International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Synthesis of Target Angle Tracking System Combining Kalman Filter Algorithm with Fuzzy Logic Controller

Nguyen Van Bang
Missile Faculty, Air Defence - Air Force Academy, Ha Noi, Viet Nam-12013

Abstract— *Based on the tracking multi-loop target angle coordinate system of self-guided missiles. The article proposes a method to synthesize the target angle tracking system combining the Kalman filter algorithm with a fuzzy logic controller, to improve the quality of the target phase coordinate filter. The tracking system is capable of adapting to the varied maneuvering present in the reality of the target as the evaluation process progresses the most suitable model. As a result, the evaluation quality of the target phase coordinates is advanced, the tracking error is significantly reduced. Because the tracking system does not use directly the signal balance direction as the target coordinate evaluation signal, but uses a separate line of sight coordinate evaluation filter. At the same time, the related states are evaluated by the Kalman filter algorithm in the tracking loop. The structure of the filter is simple, the evaluation error is small. The results are verified through the simulation, ensure in all cases the target changes maneuvering type, including the uncertainty related to the maneuvering moment with different maneuver intensity and maneuver frequency, the line of sight angle coordinate filter always accurately determines the target angle coordinates.*

Keywords— *Target, Maneuvering, The line of sight angle, Kalman filter, Fuzzy logic controller.*

I. INTRODUCTION

In radar stations and self-guided heads, the determination of target coordinates is done by tracking systems [3]. Including, angle tracking channels and distance tracking channels. The maneuverability of the target causes the appearance of higher-order derivative components of the coordinates to be tracked (input of the tracking systems) [1], [2]. As a result, the tracking error will increase and may lead to loss of tracking. Especially the angle coordinate tracking channels. Because, in angle tracking channels, it is required to control the antenna towards the target through a drive system with high inertia [8].

In the case of a maneuvering target, the tracking systems used to determine the target angle coordinates used on current self-guided systems have low accuracy. Because the tracking system uses directly the information from the antenna drive system as a signal to evaluate the target coordinates [3], [7].

Applying the linear Kalman filter technique [5], [6], [10], the maneuvering frequency and maneuvering intensity parameters are only selected by constants. Therefore, these parameters are not representative of every maneuvering motion of the target. The tracking error will increase when the actual movement of the target does not suitable for the hypothetical model.

Adaptive filtering algorithm with maneuvering detection based on amplification factor correction according to tracking error signal [6], the amplification factor increase when the target is maneuvering will increase the influence of observe noise on the result of evaluation. Therefore, the evaluation accuracy is not high, especially for the line of sight angular speed component. Although simple algorithms but need improvements when used [2].

Adaptive filtering algorithm with maneuvering detection based on state model parameter recognition [1]. Because the forecast matrix is continuously recognized, the state forecast will be more accurate. Because the algorithm does not increase the amplification factor, it does not increase the influence of observations noise to the state evaluation. Therefore, the state evaluation accuracy of the algorithm is advanced. However, because the state transition matrix adaptation does not use prior information about the different maneuverability of the target, the recognition accuracy for different maneuvering cases is different [2].

With the application of fuzzy logic controller, the article proposes algorithm to perfect the state forecasts matrix recognition plan to respond to more diverse layer of maneuvering models than the two above options. In this algorithm, the target kinematics are hypothesized to follow N models. The fuzzy system determines the reasonableness level of each model to correct the state forecasts matrix, in order to increase the factor of the most reasonable model to the recognition model.

Therefore, the article is set out to build a target angle tracking system with high accuracy in the conditions of maneuvering and super maneuvering targets.

II. ALGORITHM TO DETERMINE TARGET ANGLE COORDINATES COMBINING KALMAN FILTER WITH FUZZY LOGIC CONTROLLER

Through the recognition of the state transition matrix Φ, the line-of-sight angle coordinate evaluation filter is synthesized on the basis of the state model parameter recognition to perform the adaptation on the forecast phase and on the correction phase of the Kalman filter algorithm [2]. The essence of this method is to find the matrix Φ such that the "state forecast at time k" coincides with the "state evaluation at time k" when using the model recognized at the previous time point. The result of recognition Φ will be used for the next time step [2].

However, the change of matrix Φ depends on the imposition of the error correlation matrix D_a and the intensity matrix Q_a. In other words, the amplification matrix K_a in the

Nguyen Van Bang, "Synthesis of Target Angle Tracking System Combining Kalman Filter Algorithm with Fuzzy Logic Controller," *International Research Journal of Advanced Engineering and Science*, Volume 6, Issue 3, pp. 302-310, 2021.

matrix recognition algorithm Φ does not take into account the different changes of the maneuvering frequency and maneuvering intensity. When the target is maneuvering with high frequency and intensity, Q_a needs to choose large, and when the target is not maneuvering, Q_a needs to choose very small, in order to reduce the evaluation error of Φ.

The determine domain of Φ is not known in advance. Φ is continuously adapted to fit the maneuvering motion of the target [2]. This adaptation process only uses a priori information D_a, Q_a and does not use prior information about the frequency of the target's maneuvers.

The article proposes an algorithm to recognize matrix Φ to overcome the above mentioned disadvantages. The essence of the algorithm is the matrix Φ is recognized on the basis of N matrix Φ^n $(n=1,2,...,N)$ (Φ^n are known in advance and are created from two layeres of target models); where, each Φ^n is suitable for a degree of maneuverability. Determining the influence level of each model (Φ^n) on the result of recognition Φ is performed by a fuzzy logic controller.

2.1. Structure of the line-of-sight angle coordinate evaluation filter using Kalman filter algorithm combined with fuzzy logic controller

The change of line of sight angle coordinate φ_m and angular speed ω_m are determined by the initial state space model, which has the form [1], [2], [3]:

$$\varphi_m(k)=\varphi_m(k-1)+\tau\omega_m(k-1),\ \varphi_m(0)=\varphi_{m0};$$
$$\omega_m(k)=(1-\tau\alpha_m)\omega_m(k-1)+\xi_{\omega_m}(k-1),\ \omega_m(0)=\omega_{m0}; \quad (1)$$

k - Discrete time;

τ - Discrete time step;

α_m - Characteristic coefficient for the maneuverability of the target;

ξ_{ω_m} - Central Gause white noise with known variance.

When taking into account the maneuverability of the target, model (1) is extended into two model layeres.

