Weighted Multiple Model Adaptive Control of Slowly Time-Varying Systems

Weicun Zhang, Qing Li

School of Automation and Electrical Engineering, University of Science and Technology Beijing Beijing 100083, China

E-mails: weicunzhang@ustb.edu.cn; qingli@ies.ustb.edu.cn

Abstract

Weighted multiple model adaptive control is a combination of off-line design and on-line decision, which combines a finite number of simple controllers by weighting algorithm. This paper presents an improved weighting algorithm of slowly time-varying systems and a new method to partition model uncertainty using v-Gap metric. The effectiveness of the proposed methods is verified through Matlab simulations.

Keywords: MMAC; v-Gap; Weighting Algorithm; Time-Varying System

1. INTRODUCTION

Weighted multiple model adaptive control (WMMAC) and weighted multiple model adaptive estimation (WMMAE) appeared around 1960's to 1970's, which consists mainly of three components, i.e., the model-set, local controller/estimator design, and weighting algorithm. Multiple Kalman filter is the basis of classical MMAC or MMAE. The purpose of using multiple models is to improve the accuracy of state estimation or control problems with parameter uncertainties[1, 2]. This was followed in later years by several practical applications. In recent years, a new type of WMMAC, i.e., robust multiple model adaptive control (RMMAC) was put forward with convincing experiment results[3,4], which arouses once again the enthusiasm of researchers in the field of adaptive control in recent years.

Some results are made on MMAC of time-invariant systems. MMAC is proved to be a useful control method to deal with time-invariant system whose parameter is not quite certain.

As previous weighting algorithm could not easily change weight once sub-model's weight converged to 0, it is not suitable for time-varying system. This paper comes up with an improved weighting algorithm to solve this problem. With the help of improved algorithm, the changes of system parameters can be detected and then sub-models' weights can be recalculated. In this way, WMMAC can be used to slow-varying systems.

For WMMAC, weighting algorithm is one of the core contents, it will directly affect whether the weighting algorithm could choose the sub-model from model-set which is the nearest to the actual control system accurately and rapidly.

The corresponding WMMAC system is shown in Fig. 1 and Fig. 2.

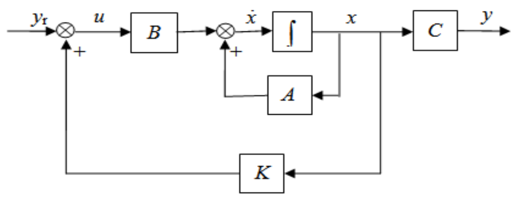

Fig.1 Block diagram of a general control system

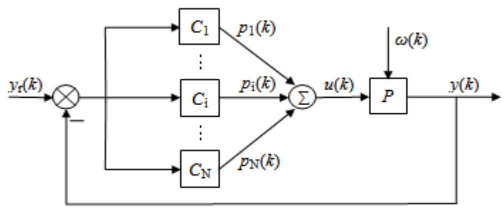

Fig. 2 Block diagram of WMMAC

Consider the following time-varying plant:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$
$$y(t) = C(t)x(t)$$

2. Weighting algorithm

The traditional weighted multi-model adaptive control use maximum a posteriori probability method to calculate the weights of each controller. In the process of calculation, traditional method is quite complex and its calculation is huge.

When one sub-model of model-set is very close to the actual system, its weight should be close to 1. That's the core concept of weighted multi-model adaptive control method[5]. As the change of weight is smooth rather than mutations, so we could achieve the purpose of model soft switching by using weighted multiple model adaptive control method.

As the actual controlled system and sub-model of model-set is not necessarily the same, the output error $e_i(k)$ is exist between actual system output $y(k)$ and each sub-model's output $y_i(k)$.

Model's similar degree is evaluated by the magnitude of output error $e_i(k)$. If one of the sub-model's output error is the smallest when system became stable, then we can say this sub-model is the nearest to the actual system and its weight should finally converge to 1

Based on the above concept, designing the following weighting algorithm:

1) Initialize weights.
2) Calculation of error performance metrics.
3) Compare the error performance metrics, find the minimum error performance metrics.
4) Calculation of weight metrics.
5) Calculation of sub-models' weights based on weight metrics.
6) Calculation of control law.

The advantage of this method is smooth switching process and stabilized system output. This could decrease the damage to actual system when system parameter changed, as a result this method is quite suitable for industrial.

The weight metrics and sub-models' weights of weighting algorithm are designed as follows:

$$l_i(k) = \frac{l^{\Gamma}_{\min}(k)}{l^{\Gamma}_i(k)} l_i(k-1)$$

Once sub-model's weight $p_i(k)$ has converged to 0, its weight metric $l_i(k)$ is also converged to 0. It is not difficult to find that no matter how error performance metrics changes, this sub-model's weights could not become 1. In other word, weighting algorithm could not adapt to the change of actual system parameters.

