Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Probability Density Function Control for Stochastic Nonlinear Systems using Monte Carlo Simulation

Qichun Zhang * **Hong Wang** **

* *Department of Computer Science, University of Bradford, Bradford, BD7 1DP, UK. (e-mail: q.zhang17@bradford.ac.uk)*
** *Energy and Transportation Science Division, Oak Ridge National Laboratory, Oak Ridge, TN37831, USA.*

Abstract: This paper presents an implementable framework of output probability density function (PDF) control for a class of stochastic nonlinear systems which are subjected to non-Gaussian noises. The statistical properties of the system outputs can be adjusted by shaping the dynamic output probability density function to track the reference stochastic distribution. However, the dynamic probability density function evolution is very difficult to obtain analytically even if the system model and the stochastic distributions of the noises are known. Motivated by Monte Carlo simulation, the dynamic probability density function can be estimated by sampling data which forms the contribution of this paper. In particular, the sampling points are generated following the stochastic distribution of the noise for each instant. These points go through the system and generate the histogram for system outputs, then the dynamic model can be established based on the dynamic histogram which reflects the randomness and the nonlinear dynamics of the investigated system. Based on the established model, the output probability density function tracking can be achieved and the simulation results and discussions show the effectiveness and benefits of the presented framework.

Keywords: Probability density function control, stochastic nonlinear systems, non-Gaussian distribution, Monte Carlo simulation

1. INTRODUCTION

A probability density function reflects the complete stochastic properties of a random variable. It is an important research topic of the extended form of the traditional stochastic control such as minimum variance control(Åström (2012)), covariance control(Zhang et al. (2016),) etc. Since many practical industrial processes are subjected to non-Gaussian noises(Wang (2012)), the Gaussian-assumption-based stochastic control methods cannot be adopted properly. Therefore, investigating the probability density function control problem for non-Gaussian stochastic system is significant both for control theory and real applications (Ren et al. (2019)). In particular, the probability density function control would influence many theoretical research topics such as minimum entropy control(Ren et al. (2013)), non-Gaussian filtering(Yin et al. (2020)), particle filtering(Liu et al. (2019)), probabilisitic decoupling(Zhang et al. (2017)), performance enhancement(Zhou et al. (2017)), etc. As an example of real application, the probability density function control would improve the product quality of paper making process (Wang et al. (2001)).

To control the output probability density function, the analytical formulation of the probability density function has to be obtained for each instant. There are two different approaches: 1) Guo and Wang (2010) presented the model-based direct evolution method where the output probability density function can be formulated using Jacobian matrix. The shortcoming of this method is that only a class of system model satisfies the evolution condition and the inverse calculation is expensive for high-dimensional model. And 2) Estimating the probability density function using probability density estimation (Zhang and Hu (2018)) with the collected output data along the instant, where the accuracy cannot be guaranteed as the dynamic system would lead to an un-stationary process thus the estimation performance will deteriorate. As a summary of the aforementioned methods, the problem can be solved if the dynamic probability density function can be represented by the estimable form for each instant.

Monte Carlo methods have been widely used as an effective numerical estimation(see Rubinstein and Kroese (2016)). For dynamic systems, the randomness can be simulated by Monte Carlo simulations where the sampling points can be generated for each instant. Recursively, the dynamics

⋆ This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

Copyright lies with the authors

of the system can be characterised by these sampling points. Since the sampling points can be used to estimated the stochastic distribution, Monte Carlo methods would supply a fast solution for probability density function control implementation. In particular, the sampling points can be generated for each instant following the known distribution of random noise. Using these sampling points and the system model, the numerical simulation can be done for each instant thus the histogram of the system output can be obtained in real-time. The segments of the histogram can be considered as various events and the events can be re-used as the states of the probability density function model. In other words, the dynamic histogram results in a state-space model to describe the dynamics of the output probability density function.