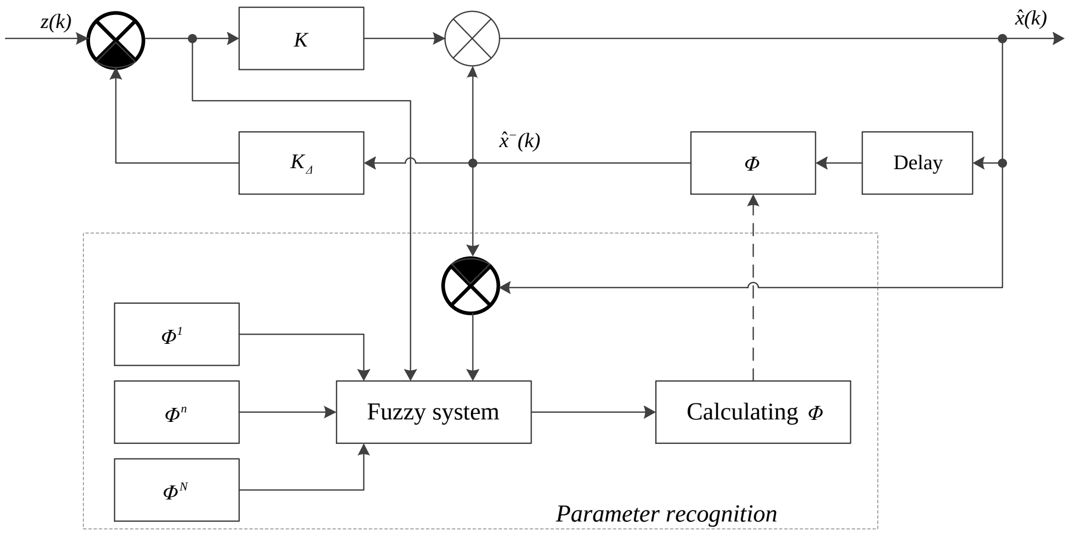

Fig. 1. Structure diagram of adaptive filter using Kalman algorithm combined with fuzzy logic controller.

The first layer consists of N_1 models with size 2. The models differ only by the parameter α_m^n. The model has the form:

$$\varphi_m(k)=\varphi_m(k-1)+\tau\omega_m(k-1)$$
$$\omega_m(k)=(1-\tau\alpha_m^n)\omega_m(k-1)+\xi_{\omega_m}(k-1) \quad (2)$$

Where, the state transition matrix Φ^n:

$$\Phi^n=\begin{bmatrix}1 & \tau\\ 0 & 1-\tau\alpha_m^n\end{bmatrix};\ n=1,2,...,N_1 \quad (3)$$

The second layer consists of N_2 models with size 3, $(N_2=N-N_1)$, the state space size is expanded compared to model (1). The models differ only by the parameter α_m^n. The model has the form:

$$\varphi_m(k)=\varphi_m(k-1)+\tau\omega_m(k-1)+\tfrac{1}{2}\tau^2\dot{\omega}_m(k-1)$$
$$\omega_m(k)=\omega_m(k-1)+\tau\dot{\omega}_m(k-1) \quad (4)$$
$$\dot{\omega}_m(k)=(1-\tau\alpha_m^n)\dot{\omega}_m(k-1)+\xi_{\omega_m}(k-1)$$

The state transition matrix Φ^n:

$$\Phi^n=\begin{bmatrix}1 & \tau & \tfrac{1}{2}\tau^2\\ 0 & 1 & \tau\\ 0 & 0 & 1-\tau\alpha_m^n\end{bmatrix};\ n=N_1+1,...,N \quad (5)$$

In (3), (5) parameters α_m^n, $(n=1,2,...,N)$ are selected in range $\tau\alpha_m^n=0\div 1$.

2.2. Algorithm to evaluate the line of sight angle coordinates

Suppose, the state transition matrix Φ is used to evaluate the line of sight angle coordinates of the form:

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} \\ \Phi_{31} & \Phi_{32} & \Phi_{33} \end{bmatrix} \quad (6)$$

The line-of-sight angle coordinate evaluation filter is performed on two "blocks" (figure 1):

The block evaluates the phase coordinates of the line of sight when the matrix Φ is known;

The matrix recognition block Φ, when evaluating the known phase coordinates and according to the parameters of the hypothetical model α_m^n.

- To recognition Φ, suppose at the time $(k-1)$ evaluates the state corresponding to the recognized model as $\hat{\varphi}_m(k-1)$, $\hat{\omega}_m(k-1)$ and $\hat{\dot{\omega}}(k-1)$ then:

+ If using the first model layer, the state forecast is determined by:

$$\begin{bmatrix} \hat{\varphi}_m^{n-}(k) \\ \hat{\omega}_m^{n-}(k) \end{bmatrix} = \Phi^n \begin{bmatrix} \hat{\varphi}_m(k-1) \\ \hat{\omega}_m(k-1) \end{bmatrix}; \; n = 1,2,...,N_1 \quad (7)$$

Φ^n is determined according to (3).

+ If using the second model layer, the state forecast would be:

$$\begin{bmatrix} \hat{\varphi}_m^{n-}(k) \\ \hat{\omega}_m^{n-}(k) \\ \hat{\dot{\omega}}_m^{n-}(k) \end{bmatrix} = \Phi^n \begin{bmatrix} \hat{\varphi}_m(k-1) \\ \hat{\omega}_m(k-1) \\ \hat{\dot{\omega}}_m(k-1) \end{bmatrix}; \; n = N_1+1, N_1+2,...,N \quad (8)$$

Φ^n is determined according to (5).

+ According to the recognized model, the state forecast will be:

$$\begin{bmatrix} \hat{\varphi}_m^{-}(k) \\ \hat{\omega}_m^{-}(k) \\ \hat{\dot{\omega}}_m^{-}(k) \end{bmatrix} = \Phi \begin{bmatrix} \hat{\varphi}_m(k-1) \\ \hat{\omega}_m(k-1) \\ \hat{\dot{\omega}}_m(k-1) \end{bmatrix} \quad (9)$$

Φ has the form (6).

The matrix Φ is determined such that the state forecast produced by it is equal to the weighted sum of the state forecasts produced by the hypothetical models, i.e.:

$$\begin{bmatrix} \hat{\varphi}_m^{-}(k) \\ \hat{\omega}_m^{-}(k) \\ \hat{\dot{\omega}}_m^{-}(k) \end{bmatrix} = \sum_{n=1}^{N} \beta^n \begin{bmatrix} \hat{\varphi}_m^{n-}(k) \\ \hat{\omega}_m^{n-}(k) \\ \hat{\dot{\omega}}_m^{n-}(k) \end{bmatrix} \quad (10)$$

