lower bloc we have that $w_i = u_i$, which proves the equivalence.

Given the noise formulation (16), an equivalent to state-space (1) with MA observation noise can be proposed:

$$\begin{cases} \begin{pmatrix} \mathbf{x}_i \\ n_i \\ \mathbf{w}_i \end{pmatrix} = \begin{pmatrix} F_i & \mathbf{0}_{m,1} & \mathbf{0}_{m,q} \\ \mathbf{0}_{1,m} & 0 & \mathbf{b}^T \\ \mathbf{0}_{q,m} & \mathbf{0}_{q,1} & W \end{pmatrix} \begin{pmatrix} \mathbf{x}_{i-1} \\ n_{i-1} \\ \mathbf{w}_{i-1} \end{pmatrix} + \begin{pmatrix} \mathbf{v}_i \\ u_i \\ u_i \mathbf{e}_1 \end{pmatrix} \\ y_i = \begin{pmatrix} H_i & 1 & \mathbf{0}_{1,q} \end{pmatrix} \begin{pmatrix} \mathbf{x}_i \\ n_i \\ \mathbf{w}_i \end{pmatrix} \end{cases} \tag{17}$$

This adds $q+1$ components to the state.

V. Auto-Regressive Moving-Average noise

Consider now an ARMA observation noise defined as:

$$n_i + a_1 n_{i-1} + \cdots + a_p n_{i-p} = u_i + b_1 u_{i-1} + \cdots + b_q u_{i-q} \tag{18}$$

The tricky part is in the right member of equation (18), which corresponds to the MA component of the noise. Hopefully the technique developed in section IV can be used again. We define B the $p \times q$ matrix as:

$$B = [\mathbf{b}, \mathbf{0}_{q,p-1}]^T \tag{19}$$

Using B and the already introduced notations (9,15), noise (18) is equivalent to the following one:

$$\begin{pmatrix} \mathbf{n}_i \\ \mathbf{w}_i \end{pmatrix} = \begin{pmatrix} -A & B \\ \mathbf{0}_{q,p} & W \end{pmatrix} \begin{pmatrix} \mathbf{n}_{i-1} \\ \mathbf{w}_{i-1} \end{pmatrix} + \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_1 \end{pmatrix} u_i \tag{20}$$

This vectorial form of the noise model being defined, an equivalent to state-space (1) with ARMA observation noise can be proposed:

$$\begin{cases} \begin{pmatrix} \mathbf{x}_i \\ \mathbf{n}_i \\ \mathbf{w}_i \end{pmatrix} = \begin{pmatrix} F_i & \mathbf{0}_{m,p} & \mathbf{0}_{m,q} \\ \mathbf{0}_{p,m} & -A & B \\ \mathbf{0}_{q,m} & \mathbf{0}_{q,p} & W \end{pmatrix} \begin{pmatrix} \mathbf{x}_{i-1} \\ \mathbf{n}_{i-1} \\ \mathbf{w}_{i-1} \end{pmatrix} + \begin{pmatrix} \mathbf{v}_i \\ u_i \mathbf{e}_1 \\ u_i \mathbf{e}_1 \end{pmatrix} \\ y_i = \begin{pmatrix} H_i & \mathbf{e}_1^T & \mathbf{0}_{1,q} \end{pmatrix} \begin{pmatrix} \mathbf{x}_i \\ \mathbf{n}_i \\ \mathbf{w}_i \end{pmatrix} \end{cases} \tag{21}$$

This adds $p+q$ components to the state.

VI. Experiments

The first experiment consists in denoising a sinusoidal signal corrupted by some MA noise. This toy problem aims at illustrating the gain which can be obtained by considering the true noise model (compared to a white noise model). The second experiment deals with reinforcement learning, and explains why an MA observation noise model is necessary to unbias a given value function estimator.

A. Sinusoidal signal denoising

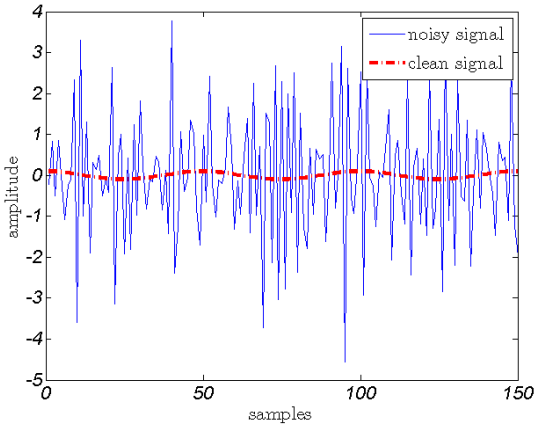

Fig. 1. Signals.