Following the idea above, the histogram can be used to establish a model while the parameters of the model can be identified using the least-square method directly with the obtained dynamic information of the histogram. Thus the control design can be achieved using any existing methods. In this paper, the parametric state feedback has been adopted as a case demonstration of the presented framework. The effectiveness of the presented framework is also demonstrated by the simulation results with a numerical example. In addition, discussions are given after the simulation to analyse the performance of the present framework, including convergence, direct optimisation method, and the relationship to the B-spline neural network model, etc.

The rest of this paper has been organised as follows: The problem has been described and formulated in Section 2, where the histogram-based representative has been developed. In Section 3, the modelling procedure has been given using the calculated histogram. The probability density function tracking algorithm has been achieved in Section 4 as the main result of the presented framework and the pseudo-code is also given for implementation. The simulation and discussion have been shown in Section 5 and Section 6, respectively. In particular, the validation has been demonstrated and the theoretical analysis has been analysed, e.g. convergence, optimisation and discussion about B-spline neural network model. In the end, section 7 summarises this paper as conclusion.

2. FORMULATION

Considering the following general stochastic nonlinear system model:

$$\begin{aligned} x_{k+1} &= f\left(x_k, u_k, w_k\right) \\ y_k &= h\left(x_k\right) \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}$, $u \in \mathbb{R}$ denote the n-dimensional vector-valued system state, system output and control input. w stands for the non-Gaussian random noise and k denotes the index for discrete-time sample instant. Suppose that the nonlinear functions $f(\cdot)$ and $h(\cdot)$ are known. The stochastic distribution of the random noise w is known as γ_w.

The sampling points following γ_w can be described as $\{\sigma_{1,k}, \sigma_{2,k}, \dots, \sigma_{N,k}\}$ where positive integer N denotes the size of the sampling set of γ_w for each k. Therefore, the samples for system output y_k can be described by a data set as $\{y_{\sigma_1,k}, y_{\sigma_2,k}, \dots, y_{\sigma_N,k}\}$. In practice, the events can be pre-defined based on the boundary values α_i. In particular, $\alpha_1 < \alpha_2 < \cdots < \alpha_m$ results in $m+1$ segments in sample space of y_k. Then the histogram can be obtained while the probability values can be calculated as $P(y < \alpha_1)$, $P(\alpha_1 < y < \alpha_2)$, $\dots$, $P(\alpha_m < y)$. $P(\cdot)$ stands for the probability value which can be obtained by counting the number of the sampling points in the segment dividing N.

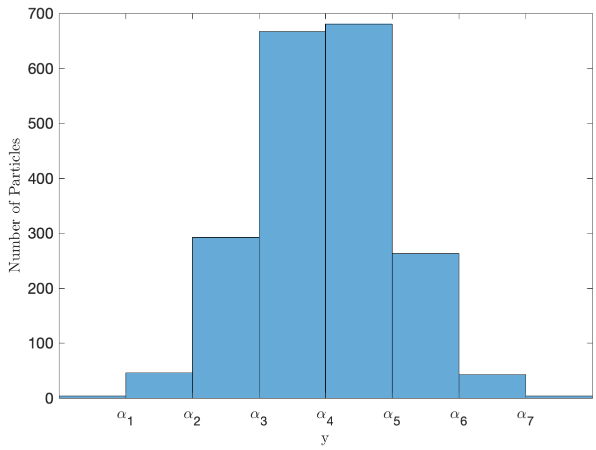

Fig. 1. The histogram of Gaussian distribution, where $\alpha_1, \dots, \alpha_7$ denotes the pre-specified segments. In particular, $m = 7$ and the point numbers within the segments result in the probabilities.