The weight β^n determines suitable level (probability) of the n^{th} model with the actual motion of the target, so we can choose:

$$\sum_{n=1}^{N} \beta^n = 1 \quad (11)$$

Replace (3), (5), (8), (9) in (10), get:

$$\begin{bmatrix} \hat{\varphi}_m^{-}(k) \\ \hat{\omega}_m^{-}(k) \\ \hat{\dot{\omega}}_m^{-}(k) \end{bmatrix} = \left(\sum_{n=1}^{N_1} \beta^n \begin{bmatrix} 1 & \tau & 0 \\ 0 & 1-\tau\alpha_m^n & 0 \\ 0 & 0 & 0 \end{bmatrix} + \sum_{n=N_1+1}^{N} \beta^n \begin{bmatrix} 1 & \tau & \frac{1}{2}\tau^2 \\ 0 & 1 & \tau \\ 0 & 0 & 1-\tau\alpha_m^n \end{bmatrix} \right) \begin{bmatrix} \hat{\varphi}_m(k-1) \\ \hat{\omega}_m(k-1) \\ \hat{\dot{\omega}}_m(k-1) \end{bmatrix}$$

$$\begin{bmatrix} \hat{\varphi}_m^{-}(k) \\ \hat{\omega}_m^{-}(k) \\ \hat{\dot{\omega}}_m^{-}(k) \end{bmatrix} = \begin{bmatrix} 1 & \tau & \frac{1}{2}\tau^2 \sum_{n=N_1+1}^{N} \beta^n \\ 0 & 1-\sum_{n=1}^{N_1} \beta^n \tau\alpha_m^n & \tau \sum_{n=N_1+1}^{N} \beta^n \\ 0 & 0 & \sum_{n=N_1+1}^{N} \beta^n (1-\tau\alpha_m^n) \end{bmatrix} \begin{bmatrix} \hat{\varphi}_m(k-1) \\ \hat{\omega}_m(k-1) \\ \hat{\dot{\omega}}_m(k-1) \end{bmatrix} \quad (12)$$

From (9) and (12), get:

$$\Phi = \begin{bmatrix} 1 & \tau & \frac{1}{2}\tau^2 \sum_{n=N_1+1}^{N} \beta^n \\ 0 & 1-\tau\sum_{n=1}^{N_1} \beta^n \alpha_m^n & \tau \sum_{n=N_1+1}^{N} \beta^n \\ 0 & 0 & \sum_{n=N_1+1}^{N} \beta^n (1-\tau\alpha_m^n) \end{bmatrix} \quad (13)$$

In (13), the matrix Φ depends on the known parameters α_m^n $(n = 1,2,...,N)$ of the hypothetical models and depends on the unknown parameters $\beta^n(k)$ $(n = 1,2,...,N)$.

The description model the change of $\beta^n(k)$ is selected as follows:

$$\beta^n(k) = \frac{\beta^n(k-1) + \Delta\beta^n}{\sum_{i=1}^{N} \left(\beta^i(k-1) + \Delta\beta^i \right)} \quad (14)$$

The amount of correction $\Delta\beta^n$ is chosen to be $\Delta\beta^n = K_\gamma \times \gamma^n$, for $K_\gamma = const$, then:

$$\beta^n(k) = \frac{\beta^n(k-1) + K_\gamma \gamma^n}{1 + K_\gamma} \quad (15)$$

Models (14), (15) show that, the fit level of the n^{th} model at time k is determined from the fit level at time $(k-1)$ (is $\beta^n(k-1)$), by adding an amount of correction $\Delta\beta^n$. The correction amount $\Delta\beta^n$ is determined from parameter γ^n, with the amplification coefficient K_γ by a constant; where, γ^n needs to be adapted from the observation information. The denominator component of (14) is included for the purpose of normalizing $\beta^n(k)$, ensuring $\sum_{n=1}^{N} \beta^n = 1$.

The parameters γ^n, $(n = 1,2,...,N)$, needs to be determined from the observation information. The value of γ^n is large when the n^{th} model is considered fit and opposite. To build an algorithm to determine the vector γ $(\gamma = [\gamma^1, \gamma^2, ..., \gamma^n, ...\gamma^N]^T)$, need to consider the errors from the hypothetical models.

- Evaluate phase coordinates of line of sight when matrix Φ is known:

+ Tracking error in hypothetical models

Suppose, at time $(k-1)$ the received status evaluations are $\hat{\varphi}_m(k-1)$, $\hat{\omega}_m(k-1)$ and $\hat{\dot{\omega}}(k-1)$.

When using the first model layer, the tracking error at time k will be:

$$\Delta z^n(k) = z(k) - K_\Delta\left(\hat{\varphi}_m(k-1) + \tau\hat{\omega}_m(k-1)\right);\ n = 1,2,...,N_1 \quad (16)$$

When using the second model layer, the tracking error at time k will be:

$$\Delta z^n(k) = z(k) - K_\Delta\left(\hat{\varphi}_m(k-1) + \tau\hat{\omega}_m(k-1) + \tfrac{1}{2}\tau^2\hat{\dot{\omega}}_m(k-1)\right) \quad (17)$$
$$n = N_1+1, N_1+2,...,N$$

When using recognized model Φ, the tracking error at time k will be:

$$\Delta z(k) = z(k) - K_\Delta\left(\hat{\varphi}_m(k-1) + \tau\hat{\omega}_m(k-1) + \tfrac{1}{2}\tau^2\sum_{l=N_1+1}^{N}\beta^l\hat{\dot{\omega}}_m(k-1)\right) \quad (18)$$

The tracking error of the first model layer can be determined through the tracking error of the recognized model:

$$\Delta z^n(k) = \Delta z(k) + \tfrac{1}{2}\tau^2 K_\Delta\sum_{l=N_1+1}^{N}\beta^l\hat{\dot{\omega}}_m(k-1) \quad (19)$$

Or can be rewritten as:

$$\Delta z^n(k) = \Delta z(k) + \tfrac{1}{2}\tau^2 K_\Delta\left(1 - \sum_{l=1}^{N_1}\beta^l\right)\hat{\dot{\omega}}_m(k-1);\ n = 1,2,...,N_1 \quad (20)$$

Similarly, the tracking error for the second model layer is:

$$\Delta z^n(k) = \Delta z(k) + \tfrac{1}{2}\tau^2 K_\Delta\left(\sum_{l=N_1+1}^{N}\beta^l - 1\right)\hat{\dot{\omega}}_m(k-1);\ n = N_1+1, N_1+2,...,N \quad (21)$$

Or can be rewritten as:

$$\Delta z^n(k) = \Delta z(k) - \tfrac{1}{2}\tau^2 K_\Delta\sum_{l=1}^{N_1}\beta^l\hat{\dot{\omega}}_m(k-1);\ n = N_1+1, N_1+2,...,N \quad (22)$$

The tracking error of the hypothetical filters (of the layer of filters) is determined from the tracking error of the filter using the recognition model.

+ Error of state evaluation from state forecast.