In order to solve this problem, this paper comes up with an improved weighting algorithm that could make weights recalculate to adapt to system parameter changes.

Compare the output error of each sub-model at every sample instance. Mark the sub-model's number i which output error $e_i(k)$ is the smallest. When system parameter has changed, Remark the sub-model which output error is the smallest as $M_{e\min}(k)$. If the sub-model marked at current sample instance is different from the one marked at previous sample instance:

$M_{e\min}(k) \neq M_{e\min}(k-1)$ and

$$M_{e\min}(k-1) = M_{e\min}(k-2) = M_{e\min}(k-3)$$

Initialize the weight of each sub-model. In this way, we can recalculate each sub-model's weight and reselect the best sub-model's controller to control the actual system when system parameter is changed.

3. Partition the Model-set Using v-Gap Metric

3.1 v-Gap metric basic concepts and definitions

In order to measure the distance between two nominal plants from a closed-loop perspective, we need a measure to quantify the closeness of closed-loop behavior of two open-loop plants. For this reason, the v-Gap metric introduced by Vinnicombe in is used throughout this paper [6].

The v-Gap metric is defined as:

$$\delta_v(P_1,P_2) = \begin{cases} \left\| \dfrac{\left|P_1(e^{jw}) - P_2(e^{jw})\right|}{\sqrt{1+\left|P_1(e^{jw})\right|^2}\sqrt{1+\left|P_2(e^{jw})\right|^2}} \right\|_\infty, \\ 1 \end{cases}$$

subject to

$\det(1+P_1^* P_2)(e^{jw}) \neq 0, \forall w$ And

$$wno(1+P_1^* P_2) + \eta(P_2) - \tilde{\eta}(P_1) = 0$$

Where, wno denotes the winding number evaluated on the standard Nyquist contour indented around any imaginary axis poles of P_1 and P_2.

When $\delta_v(P_1,P_2)$ is small enough, then we can consider that the controller stabilizing system P_1 could also make system P_2 stable.

3.2 Partitioning the parameter set

As v-Gap can be used to measure the distance of two systems, we use v-Gap to partition the model-set.

For the system with only one uncertain parameter, we can choose a certain value r as v-Gap metrics ($0<r<1$), in order to find sub-models' parameter and partition the model-set into N sub-models.

If actual system's parameter as follows:

$$A = \begin{bmatrix} 0 & 1 \\ \lambda & -4 \end{bmatrix}, B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix},$$

$\lambda \in (-103, -50]$

System transfer function is calculated as $G = C(sI - A)^{-1}B$, Assume two sub-models:

$$G_a(s) = \begin{bmatrix} G_{a11}(s) & G_{a12}(s) \\ G_{a21}(s) & G_{a22}(s) \end{bmatrix}$$

$$G_b(s) = \begin{bmatrix} G_{b11}(s) & G_{b12}(s) \\ G_{b21}(s) & G_{b22}(s) \end{bmatrix}$$

Calculate the v-Gap metric between those transfer functions in the same position. In this way, we can get a v-Gap matrix of two sub-models. Find the minimum parameter of this v-Gap matrix.

$$d_v(G_a, G_b) = \max\{d_v(G_{a11}, G_{b11}) \quad d_v(G_{a12}, G_{b12}) \quad d_v(G_{a21}, G_{b21}) \quad d_v(G_{a22}, G_{b22})\}$$

Choose a certain v-Gap value r=0.2, partition the model-set model parameter based on r.

Table 1 Sub-model parameters

Parameter 1	Parameter 2	v-GAP
-50	-60	0.2083
-60	-72	0.2096
-72	-86	0.2054
-86	-103	0.2094

From table 1, we can easily found that though v-Gaps between each two sub-models are the same, sub-model parameter selection are not uniform.

4. Simulation result

Assume the actual system's parameter mentioned in chapter 3 varied from -50 to -72. Partitioning the parameter set into three sub-model by a certain v-Gap r=0.2. The three sub-models' parameters are as follows:

$$A_1 = \begin{bmatrix} 0 & 1 \\ -50 & -4 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1 \\ -60 & -4 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 & 1 \\ -72 & -4 \end{bmatrix}$$

If system is a slowly time-varying system, and its parameter varies according to Table 2.

Table 2 System parameters

Sample Time	System Parameter
0-20s	-50
20-40s	-60
40-60s	-72
60-80s	-50
80-100s	-71

Record each sub-model output error data when system parameter changes from -50 to -60 at the time of 20s, as Table 3.

Table 3 Sub-models output error at the time of 20s

Sampling Point	$e_1(k)$	$e_2(k)$	$e_3(k)$
19.9s	0	1.19	2.18
20s	5.20	5.27	5.52
⋮	⋮	⋮	⋮
21.2s	1.28	0.37	1.02
21.3s	1.45	0.13	1.7

From table 3, when system parameter changed at the time of 20s, weighting algorithm determined the change of system parameter and initialized each sub-model's weight, and then recalculate each sub-model's weight. In this way, weighting algorithm can reselect the sub-model which is nearest to the actual system.