For each instant, the histogram can be generated and an example is shown by Fig.1, where the probability value can be calculated simply. Notice that the integral of probability density function over the sample space is equal to 1, we have

$$P\left(y \le \alpha_1\right) + P\left(\alpha_m < y\right) + \sum_{i=1}^{m-1} P\left(\alpha_i < y \le \alpha_{i+1}\right) = 1 \tag{2}$$

which indicates that there are m events which are independent out of $m+1$ events defined by segments. Therefore, the probability density function can be represented by m probability values. Furthermore, the stochastic properties of the system output y_k for each instant k can be characterised by a probability vector as follows:

$$W_k = \left[P\left(y \le \alpha_1\right), \dots, P\left(\alpha_{m-1} < y \le \alpha_m\right)\right]^T \tag{3}$$

Based on the pre-specified $m+1$ segments, the reference vector can be converted from the reference probability density function, for example,

$$P\left(\alpha_{m-1} < y \le \alpha_m\right) = \int_{\alpha_{m-1}}^{\alpha_m} \gamma_{ref}\left(y\right) dy \tag{4}$$

where γ_{ref} denotes the reference probability density function. Thus, the reference vector W_{ref} can be obtained from γ_{ref} . Note that the reference is a vector which is independent of k.

To track the given reference, the following equation should be achieved using the controller design,

$$\lim_{k \to \infty} \left(W_k - W_{ref}\right) = 0 \tag{5}$$

which forms the objective of the investigated framework. In particular, the distribution tracking error can be represented by the vector error.

3. HISTOGRAM-BASED MODELLING

Based on Kolmogorov forward equation, the stochastic distribution is governed by a stochastic partial differential equation, which can be simplified as a stochastic differential equation for each instant. As a discrete-time form, the following equation can be generated.

$$\bar{W}_{k+1} = g\left(\bar{W}_k, u_k\right) \tag{6}$$

where $g(\cdot)$ denotes an unknown differentiable nonlinear function and $\bar{W} = \left[W^T, P\left(\alpha_m < y\right)\right]^T$. Basically, the vector $\bar{W}$ stands for the complete information of the output probability density function.

Eq. (6) leads to the following equation using linearisation operation,

$$\bar{W}_{k+1} = A_k \bar{W}_k + B_k u_k + \bar{g}\left(\bar{W}_k\right) \tag{7}$$

where $A_k \in \mathbb{R}^{(m+1)\times(m+1)}$ and $B_k \in \mathbb{R}^{(m+1)}$ are the time-variant coefficient matrices for instant k. Nonlinear function $\bar{g}(\cdot)$ denotes the un-modelled dynamics. In particular, we have

$$\{A_k, B_k\} = \left\{\frac{\partial g}{\partial \bar{W}}, \frac{\partial g}{\partial u}\right\}\Bigg|_{\bar{W}=\bar{W}_k, u=u_k} \tag{8}$$

Note that the dimension of the state vector can be reduced to m based on Eq.(2). The following decomposition can be considered

$$\begin{bmatrix} W_{k+1} \\ P\left(\alpha_m < y_{k+1}\right) \end{bmatrix} = \begin{bmatrix} A_{11,k} & A_{12,k} \\ A_{21,k} & A_{22,k} \end{bmatrix} \begin{bmatrix} W_k \\ P\left(\alpha_m < y_k\right) \end{bmatrix} + \begin{bmatrix} B_{1,k} \\ B_{2,k} \end{bmatrix} u_k + \bar{g}\left(\bar{W}_k\right) \tag{9}$$

Thus the reduced-order model can be obtained as follows:

$$W_{k+1} = A_{11,k} W_k + A_{12,k} P\left(\alpha_m < y_k\right) + B_{1,k} u_k + \bar{g}_m(\bar{W}_k) \tag{10}$$

where $A_{11,k} \in \mathbb{R}^{m\times m}$, $B_k \in \mathbb{R}^m$, and $\bar{g}_m$ denotes the sub-function of $\bar{g}$ with m-dimensional outputs.

To generalise and simplify the expression, the following equation can be used as the complete form of the presented histogram-based model.

$$W_{k+1} = \bar{A}_k W_k + \bar{B}_k u_k + \delta(\bar{W}_k) \tag{11}$$

where $\delta(\bar{W}_k) = A_{12,k} P\left(\alpha_m < y_k\right) + \bar{g}_m(\bar{W}_k)$, $\bar{A}_k = A_{11,k}$, and $\bar{B}_k = B_{1,k}$.