The aim of this experiment is to denoise a sinusoidal signal with unknown phase and amplitude corrupted by some moving-average noise. The observations are as follow:

$$y_i = G\cos(i\omega\Delta t + \varphi) + n_i \text{ with } n_i = u_i - u_{i-1} \tag{22}$$

The amplitude is chosen equal to $G = 0.1$, the pulsation to $\omega = \frac{2\pi}{5}$ and the sampling rate to $\Delta t = 0.1$. For each trial the phase is sampled randomly and uniformly between 0 and π: $\varphi \sim \mathcal{U}_{[0,\pi]}$. The white noise u_i is chosen to be Gaussian, centered and of unit variance: $u_i \sim \mathcal{N}(0,1)$. An example of clean and noisy signals is given in figure 1.

The goal being to denoise the corrupted signal, the chosen state representation is:

$$\mathbf{x}_i = \begin{pmatrix} G\cos(i\omega\Delta t + \varphi) & G\sin(i\omega\Delta t + \varphi) \end{pmatrix}^T \tag{23}$$

Using this state representation, the state-space formulation is obtained thanks to a rotation matrix:

$$\begin{cases} \mathbf{x}_i = \begin{pmatrix} \cos(\omega\Delta t) & -\sin(\omega\Delta t) \\ \sin(\omega\Delta t) & \cos(\omega\Delta t) \end{pmatrix} \mathbf{x}_{i-1} \\ y_i = \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{x}_i + n_i \end{cases} \tag{24}$$

Thanks to the method described in section IV, state-space (24) with MA observation noise (22) can be reformulated:

$$\begin{cases} \begin{pmatrix} \mathbf{x}_i \\ n_i \\ w_i \end{pmatrix} = \begin{pmatrix} F_i & \mathbf{0}_{2,1} & \mathbf{0}_{2,1} \\ \mathbf{0}_{1,2} & 0 & -1 \\ \mathbf{0}_{1,2} & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_{i-1} \\ n_{i-1} \\ w_{i-1} \end{pmatrix} + \begin{pmatrix} \mathbf{0}_{2,1} \\ u_i \\ u_i \end{pmatrix} \\ y_i = \begin{pmatrix} H_i & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_i \\ n_i \\ w_i \end{pmatrix} \end{cases} \tag{25}$$

Estimates obtained by the proposed augmented Kalman filter are compared to estimates obtained with a classic Kalman filter (which assumes a white observation noise). Performance is measured with the euclidian distance between the true state $\mathbf{x}_i$ and its estimation $\hat{\mathbf{x}}_{i|i}$, that is $\|\mathbf{x}_i - \hat{\mathbf{x}}_{i|i}\|$, averaged over 1000 trials. Results are presented in figure 2.

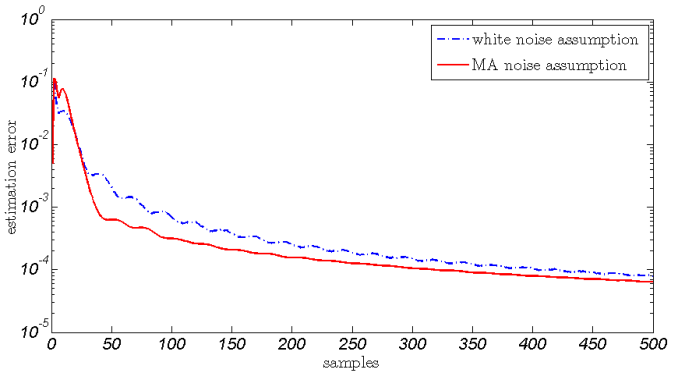

Fig. 2. Estimation error.

If the Kalman filter with a white noise assumption manages to give good estimates, it is clear from this experiment that correctly modeling the noise improves estimates quality. It is even more obvious in the next addressed problem.

B. Reinforcement Learning

This work was initially motivated by research in reinforcement learning (RL) where it finds an application to real world problems. RL [4] is the machine learning answer to the well-known problem of optimal control of dynamic systems. In this paradigm, an agent learns to control a dynamic system through examples of actual interactions. To each interaction is associated an immediate reward which is a local hint about the quality of the current control policy. More formally, at each (discrete) time step i the dynamic system to be controlled is in a configuration[1] s_i. The agent chooses an action a_i, and the dynamic system is then driven in a new configuration, say s_{i+1}, following its own dynamics. The agent receives a reward r_i associated to the transition (s_i, a_i, s_{i+1}). The agent objective is to maximize the expected cumulative rewards, which it internally models as a so-called value function. Kalman filtering has been shown to be very efficient to estimate this value function [9] and to provide a useful uncertainty information [10] but is a biased estimator in the case where transitions between system configurations are not deterministic. [11][9] explains this further.