Through matlab simulation, we can clearly see that the effectiveness of weighted multiple model adaptive control of slowly time-varying systems. Simulation results are shown in Fig. 5 and Fig. 6, respectively

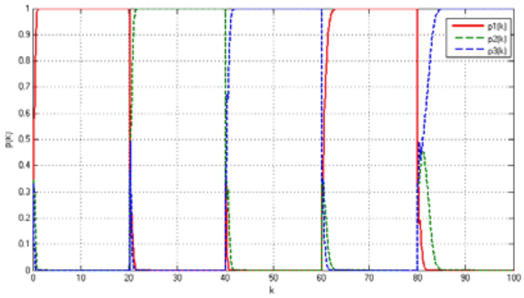

Fig. 5 Controller weight curves

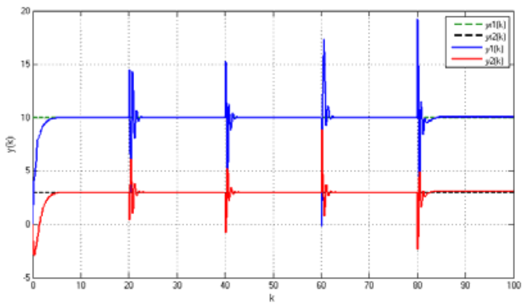

Fig. 6 Slowly time-varying system

From Fig. 5 and Fig. 6, we can easily see that at the time of 20s、40s、60s、80s when system parameter changed, sub-models weights are initialized and recalculated, and finally the nearest sub-model weight is converged to 1. Actual system output finally become stable.

From 80s to 100s, as actual system's parameter is not equal to one sub-model of model-set, steady-state error is existed between system stable output and reference input. What' s more, the weight converged much more slowly than usual. However, weighting algorithm still finally identified the nearest sub-model from model-set, and uses its controller to control actual system.

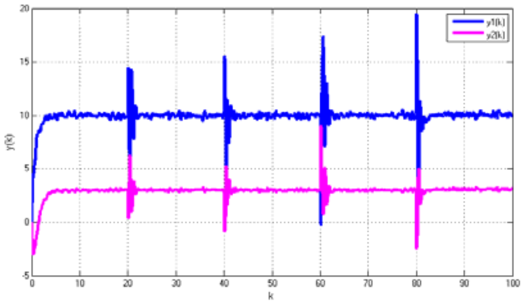

Fig. 7 Slowly time-varying system

From Fig. 7, we can see that even system is disturbed by white-noise signal, WMMAC using weighting algorithm presented in this paper can still control system well. No matter how system parameter changed, weighting algorithm can accurately choose the sub-model which is the nearest to the actual system from model-set (parameter set), and finally stabilized system output.

6. Conclusion

Weighted multiple model adaptive control is a useful control method to deal with large uncertainty of system parameters.

V-Gap metric is an important metric of two systems distance, through simulation we can draw a conclusion, though v-Gaps between each two sub-model are the same, sub-model parameter selection are not uniform. v-Gap can be used as the basis of the division of model-set.

To the slowly time-varying system, an improved weighting algorithm mentioned in this paper could make weights recalculate to adapt to system parameter changes. Through simulation, when system parameter changes, weighting algorithm can quickly realized this change and initialized each weight. If the real model of system to be contrlled is included in the model set, WMMAC can quickly find the right model and the corresponding controller.

Acknowledgements

This work is partially supported by National Basic Research Program of China (973 Program) (No. 2012CB821200), National High Technology Research and Development Program of China (863 Program) (No. 2011AA060408), Beijing Higher Education Young Elite Teacher Project (No. YETP0362), National Key Technology Research and Development Program of China (No. 2013BAB02B07).

References

1. Magill D.T. Optimal adaptive estimation of sampled stochastic processes. IEEE Transactions on Automatic Control. 1965, 10:434-439
2. Lainiotis, D.G. Optimal adaptive estimation structure and parameter adaption, IEEE Transactions on Automatic Control,1972,16:160-170
3. M. Athans, S. Fekri, and A. Pascoal, Issues on robust adaptive feedback control, in Preprints 16th IFAC World Congress, pp. 9-39, Prague, Czech Republic, July 2005
4. Sajjad Fekri, Michael Athans, and Antonio Pascoal, Issues, progress and new results in robust adaptive control, International Journal of Adaptive Control and Signal Processing, Vol. 20(10):519-579, 2006
5. Zhang Weicun. Stable Weighted Multiple Model Adaptive Control [J]. International Journal of Adaptive Control and Signal Processing, Vol. 27(7):562-581, 2013
6. Hessam Mahdianfar, Sadjaad Ozgoli, Hamid Reza Momeni. Robust multiple model adaptive control: Modified using v-GAP metric. International Journal of Robust and Nonlinear Control, 2011, 21:2027–2063