Note that the value of function $\delta(\bar{W}_k)$ can be arbitrarily small if the dimension number m goes sufficiently large. Moreover, the $P\left(\alpha_m < y_k\right)$ can be very close to zero with the pre-specified event boundaries. The nonlinear function $\delta(\bar{W}_k)$ can be considered as a disturbance of the model while the robustness of the controller design would eliminate the effect of this nonlinear function. The matrix $\bar{A}_k$ can be considered as a probability transition matrix. It implies that the presented model is given with explicit physical meaning.

The parametric identification is the next step for completing the modelling procedure. Since the presented model is linear, the model can be rewritten as follows:

$$W_{k+1} = \theta_k \Phi_k \tag{12}$$

where $\theta_k = \left[\bar{A}_k, \bar{B}_k\right]$ denotes the unknown parameter matrix and $\Phi_k = \left[W_k^T, u_k\right]^T$ stands for the known information for parameter identification.

Based upon Eq.(12), the recursive least square (RLS) method (see Björck (1996)) can be adopted directly. In particular, the algorithm has been recalled as follows:

$$\begin{aligned} \theta_{k+1} &= \theta_k + \frac{P_{k-1}\Phi_k \varepsilon_k}{1 + \Phi_k^T P_{k-1} \Phi_k} \\ \varepsilon_k &= W_{k+1} - \theta_k \Phi_k \\ P_k &= \left(I - \frac{P_{k-1}\Phi_k}{1 + \Phi_k^T P_{k-1}\Phi_k}\right) \times P_{k-1} \end{aligned} \tag{13}$$

4. PDF TRACKING

To track the given reference distribution, the error model can be defined comparing the reference W_{ref} and the state W_k. An integrator has been introduced into the control design where we have $e_{k+1} = e_k + W_{ref} - W_k$. Thus e can be considered as the extended state for the model and the complete form of the error model is obtained by substituting the integrator.

$$\begin{bmatrix} e_{k+1} \\ W_{k+1} \end{bmatrix} = A_{e,k} \begin{bmatrix} e_k \\ W_k \end{bmatrix} + B_{e,k} u_k + \begin{bmatrix} 0 \\ \delta(\bar{W}_k) \end{bmatrix} + \begin{bmatrix} W_{ref} \\ 0 \end{bmatrix} \tag{14}$$

where $A_{e,k} = \begin{bmatrix} I & -I \\ 0 & \bar{A}_k \end{bmatrix}$ and $B_{e,k} = \begin{bmatrix} 0 \\ \bar{B}_k \end{bmatrix}$.

Using the developed model (14), any existing control methods can be inserted into this framework. As a case study, the parametric state feedback method has been used in this paper as the design parameters can be further optimised to increase the robustness of the control strategy.

In particular, the controller can be obtained as follows:

$$u_k = K_k \left[e_k^T, W_k^T\right]^T \tag{15}$$

where $K_k \in \mathbb{R}^{2m}$ denotes the gain vector of the state feedback.

Based on the parametric state feedback, which was presented by Roppenecker (1986), the feedback gain K_k can be calculated as follows:

$$\begin{aligned} K_k = &[M_1 f_1, \ldots, M_{2m} f_{2m}] \\ &\times \Big[\left(\lambda_1^* I - A_{e,k,1}\right)^{-1} B_{e,k,1} f_1, \ldots, \\ &\qquad \left(\lambda_{2m}^* I - A_{e,k,2m}\right)^{-1} B_{e,k,2m} f_{2m}\Big]^{-1} \end{aligned} \tag{16}$$