Error between the status evaluated at the current time k and status forecast when using the n^{th} hypothetical model is:

$$\begin{bmatrix}\Delta\varphi^n(k)\\ \Delta\omega^n(k)\\ \Delta\dot{\omega}^n(k)\end{bmatrix} = \begin{bmatrix}\hat{\varphi}_m(k)\\ \hat{\omega}_m(k)\\ \hat{\dot{\omega}}_m(k)\end{bmatrix} - \Phi^n\begin{bmatrix}\hat{\varphi}_m(k-1)\\ \hat{\omega}_m(k-1)\\ \hat{\dot{\omega}}_m(k-1)\end{bmatrix} \quad (23)$$

Using the first model layer, we have:

$$\Delta\varphi^n(k) = \hat{\varphi}_m(k) - \left(\hat{\varphi}_m(k-1) + \tau\hat{\omega}_m(k-1)\right)$$
$$\Delta\omega^n(k) = \hat{\omega}_m(k) - (1 - \tau\alpha_m^n)\hat{\omega}_m(k-1) \qquad with\ n = 1,2,...,N_1 \quad (24)$$
$$\Delta\dot{\omega}^n(k) = \hat{\dot{\omega}}_m(k)$$

Using the second model layer, we have:

$$\Delta\varphi^n(k) = \hat{\varphi}_m(k) - \left(\hat{\varphi}_m(k-1) + \tau\hat{\omega}_m(k-1) + \tfrac{1}{2}\tau^2\hat{\dot{\omega}}_m(k-1)\right)$$
$$\Delta\omega^n(k) = \hat{\omega}_m(k) - \left(\hat{\omega}_m(k-1) + \tau\hat{\dot{\omega}}_m(k-1)\right)\ with\ n = N_1+1, N_1+2,...,N \quad (25)$$
$$\Delta\dot{\omega}^n(k) = \hat{\dot{\omega}}_m(k) - (1 - \tau\alpha_m^n)\hat{\dot{\omega}}_m(k-1)$$

When using the recognition model, we have:

$$\Delta\varphi(k) = \hat{\varphi}_m(k) - \left(\hat{\varphi}_m(k-1) + \tau\hat{\omega}_m(k-1) + \tfrac{1}{2}\tau^2\sum_{l=N_1+1}^{N}\beta^l\hat{\dot{\omega}}_m(k-1)\right)$$
$$\Delta\omega(k) = \hat{\omega}_m(k) - \left(\left(1 - \tau\sum_{l=1}^{N_1}\beta^l\alpha_m^l\right)\hat{\omega}_m(k-1) + \tau\sum_{l=N_1+1}^{N}\beta^l\hat{\dot{\omega}}_m(k-1)\right) \quad (26)$$
$$\Delta\dot{\omega}(k) = \hat{\dot{\omega}}_m(k) - \sum_{l=N_1+1}^{N}\beta^l(1 - \tau\alpha_m^l)\hat{\dot{\omega}}_m(k-1)$$

From (24), (25), (26), get:

$$\Delta\varphi^n(k) = \Delta\varphi(k) + \tfrac{1}{2}\tau^2\sum_{l=N_1+1}^{N}\beta^l\hat{\dot{\omega}}_m(k-1)$$
$$\Delta\omega^n(k) = \Delta\omega(k) + \left(-\tau\sum_{l=1,l\neq n}^{N_1}\beta^l\alpha_m^l\hat{\omega}_m(k-1) + \tau\sum_{l=N_1+1}^{N}\beta^l\hat{\dot{\omega}}_m(k-1)\right) \quad (27)$$
$$\Delta\dot{\omega}^n(k) = \Delta\dot{\omega}(k) + \sum_{l=N_1+1}^{N}\beta^l(1 - \tau\alpha_m^l)\hat{\dot{\omega}}_m(k-1)\ with\ n = 1,2,...,N_1$$

$$\Delta\varphi^n(k) = \Delta\varphi(k) - \tfrac{1}{2}\tau^2\sum_{l=1}^{N_1}\beta^l\hat{\dot{\omega}}_m(k-1)\ with\ n = N_1+1, N_1+2,...,N$$
$$\Delta\omega^n(k) = \Delta\omega(k) + \left(-\tau\sum_{l=1}^{N_1}\beta^l\alpha_m^l\hat{\omega}_m(k-1) - \tau\sum_{l=1}^{N_1}\beta^l\hat{\dot{\omega}}_m(k-1)\right) \quad (28)$$
$$\Delta\dot{\omega}^n(k) = \Delta\dot{\omega}(k) + \sum_{l=N_1+1,l\neq n}^{N}\beta^l(1 - \tau\alpha_m^l)\hat{\dot{\omega}}_m(k-1)$$

From (27), (28), found that:

The errors when using the n^{th} hypothetical model are determined from the error of the recognition model.

The comments for $\Delta\varphi^n(k)$ are similar to those for $\Delta z^n(k)$. Therefore, to determine γ either use $\Delta z^n(k)$ or use $\Delta\varphi^n(k)$.

2.3. Evaluation algorithm the line of sight angle coordinates using fuzzy logic controller

To determine γ^n, $(n = 1,2,...,N)$, we use a fuzzy logic controller with errors $\Delta\varphi(k)$, $\Delta\omega(k)$ and $\Delta\dot{\omega}(k)$ as input. This fuzzy system ensures to increase the efficiency of the most reasonable model to the recognition model. That is, if the n^{th} model is considered reasonable, then γ^n takes a large value, and γ^i $(i \neq n)$ takes a small value.

A fuzzy logic controller, the basic components include: fuzzying stage, composition equipment and defuzzification (fuzzy solving) stage [9].

- Fuzzying stage:

The input variable of the fuzzy system includes: $\Delta\varphi(k)$,

Nguyen Van Bang, “Synthesis of Target Angle Tracking System Combining Kalman Filter Algorithm with Fuzzy Logic Controller,” *International Research Journal of Advanced Engineering and Science*, Volume 6, Issue 3, pp. 302-310, 2021.

$\Delta\omega(k)$ and $\Delta\dot{\omega}(k)$.

+ For the input variable $\Delta\varphi(k)$, is fuzzed by two language variables with a selection interdependencies function of the form Gause:

$$\mu_{1i} = exp\left(-\frac{1}{2}\left(\frac{\Delta\varphi(k) - c_{1i}}{\sigma_{1i}}\right)^2\right); i = 1,2 \quad (29)$$

The first language variable L_1^1 determines the reasonableness of the first model layer according to the observation information. From (27), the center (midpoint) of the interdependencies function of L_1^1 is determined by:

$$c_{11} = -\tfrac{1}{2}t^2 \sum_{n=N_1+1}^{N} b^n(k-1)\hat{w}_m(k-1) \quad (30)$$

With this choice, the expression $(\Delta\varphi(k) - c_{1i})$ in (29) is the error (deviation) of the state evaluation from the state forecast when using the i^{th} model layer. This value is smaller, then μ_{1i} is larger. Thus, μ_{1i} determines the reasonableness of the i^{th} model layer according to information $\Delta\varphi(k)$.