The dynamic system is usually modeled as a Markov decision process (MDP) defined by the tuple $\{S, A, P, R, \gamma\}$. S is the finite configuration space, A the finite action space, $P : s, a \in S \times A \rightarrow p(.|s,a) \in \mathcal{P}(S)$ a family of transition probabilities, $R : S \times A \times S \rightarrow \mathbb{R}$ the bounded reward function, and γ the discount factor. A policy π associates to each configuration a probability over actions, $\pi : s \in S \rightarrow \pi(.|s) \in \mathcal{P}(A)$. The value function of a given policy is defined as $V^\pi(s) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, \pi]$ where r_i is the immediate reward observed at time step i, and the expectation is done over all possible trajectories starting in s given the system dynamics and the followed policy.

An important problem in reinforcement learning is the evaluation of a given policy, that is the estimation of its value function for every configuration. It is often the case that the configuration space is too large, and a compact representation of the value function has to be adopted. A parametric representation $\hat{V}^\pi_\theta(s)$ is often adopted. An algorithm to learn the best set of parameters θ is required and Kalman filtering is a good candidate. The hidden state to be tracked is the optimal parameter vector θ while transitions between configurations and rewards are observed. A state-space formulation has to be found. We do not consider any particular parameterization here but it can be linear, kernel-baesd, artificial neural networks or any other compact representation of a function.

Let's start with the observation equation. Let's define the random process

$$D^\pi(s) = \sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, \pi \tag{26}$$

First, it can be written as a Bellman-like recursion:

$$D^\pi(s) = R(s, A, S') + \gamma D^\pi(S') \tag{27}$$

with $A \sim \pi(.|s)$ and $S' \sim p(.|s, A)$. Second, the value function is actually the expectation of this process : $V^\pi(s) = E[D^\pi(s)]$. This random process can thus be broken down into the value function plus a random zero-mean residual:

$$D(s) = V^\pi(s) + \Delta V^\pi(s) \tag{28}$$

where $\Delta V^\pi(s) = D^\pi(s) - V^\pi(s)$. Manipulating the two previous equations, the reward can be expressed as a function of the value plus a noise:

$$R(s, a, s') = V^\pi(s) - \gamma V^\pi(s') + N(s, s') \tag{29}$$

with $N(s, s') = \Delta V^\pi(s) - \Delta V^\pi(s')$. This relates the reward function to the value function and the observation function

[1] Actually, in RL, a configuration is called a state. We name it differently in order to disambiguate from the Kalman's state.

can be derived as a sampled version of this equation : $r_i = \hat{V}_\theta^\pi(s) - \gamma \hat{V}_\theta^\pi(s') + n_i$.

Assuming that the residual $\Delta V^\pi(s)$ is white and since it is centered by definition, it can be modeled as a centered white noise u_i of theoretical variance $E[u_i^2] = \mathrm{var}(D(s_{i+1}))$. This leads to the following moving average (MA) noise model:

$$n_i = -\gamma u_i + u_{i-1}, \quad u_i \sim (0, \sigma_i^2) \tag{30}$$

which leads us to our point. Assuming a random walk as the evolution equation we end up with the following state-space equations:

$$\begin{cases} \theta_i = \theta_{i-1} + \mathbf{v}_i \\ r_i = \hat{V}_\theta^\pi(s) - \gamma \hat{V}_\theta^\pi(s') + n_i \end{cases} \tag{31}$$

Notice that even more general noise models can be envisioned in the RL setting [12].

State-space (31) with observation noise (30) can be handled thanks to the method described in section IV. Let I_m be the $m \times m$ identity matrix, m being the number of parameters. Handling the MA noise leads to the following equivalent state-space formulation:

$$\begin{cases} \begin{pmatrix} \theta_i \\ n_i \\ w_i \end{pmatrix} = \begin{pmatrix} I_m & \mathbf{0}_{m,1} & \mathbf{0}_{m,1} \\ \mathbf{0}_{1,m} & 0 & 1 \\ \mathbf{0}_{1,m} & 0 & 0 \end{pmatrix} \begin{pmatrix} \theta_{i-1} \\ n_{i-1} \\ w_{i-1} \end{pmatrix} + \begin{pmatrix} \mathbf{v}_i \\ u_i \\ -\gamma u_i \end{pmatrix} \\ r_i = \hat{V}_{\theta_i}^\pi(s_i) - \gamma \hat{V}_{\theta_i}^\pi(s_{i+1}) + n_i \end{cases} \tag{32}$$

Recall that linearity of parameterization is not mandatory. For example, the unscented transform [6] can be used, as in [11][9].