In the case of a common open-loop and closed-loop eigenvalue, other parameters in Eq.(16) can be determined as follows:

$$\begin{aligned} A_{e,k,i} &= A_{e,k} + v_j^0 {s_j^0}^T \\ M_i &= I - \frac{d_{\bar{k}} {s_j^0}^T B_{e,k}}{{s_j^0}^T b_{\bar{k}}} \\ B_{e,k,i} &= B_{e,k} M_i + v_j^0 d_{\bar{k}}^T \end{aligned} \tag{17}$$

where v_j^0 and $s_j^0\,(j = 1, \ldots, 2m)$ denote the open-loop eigenvectors and eigenrows of the model (14). $b_{\bar{k}}$ is the $\bar{k}$-th column of the matrix $B_{e,k}$. $d_{\bar{k}}$ is a unit vector where the $\bar{k}$-th element is 1. In the other case, there is no common eigenvalue, $s_j^{0^T} b_{\bar{k}} = 0$, so that the equations (17) can be simplified as follows:

$$\begin{aligned} A_{e,k,i} &= A_{e,k} \\ M_i &= I \\ B_{e,k,i} &= B_{e,k} \end{aligned} \tag{18}$$

Based on the control design, the error vector between the reference W_{ref} and W_k will converge to zero with arbitrary parameters $f_1, f_2, \ldots, f_{2m}$ and the probability density function tracking will be achieved. To summarise the presented framework, the following block diagram (see Fig.2) is given with pseudo-code for implementation.

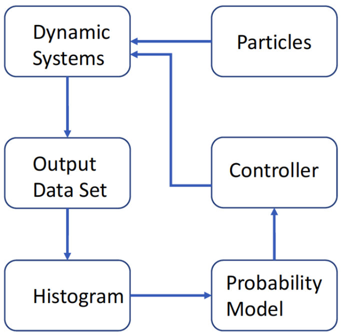

Fig. 2. The block diagram for the presented probability distribution tracking framework.

5. SIMULATION

To validate the presented framework, the following system model can be considered as a numerical example.

$$\begin{aligned} x_{k+1} &= 0.5 x_k \sin x_k + u_k + w_k \\ y_k &= 2 x_k \end{aligned} \tag{19}$$

where w_k is random noise subjected to Gamma distribution with shape and scale factors are equal to 2.

For tracking purpose, the reference distribution can be given using a set of pre-specified probabilities where $W_{ref} = [0, 0, 0, 0, 0, 0.05, 0.1, 0.15, 0.6]$ while the segment boundaries are defined as $[-4, -3, -2, -1, 0, 1, 2, 3, 4]$.

To obtain the histogram of the system output, 500 points have been generated using Monte Carlo simulation. Based on the design procedure, the simulation results can be indicated here by the following figures. In particular, Fig. 3 shows the vector value goes to the reference vector along the instant k which implies that tracking of the given probability density function is achieved. Meanwhile, the histograms of vector W_k and tracking error e_k are also given in Fig. 4 and Fig. 5, respectively. The figures demonstrate that the tracking error has been controlled within an acceptable error and the tracking performance has been achieved. In addition, the state value of the investigated system as indicated by Fig. 6 shows the system state is not convergent as the reference probability-based vector does not have zero mean value, however the system state is still bounded which means that the real implementation is feasible and the stability of the investigated system can be guaranteed. Further information on convergence will be analysed in next Section.

Algorithm 1 Pseudo code for implementing the presented probability density function tracking framework

---

Require: System model (1) and the distribution of the noise w

Input: The pre-specified segments/events with given boundaries $\alpha_1, \ldots, \alpha_m$ and the point number N for Monte Carlo simulation

Output: The final histogram of the system output y_k

Initialisation: Setup the operation time t_s and initial value for the system model.

for $k \leq t_s$ **do**

  Generating the point set obeys the given stochastic distribution of the noise.

  for $i \leq N$ **do**

    Put the point through the system model and update the data set for system output y_i.

  end for

  Obtain the histogram of the system output at instant k using the collected N points for system output.

  Calculate the probability values for the pre-specified segments and form them as a vector W_k.

  Using the recursive least square method to identify the probability model.

  Based on the parameters of the identified model construct the parametric controller.

  Select one sampling point as the measured noise w_k

  Substituting the control input value u_k into the investigated system model with w_k and update the system states and output.

  $k \leftarrow k + 1$

end for

Obtain the actual final state W_k and convert W_k to probability density function.