Similarly, the second language variable L_2^1 determines the reasonableness of the second model layer according to the observation information. From (28) the center of the interdependencies function of L_2^1 is determined by:

$$c_{12} = \tfrac{1}{2}\tau^2 \sum_{n=1}^{N_1} \beta^n \hat{\omega}_m(k-1) \quad (31)$$

The widths of these interdependencies function are chosen equally $\sigma_{11} = \sigma_{12} = \sigma_1$, so that the interdependencies function μ_{11} and μ_{12} intersect. Choose

$$\sigma_1 = k_{\sigma_1}\left|c_{11} - c_{12}\right| \quad (32)$$

+ For the input variable $\Delta\omega(k)$, is fuzzed by (N_1+1) language variables. Because, in expression (27), $\Delta\omega^n(k)$ distinguishes N_1 models belonging to the first layer. In expression (28), $\Delta\omega^n(k)$ can only distinguish models belonging to the second layer. The selected interdependencies function has the form Gause:

$$\mu_{2i} = exp\left(-\frac{1}{2}\left(\frac{\Delta\omega(k) - c_{2i}}{\sigma_{2i}}\right)^2\right); i = 1,2,...,N_1+1 \quad (33)$$

Language variables L_n^2 $(n = 1,2,...,N_1)$ show the reasonableness of n^{th} model in the first model layer. From (27), the center of the interdependencies function of L_n^2 is determined by:

$$c_{2n} = \tau\left(\sum_{l=1}^{N_1}\beta^l\alpha_m^l - \beta^n\alpha_m^n\right)\hat{\omega}_m(k-1) - \tau\sum_{l=N_1+1}^{N}\beta^l\hat{\omega}_m(k-1); n = 1,2,...,N_1 \quad (34)$$

Language variables $L_{N_1+1}^2$ show the reasonableness of models in the second model layer. From (28), the center of the interdependencies function of $L_{N_1+1}^2$ is determined by:

$$c_{2n} = \tau\sum_{l=1}^{N_1}\beta^l\alpha_m^l\hat{\omega}_m(k-1) + \tau\sum_{l=1}^{N_1}\beta^l\hat{\omega}_m(k-1); n = N_1+1 \quad (35)$$

The interdependencies function widths of the language variables L_n^2, $(n = 1,2,...,N_1+1)$, are chosen so that consecutive interdependencies function intersect. For simplicity, can choose $\sigma_{2n} = \sigma_2$ with every $n = 1,2,...,N_1+1$.

$$\sigma_2 = k_{\sigma_2}\left|max(c_{2n}) - min(c_{2n})\right| / N_1 \quad (36)$$

+ For the input variable $\Delta\dot{\omega}(k)$, is fuzzed by (N_2+1) language variables, $(N_2 = N - N_1)$. Because, in expression (27), $\Delta\dot{\omega}^n(k)$ can only distinguish models belonging to the first layer. In expression (28), $\Delta\dot{\omega}^n(k)$ distinguishes N_2 models belonging to the second layer. The selected interdependencies function has the form Gause:

$$\mu_{3i} = exp\left(-\frac{1}{2}\left(\frac{\Delta\dot{\omega}(k) - c_{3i}}{\sigma_{3i}}\right)^2\right); i = 1,2,...,N_2+1 \quad (37)$$

Language variables L_1^3 show the reasonableness of models in the first model layer. From (27), the center of the interdependencies function of L_1^3 is determined by:

$$c_{3n} = \sum_{l=N_1+1}^{N}\beta^l(1 - \tau\alpha_m^l)\hat{\omega}_m(k-1); n = 1 \quad (38)$$

Language variables L_n^3, $(n = 2,...,N_2+1)$, show the reasonableness of $(N_1 - 1 + n)^{th}$ model in the second model layer. From (28), the center of the interdependencies function of L_n^3 is determined by:

$$c_{3n} = \left(\sum_{l=N_1+1}^{N}\beta^l(1 - \tau\alpha_m^l) - \beta^{N_1-1+n}(1 - \tau\alpha_m^{N_1-1+n})\right)\hat{\omega}_m(k-1) \quad (39)$$

The interdependencies function widths of the language variables L_n^3, $(n = 1,2,...,N_2+1)$, are chosen so that consecutive interdependencies function intersect. For simplicity, can choose $\sigma_{3n} = \sigma_3$ with every $n = 1,2,...,N_2+1$.

$$\sigma_3 = k_{\sigma_3}\left|max(c_{3n}) - min(c_{3n})\right| / N_2 \quad (40)$$

+ Fuzzying of the output variable γ:

Interdependencies function of the output variable selected form "singleton". That is, the output variable is fuzzed by N fuzzy set corresponding to N language variables. The value of the interdependencies function of each fuzzy set is equal to 1 only at the specific value of the output variable, at the remaining values of the output variable the interdependencies function is zero.

This method is extended to the vector form output variable. That is, γ is fuzzed by N fuzzy set, denoted by A_γ^n is the n^{th} fuzzy set with interdependencies function;

Nguyen Van Bang, "Synthesis of Target Angle Tracking System Combining Kalman Filter Algorithm with Fuzzy Logic Controller," *International Research Journal of Advanced Engineering and Science*, Volume 6, Issue 3, pp. 302-310, 2021.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

$$\mu_{A_\gamma^n}(\gamma) = \begin{cases} 1 & when\ \gamma = \gamma_n \\ 0 & when\ \gamma \neq \gamma_n \end{cases} \quad (41)$$

γ_n - vector of size N.

- Composition equipment:

Composition equipment are built through the selection of fuzzy composition law. Selecting N composition law, the n^{th} composition law determines the increasing influence of model n^{th} on the recognition model. The n^{th} composition law has the form:

Law n: If $\Delta\varphi(k)$ is A_1^n and $\Delta\omega(k)$ is A_2^n and $\Delta\dot{\omega}(k)$ is A_3^n then γ is A_γ^n.

The fuzzy set A_γ^n is determined by the interdependencies function $\mu_{A_\gamma^n}(\gamma)$. The values of the elements of vector γ_n are selected so that the n^{th} element has the largest value, the remaining elements have a smaller value. For simplicity can choose equal.

$$\gamma_n = \begin{bmatrix} \gamma_n^1 & \dots & \gamma_n^i & \dots & \gamma_n^N \end{bmatrix}^T \quad (42)$$

With, γ_n^i, ($i = 1,2,...,N$), determined as follows:

$$\gamma_n^i = \begin{cases} \gamma_{max} & when\ i = n \\ (1-\gamma_{max})/(N-1) & when\ i \neq n \end{cases} \quad (43)$$

γ_{max} - design parameter, its value is selected in range $1/N < \gamma_{max} \leq 1$.

Specifically, the fuzzy laws are selected as follows:

Law 1: If $\Delta\varphi$ is L_1^1 and $\Delta\omega$ is L_1^2 and $\Delta\dot{\omega}$ is L_1^3 then γ is $\mu_{A_\gamma^1}(\gamma)$

Law 2: If $\Delta\varphi$ is L_1^1 and $\Delta\omega$ is L_2^2 and $\Delta\dot{\omega}$ is L_1^3 then γ is $\mu_{A_\gamma^2}(\gamma)$

...

Law N_1: If $\Delta\varphi$ is L_1^1 and $\Delta\omega$ is $L_{N_1}^2$ and $\Delta\dot{\omega}$ is L_1^3 then γ is $\mu_{A_\gamma^{N_1}}(\gamma)$ (44)

Law N_1+1: If $\Delta\varphi$ is L_2^1 and $\Delta\omega$ is $L_{N_1+1}^2$ and $\Delta\dot{\omega}$ is L_2^3 then γ is $\mu_{A_\gamma^{N_1+1}}(\gamma)$

Law N_1+2: If $\Delta\varphi$ is L_2^1 and $\Delta\omega$ is $L_{N_1+1}^2$ and $\Delta\dot{\omega}$ is L_3^3 then γ is $\mu_{A_\gamma^{N_1+2}}(\gamma)$

...