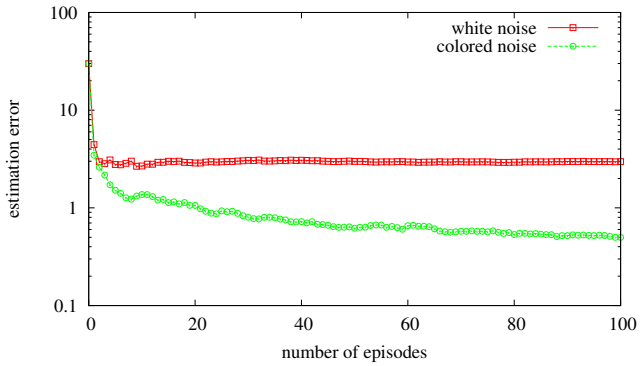

Fig. 3. Value function estimation.

To show the importance of taking the colored MA noise into account, we propose an experiment in which the value function is learnt on a simple 13-configurations valued Markov chain[2]. Configuration s^0 is absorbing, s^1 transits to s^0 with probability 1 and a reward of -2, and s^i transits to either s^{i-1} or s^{i-2}, $2 \leq i \leq 12$, each with probability 0.5 and reward -3. The value function is linearly parameterized: $\hat{V}_\theta(s) = \theta^T \phi(s)$. The feature vectors $\phi(s)$ for configurations s^{12}, s^8, s^4 and s^0 are respectively $[1,0,0,0]^T$, $[0,1,0,0]^T$, $[0,0,1,0]^T$ and $[0,0,0,1]^T$. The feature vectors for other configurations are obtained by linear interpolation. The optimal value function is linear in these features, and $\theta^* = [-24,-16,-8,0]^T$. The error measure is $\|\hat{\theta}_{i|i} - \theta^*\|$. The discount factor γ is set to 1 in this episodic task. Figure 3 shows the estimation error (averaged over 100 trials) when using a white observation noise and the colored observation noise (30) thanks to the method previously described. It is clear from this simple example that using a white noise leads to a biased estimate of the value function, while taking into account the specific structure of the colored moving-average noise leads to a unbiased estimate of the value function.

[2]An MDP with a fixed policy reduces to a valued Markov chain.

VII. Conclusion

We have introduced a generic and principled approach to handle moving-average noises in the Kalman filtering framework. The basic idea of this method is to introduce an auxiliary random variable which role is to memorize the white noise. This way, the moving-average noise can be expressed as a vectorial autoregressive noise, which allows taking it into account using a classical state extension. This method has also been extended to handle autoregressive moving-average noises. The efficiency of the proposed contribution has been experimentally demonstrated on a simple signal denoising problem. We have also briefly shown how taking a moving-average observation noise can be of great advantage in a reinforcement learning context.

References

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960. [Online]. Available: http://www.cs.unc.edu/ welch/kalman/kalmanPaper.html

[2] J. D. Gibson, B. Koo, and S. D. Gray, "Filtering of colored noise for speech enhancement and coding," *IEEE Transactions on Signal Processing*, vol. 39, no. 8, pp. 1732–1742, 1991.

[3] A. Bryson and D. Johanson, "Linear filtering for time-varying systems using measurements containing colored noise," *IEEE Transactions on Automatic Control*, vol. 10, no. 1, pp. 4–10, 1965.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 3rd ed. The MIT Press, March 1998. [Online]. Available: http://www.cs.ualberta.ca/ sutton/book/the-book.html

[5] S. F. Schmidt, "Computational techniques in Kalman filtering," NATO Advisory Group for Aerospace Research and Development, London, in: Theory and Application of Kalman Filtering, AGARDograph 139, 1970.

[6] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *International Symposium on Aerospace/Defense Sensing, Simulation and Controls 3*, 1997.

[7] M. S. Grewal and A. P. Andrew, *Kalman Filtering: Theory and Practice*. Prentice Hall, 1993.

[8] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley & Sons, August 2006. [Online]. Available: http://academic.csuohio.edu/simond/

[9] M. Geist and O. Pietquin, "Kalman Temporal Differences," *Journal of Artificial Intelligence Research (JAIR)*, vol. 39, pp. 483–532, October 2010.

[10] ——, "Managing Uncertainty within Value Function Approximation in Reinforcement Learning," in *Active Learning and Experimental Design workshop (collocated with AISTATS 2010)*, Sardinia, Italy, 2010.

[11] M. Geist, O. Pietquin, and G. Fricout, "Tracking in reinforcement learning," in *Proceedings of the 16th International Conference on Neural Information Processing (ICONIP 2009)*. Bangkok, Thailand: Springer, December 2009.

[12] M. Geist and O. Pietquin, "Eligibility Traces through Colored Noises," in *Proceedings of the IEEE International Conference on Ultra Modern Control systems (ICUMT 2010)*, Moscow (Russia), October 2010.