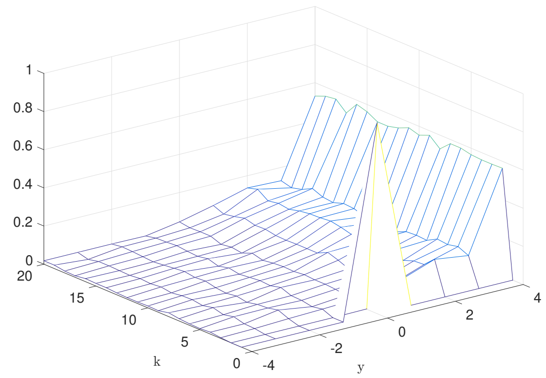

Fig. 3. The value of the state vector W_k where the probability for each segment/event is shifting along sampling instant k.

6. PERFORMANCE ANALYSIS

6.1 Convergence

Firstly, we can assume that the identified model (11) is equivalent to the investigated system model (1) in terms of the output probability density function. Thus the convergence of the system output analysis can be re-

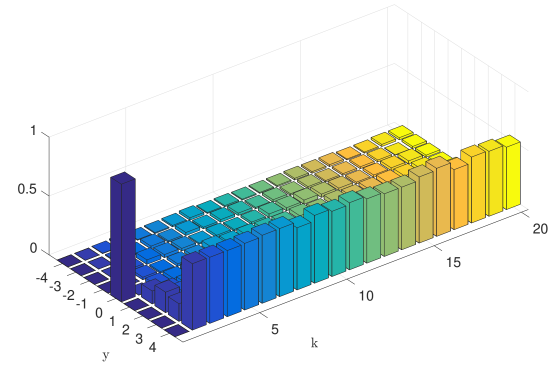

Fig. 4. The histogram of the probabilities where the result is consistent with the result shown by Fig. 3.

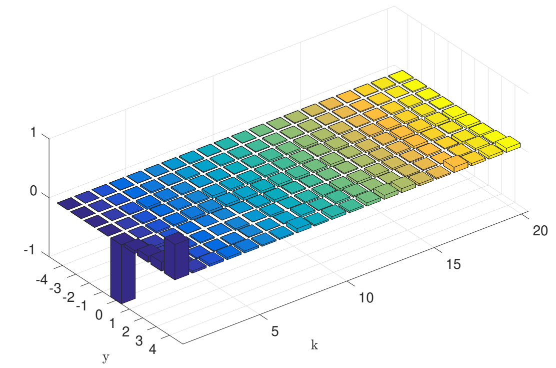

Fig. 5. The histogram of the tracking error e, the tracking performance can be validated as e has been attenuated very close to zero

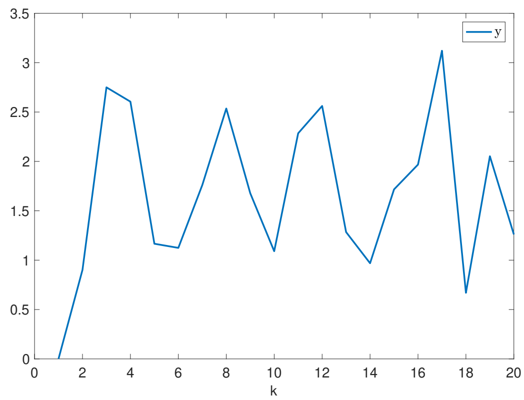

Fig. 6. The state value of the investigated system (19) is shown by curve where the bounded state implies the bounded system output y.

expressed as the stability analysis for closed-loop system model with control law (15).