Law N: If $\Delta\varphi$ is L_2^1 and $\Delta\omega$ is $L_{N_1+1}^2$ and $\Delta\dot{\omega}$ is $L_{N_2+1}^3$ then γ is $\mu_{A_\gamma^N}(\gamma)$

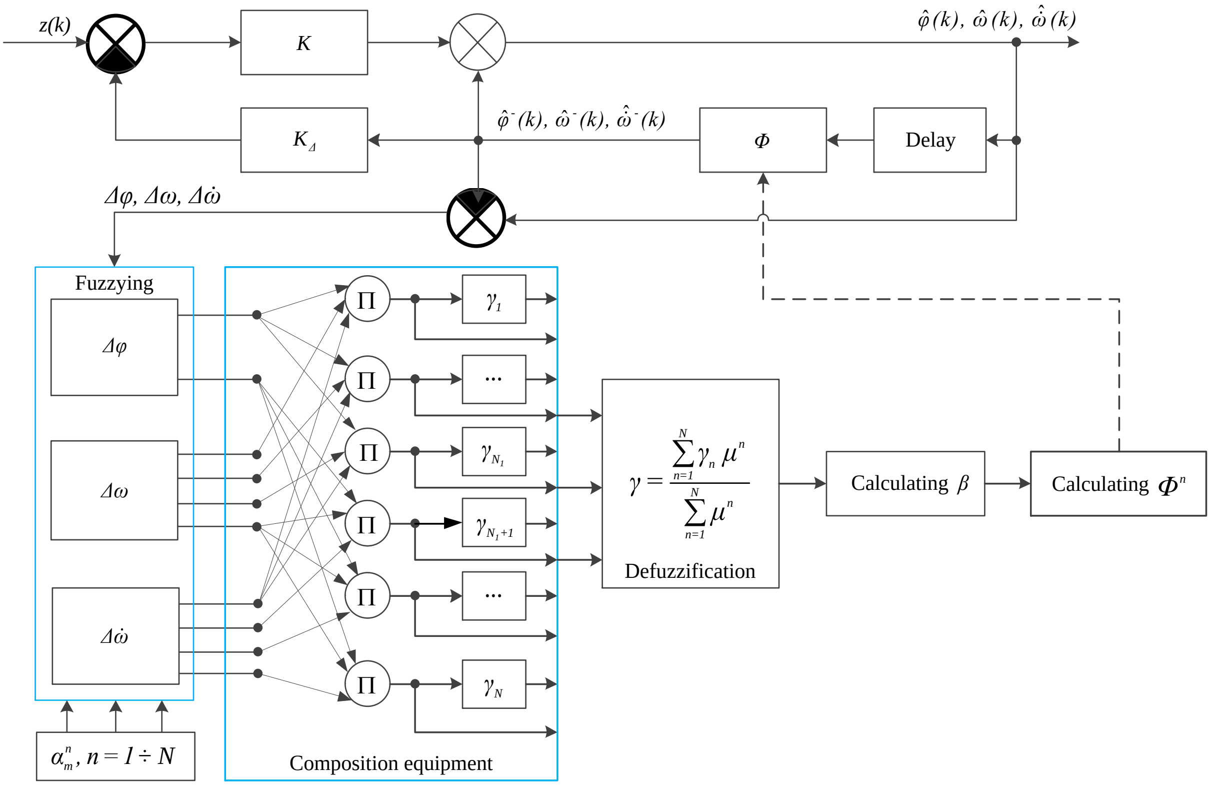

Fig. 2. Structure diagram of adaptive filter using Kalman algorithm combined with fuzzy logic controller to evaluate line of sight angle coordinates.

For each fuzzy law, apply the MAX-PROD composition rule and in the conditional clause using the algebraic product fuzzy intersection calculation, its interdependencies function is determined as follows:

Nguyen Van Bang, "Synthesis of Target Angle Tracking System Combining Kalman Filter Algorithm with Fuzzy Logic Controller," *International Research Journal of Advanced Engineering and Science*, Volume 6, Issue 3, pp. 302-310, 2021.

For $n = 1,2,...,N_1$, then:

$$\mu_{R^n} = \begin{cases} \mu_{11}(\Delta\varphi)\mu_{2n}(\Delta\omega)\mu_{31}(\Delta\dot{\omega})\mu_{A_\gamma^n}(\gamma) = \mu^n & when \quad \gamma = \gamma_n \\ 0 & when \quad \gamma \neq \gamma_n \end{cases} \quad (45)$$

Which has set $\mu^n = \mu_{11}(\Delta\varphi)\mu_{2n}(\Delta\omega)\mu_{31}(\Delta\dot{\omega}), n = 1,2,...,N_1$.

For $n = N_1+1, N_1+2,...,N$, then:

$$\mu_{R^n} = \begin{cases} \mu_{12}(\Delta\varphi)\mu_{2(N_1+1)}(\Delta\omega)\mu_{3(n-N_1+1)}(\Delta\dot{\omega})\mu_{A_\gamma^n}(\gamma) = \mu^n & when \quad \gamma = \gamma_n \\ 0 & when \quad \gamma \neq \gamma_n \end{cases} \quad (46)$$

Which has set $\mu^n = \mu_{12}(\Delta\varphi)\mu_{2(N_1+1)}(\Delta\omega)\mu_{3(n-N_1+1)}(\Delta\dot{\omega})$, $n = N_1+1, N_1+2,...,N$.

- Defuzzification:

Choosing the central defuzzification method, with the composition law as described above, the output variable γ is determined by:

$$\gamma = \frac{\sum_{n=1}^{N} \gamma_n \mu^n}{\sum_{n=1}^{N} \mu^n} \quad (47)$$

III. SIMULATION RESULTS AND ANALYSIS

- The sample trajectory is generated from the following kinetic model:

$$\begin{aligned} &\varphi_m(k) = \varphi_m(k-1) + \tau\,\omega_m(k-1) \\ &\omega_m(k) = (1-\tau\,\alpha_m)\omega_m(k-1) + \alpha_m \tau\, u \\ &z(k) = \varphi_m(k) + \xi_{z\varphi}(k) \end{aligned} \quad (48)$$

- Parameters for creating line of sight trajectory:

$\tau = 0,001(s)$, $\alpha_m = 0,6\,(1/s)$, $\varphi_{m0} = 0\,(rad)$, $\omega_{m0} = 0\,(rad/s)$, $\sigma^2_{\xi_z} = 0,05^2\,(o^2)$, simulation time $t_\Sigma = 15(s)$, . $u\ (^O/s) = \begin{cases} 0,5 & when\ t < 5\,s \\ 5 & when\ t \geq 5\,s \end{cases}$.