Based on Eq.(14), the investigated stochastic system has been converted to deterministic system. Furthermore, the control law can be decomposed as follows:

$$u_k = \left[K_{e,k}, \bar{K}_k\right] \times \left[e_k^T, W_k^T\right]^T \tag{20}$$

where implies that

$$\begin{aligned} B_{e,k} u_k &= \begin{bmatrix} 0 \\ \bar{B}_k \end{bmatrix} \left[K_{e,k}, \bar{K}_k\right] \begin{bmatrix} e_k \\ W_k \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \bar{B}_k K_{e,k} e_k + \bar{B}_k \bar{K}_k W_k \end{bmatrix} \end{aligned} \tag{21}$$

Substituting the above equation, the closed-loop formulation can be simplified as follows:

$$W_{k+1} = \left(\bar{A}_k + \bar{B}_k \bar{K}_k\right) W_k + \delta\left(\bar{W}_k\right) + \bar{B}_k K_{e,k} e_k \tag{22}$$

Notice that the parametric feedback can be designed as $A_{e,k}$ Hurwitz then the variable e_k is convergent. In other words, there always exists a positive real upper limit Θ_1, such that, $\|e_k\| \leq \Theta_1 \|W_k\|$. In addition, as aforementioned above, $\delta\left(\bar{W}_k\right)$ can be bounded as arbitrary small number. Similarly, there also exists another positive real upper limit Θ_2, such that, $\left\|\delta\left(\bar{W}_k\right)\right\| \leq \Theta_2 \|W_k\|$.

As a result, we have

$$\|W_{k+1}\| \leq \left(\left\|\bar{A}_k + \bar{B}_k \bar{K}_k\right\| + \Theta_2 + \left\|\bar{B}_k K_{e,k} \Theta_1\right\|\right) \|W_k\| \tag{23}$$

which leads to the following convergence condition.

$$\left\|\bar{A}_k + \bar{B}_k \bar{K}_k\right\| + \Theta_2 + \left\|\bar{B}_k K_{e,k} \Theta_1\right\| \leq 1 \tag{24}$$

Note that $\bar{A}_k$, $\bar{B}_k$ and K_k are bounded thus the condition is implementable.

The convergence also implies that the moment of the investigated output y_k is bounded. It shows that y_k is bounded as a random variable which completes the analysis of the output convergence.

6.2 Direct optimisation

Note that the controller design is still based on the identified model which leads to two main problems: 1) the identification error has to be introduced into the controller design, and 2) the computational loading has been increased strongly which reduces the on-line performance of the implementation in real time. To solve the problems, the direct optimisation can be considered.

The control problem is also an optimisation problem which can be described as searching for the optimal control input signals to minimum the tracking error, where the cost function can be formulated as follows:

$$J_k = \min_{u_k} \left\{W_{ref} - W_k\right\} \tag{25}$$

To achieve the minimisation, the gradient descent algorithm can be used with a pre-specified learning rate ϵ.

$$u_k = u_{k-1} + \epsilon \left.\frac{\partial J}{\partial u}\right|_{J=J_k, u=u_k} \tag{26}$$

In practice, a penalty term would be added into the cost function to guarantee the local convergence of the gradient descent searching, moreover, the difference operation can be used to replace the differentiation using the discrete-time format.

$$u_k = u_{k-1} + \epsilon \frac{\partial \bar{J}_{k-1} - \bar{J}_{k-2}}{\partial u_{k-1} - u_{k-2}} \tag{27}$$

where

$$\bar{J}_k = J_k + \frac{1}{2} R u_k^2 \tag{28}$$

while $R \in \mathbb{R}$ denotes the penalty weight of the control input u_k.

Using the direct numerical optimisation, the control input can be calculated for each instant however the convergence analysis is difficult to obtain and only local optimisation will be achieved since the states W_k are governed by the nonlinear dynamics of the investigated stochastic systems.

6.3 B-spline neural network model

Wang (2012) presented a decoupled model using a B-spline neural network, where the probability density function has been rewritten as follows based upon a set of weight values v with B-spline base functions $B(\cdot)$.

$$\gamma(y) = \sum_{i=1}^{m} v_i B_i(y) \tag{29}$$

As a result, the dynamics of the investigated system has been reflected by the dynamics of the weight vector which leads to the dynamic model for the probability density function.

$$\begin{aligned} V_{k+1} &= AV_k + Bu_k \\ \gamma(y) &= CV_k \end{aligned} \tag{30}$$

where $C = [B_1(y), \ldots, B_m(y)]$. A and B are the parametric matrices. Based on this format, the rational PDF model, square-root PDF model, etc. can be obtained. The implementation for the control design using these models need to identify the weights for each instant which restricts the real-time requirements.