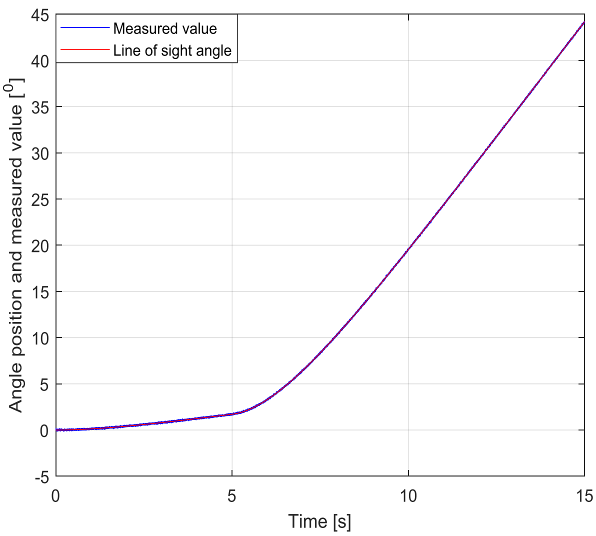

Fig. 3. Line of sight angle and measured value.

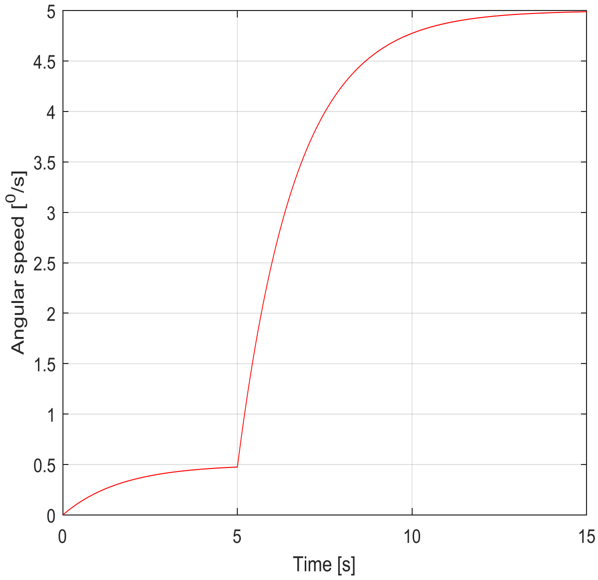

Fig. 4. Line of sight angle speed.

u - Characterizes the magnitude of the line of sight angle speed. When there is a change in the value of u, the angular speed of the line of sight will change from the current value to the new value of u;

α_m - Parameter that characterizes the speed of change of angular speed (angular acceleration); α_m greater, the larger the angular acceleration.

- Filter parameters using fuzzy adaptive system:

+ Layer 1: $N_1 = 4$; $\alpha_m = 0,001;\ 0,33;\ 0,66;\ 1$.

+ Layer 2: $N_2 = 4$; $\alpha_m = 1;\ 0,66;\ 0,33;\ 0,001$.

+ Other parameters: $\gamma_{max} = 0,85$, $\sigma_{\omega_m} = 0,0015$, $\beta^n = 0,25$.

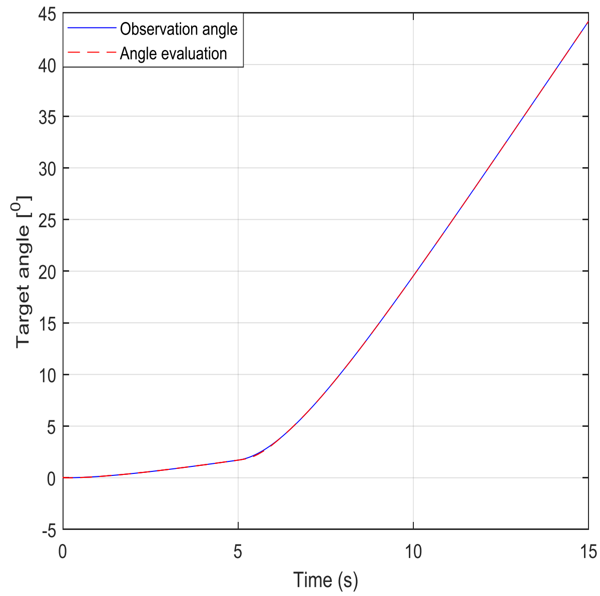

Fig. 5. Evaluation of angle coordinates $\hat{\varphi}_m$.

Nguyen Van Bang, "Synthesis of Target Angle Tracking System Combining Kalman Filter Algorithm with Fuzzy Logic Controller," *International Research Journal of Advanced Engineering and Science*, Volume 6, Issue 3, pp. 302-310, 2021.

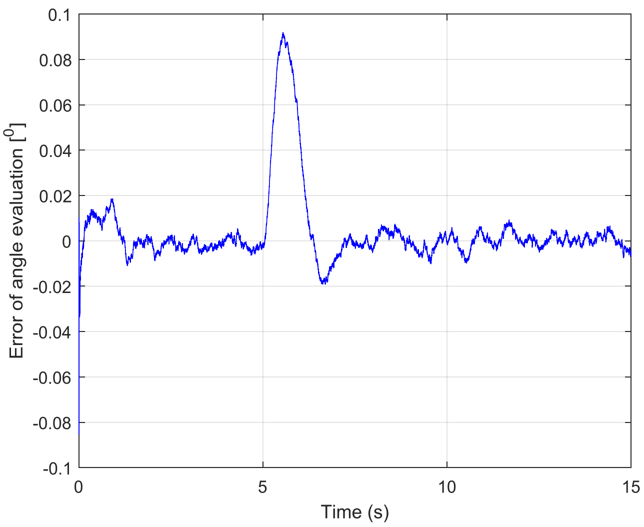

Fig. 6. Evaluation error of angular coordinates.

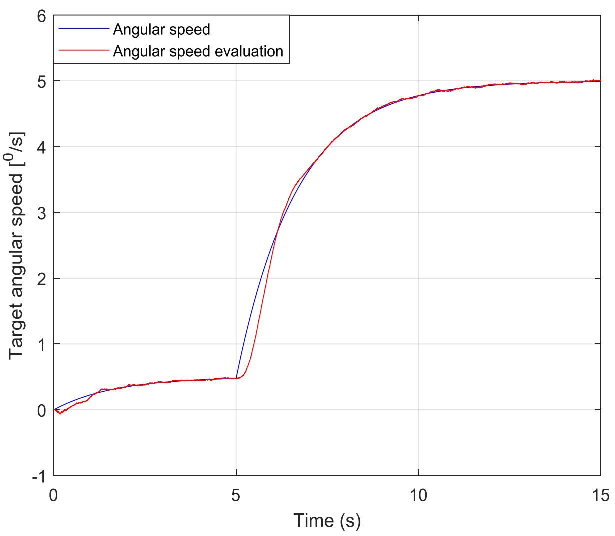

Fig. 7. Evaluation of angular speed $\hat{\omega}_m$.