Note that the histogram is the zero-order estimation of the probability density function using a B-spline function which means that the histogram-based model in this paper is a special case of the B-spline neural network model. However the Monte Carlo simulation gives the histogram directly without identification and achieves the fast implementation which shows the novelty of this paper.

7. CONCLUSION

The output probability density function control problem has been investigated in this paper. Comparing with all the existing results, the probability-based state space model has been developed as a fast implementation for probability density function control where the explicit physical meaning is also obtained. The complete model can be established with parametric identification then the control methods can be inserted into this new framework to achieve the design objective. As a demonstration, the parametric state feedback method has been used for controller design due to the fact that the free design parameters would supply more flexibilities for robustness requirements. Moreover, the performance analysis has been given. In particular, the stability of the presented algorithm has been analysed briefly and a direct optimisation method has been discussed. In addition, we claim that the presented model is a special case of B-spline model for fast implementation. To validate the effectiveness of the presented probability density function control framework, the numerical simulation results have been obtained where all the curves and figures show that the novel design has achieved the design objective.

REFERENCES

Åström, K.J. (2012). *Introduction to stochastic control theory.* Courier Corporation.

Björck, Å. (1996). *Numerical methods for least squares problems.* SIAM.

Guo, L. and Wang, H. (2010). *Stochastic distribution control system design: a convex optimization approach.* Springer Science & Business Media.

Liu, C., Li, B., and Chen, W.H. (2019). Particle filtering with soft state constraints for target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 55(6), 3492–3504.

Ren, M., Zhang, J., Fang, F., Hou, G., and Xu, J. (2013). Improved minimum entropy filtering for continuous nonlinear non-gaussian systems using a generalized density evolution equation. *Entropy*, 15(7), 2510–2523.

Ren, M., Zhang, Q., and Zhang, J. (2019). An introductory survey of probability density function control. *Systems Science & Control Engineering*, 7(1), 158–170.

Roppenecker, G. (1986). On parametric state feedback design. *International Journal of Control*, 43(3), 793–804.

Rubinstein, R.Y. and Kroese, D.P. (2016). *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons.

Wang, H., Baki, H., and Kabore, P. (2001). Control of bounded dynamic stochastic distributions using square root models: an applicability study in papermaking systems. *Transactions of the Institute of Measurement and Control*, 23(1), 51–68.

Wang, H. (2012). *Bounded dynamic stochastic systems: modelling and control.* Springer Science & Business Media.

Yin, X., Zhang, Q., Wang, H., and Ding, Z. (2020). Rbfnn-based minimum entropy filtering for a class of stochastic nonlinear systems. *IEEE Transactions on Automatic Control*, 65(1), 376–381.

Zhang, Q. and Hu, L. (2018). Probabilistic decoupling control for stochastic non-linear systems using ekf-based dynamic set-point adjustment. In *2018 UKACC 12th International Conference on Control (CONTROL)*, 330–335. IEEE.

Zhang, Q., Wang, Z., and Wang, H. (2016). Parametric covariance assignment using a reduced-order closed-form covariance model. *Systems Science & Control Engineering*, 4(1), 78–86.

Zhang, Q., Zhou, J., Wang, H., and Chai, T. (2017). Output feedback stabilization for a class of multi-variable bilinear stochastic systems with stochastic coupling attenuation. *IEEE Transactions on Automatic Control*, 62(6), 2936–2942.

Zhou, Y., Zhang, Q., Wang, H., Zhou, P., and Chai, T. (2017). Ekf-based enhanced performance controller design for nonlinear stochastic systems. *IEEE Transactions on Automatic Control*, 63(4), 1155–1162.