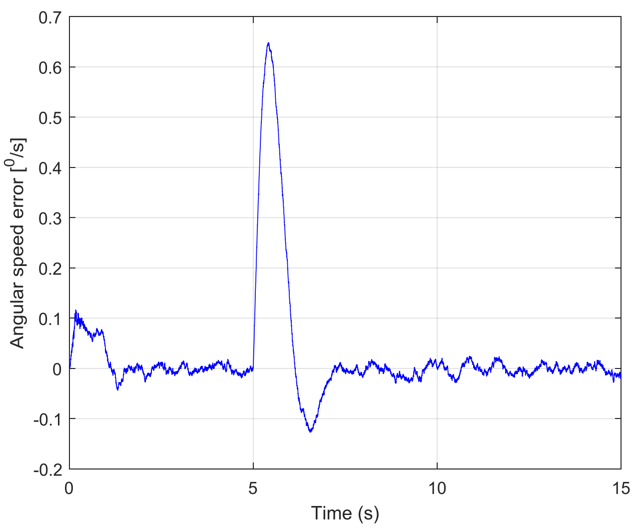

Fig. 8. Error of angular speed evaluation.

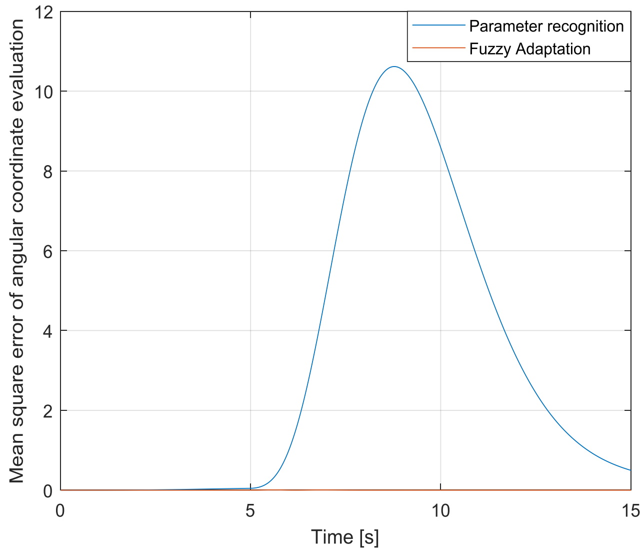

Fig. 9. Mean square error of angle coordinate evaluation.

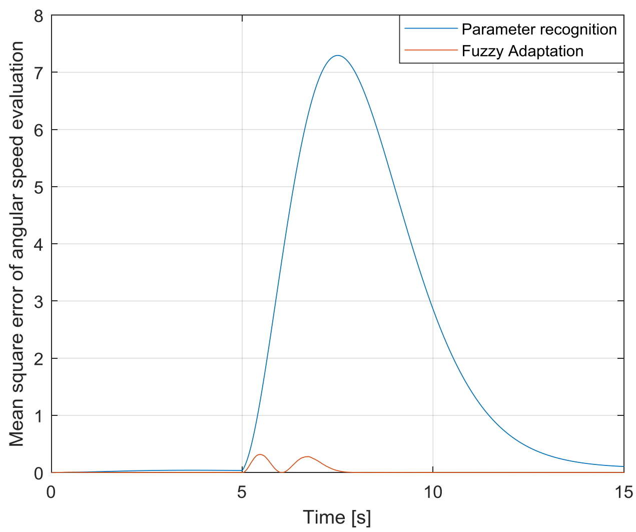

Fig. 10. Mean square error of angular speed evaluation.

The simulation results show that:

- When the target is maneuvering (at time 5s), the adaptation process will change the parameters of the model, ensuring a reduction in the error of evaluation of angle coordinates and angular speed.

- In the initial stage of maneuvering, because the model is not suitable, both the error of evaluating the angle coordinates and the angular speed are large, but this error value is still small.

- When the model's parameters are suitable, the evaluation error is reduced.

- When using fuzzy adaptive system, the quality is improved than using state model parameter recognition adaptive system. That is, the error when using an fuzzy adaptive system is reduced even if the target is maneuvering with the change angular speed. In the early stages of the adaptation process ($t = 5 \div 7s$), the errors of the two methods are roughly equivalent.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

IV. CONCLUSION

Adaptive filtering algorithm with maneuvering detection based on using fuzzy adaptive system, controlled the reasonableness of each component model and increased the weight of the most reasonable model in the recognition model, so the recognition accuracy is improved in the case of the maneuvering targets to varying degrees. The application of the adaptive fuzzy controller has received more simple algorithm, avoiding the interactive multi-model evaluation technique are very complex.

Target angle coordinate determination system with maneuvering detection based on the application of Kalman filter algorithm combined with fuzzy logic controller with high accuracy according to both position and speed in case of the maneuvering target.

From the simulation results, can be seen that the target angle coordinate system using the Kalman filter algorithm combined with a fuzzy logic controller significantly reduces the tracking error when the target changes maneuvering form. Capable of adapting to the target's maneuvering when the evaluation process progresses to the most suitable model.

REFERENCES

[1] Nguyen Van Bang, Hoang Van Ngoi, Nguyen Duy Tien. Synthesis of the maneuvering target coordinate determination algorithm on the basis of application of the interactive multi-model adaptive filtering technique. East European Scientific Journal, Vol 1, No.67, 3/2021, pp. 16-23.
[2] Nguyen Van Bang, Dang Cong Vu, Vu Quang Luong. Improving the quality of the line of sight angle coordinates filter of the self-guidance missile when using the snake style maneuvering model. International Journal of Software & Hardware Research in Engineering, Volume 9, Issue 4, April 2021, pp.7-13.
[3] Канащенкова. А.И и Меркулова. В.И. Авиационные системы радиоуправления (том 2). Радиотехника, Москва, 2013.
[4] Qian, G.H., Li, Y., Luo, R.J. One maneuvering frequency and the variance adaptive filtering algorithm for maneuvering target tracking. J. Radars 2(6).2017. pp. 258–264.
[5] D. Simon. Kalman filtering with state constraints - a survey of linear and nonlinear algorithm. Control Theory and Application, IET, Vol.4, No. 8, 2010, pp. 1303 - 1318.
[6] Rovisco Pais Av. Kalman and Extended Kalman Filters. Concept, Derivation and Properties, M. Isabel Ribeiro, 2014.
[7] Li X.R., Jilkov V.P. Survey of maneuvering target tracking. IEEE transaction on aerospace and electronic systems, 41 (4) 2015, pp. 1255-1321.
[8] Jiadong Ren, Xiaotong Zhang. Interactive Multi-model Target Maneuver Tracking Method Based on the Adaptive Probability Correction, International Conference on Swarm Intelligence, ICSI 2018. pp. 235-245.
[9] Amerongen, J. van. Intelligent Control, Lecture notes, University of Twente, The Netherlands, March, 2014.
[10] Uhrmeister. B. Kalman filter for a missile with radar and/or imaging sensor, J. Guid. Control Dyn, 2004.