Computers, Materials & Continua CMC, vol.63, no.1, pp.417-444, 2020

Intent Inference Based Trajectory Prediction and Smooth for UAS in Low-Altitude Airspace with Geofence

Qixi Fu[1], Xiaolong Liang[1, 2], Jiaqiang Zhang[1, *] and Xiangyu Fan[1, 2]

Abstract: In order to meet the higher accuracy requirement of trajectory prediction for Unmanned Aircraft System (UAS) in Unmanned Aircraft System Traffic Management (UTM), an Intent Based Trajectory Prediction and Smooth Based on Constrained State-dependent-transition Hybrid Estimation (CSDTHE-IBTPS) algorithm is proposed. Firstly, an intent inference method of UAS is constructed based on the information of ADS-B and geofence system. Moreover, a geofence layering algorithm is proposed. Secondly, the Flight Mode Change Points (FMCP) are used to define the relevant mode transition parameters and design the guard conditions, so as to generate the mode transition probability matrix and establish the continuous state-dependent-transition model. After that, the constrained Kalman filter (CKF) is applied to improve State-dependent-transition Hybrid Estimation (SDTHE) algorithm by applying equality constraint to the velocity of UAS in the straight phase and turning phase, respectively, and thus the constrained state-dependent-transition hybrid estimation (CSDTHE) algorithm is constructed. Finally, the results of intent inference and hybrid estimation are used to make trajectory prediction. Furthermore, each flight segment of trajectory is smoothed respectively by Rauch-Tung-Striebel (RTS) backward smooth method using the proposed CSDTHE-RTS algorithm, so as to obtain more accurate trajectory prediction results. The simulation shows that the proposed algorithm can reduce the errors of trajectory prediction and the time delay of intent inference.

Keywords: Trajectory prediction, unmanned aircraft system, geofence, intent inference, hybrid estimation, Rauch-Tung-Striebel (RTS) backward smooth.

1 Introduction

The rapid increase of low-altitude Unmanned Aircraft System (UAS) brings new opportunities and challenges to the development of air transportation [Ni, Yu and Rathinam (2017)]. The Unmanned Aircraft System Traffic Management (UTM) under development includes the UTM of the United States [Kopardekar, Rios and Prevot et al. (2016)], the Unmanned Aircraft System Operation Management System (UOMS) of China and the U-SPACE of Europe, etc. They aim to promote the safe and efficient

[1] National Key Laboratory of Air Traffic Collision Prevention, Air Force Engineering University, Xi'an, 710051, China.
[2] Shaanxi Province Laboratory of Meta-Synthesis for Electronic and Information System, Xi'an, 710051, China.
[*] Corresponding Author: Jiaqiang Zhang. Email: jiaqiang-z@163.com.
Received: 27 April 2019; Accepted: 11 July 2019.

operation of UAS in low-altitude airspace. Emerging airborne geofence system of UAS is an important part of UTM [Aweiss, Owens, Rios et al. (2018)], and it has been applied in some commercial-off-the- shelf UAS products, such as DJI [DJI (2019)] and Ardupilot [ArduPilot (2019)]. After obtaining the information of geofence published by UTM, geofence can divide the airspace into keep-in geofence (such as the operating area of plant protection UAS) and the keep-out geofence (such as no-fly zones, dangerous zone, thunderstorm zone, etc.) [Cho and Yoon (2018); Usach, Vila, Torens et al. (2018)]. NASA is developing a geofence system called Safeguard and has conducted a series of experiments [Dill, Young and Hayhurst (2016); Young, Dill, Hayhurst et al. (2017); Dill, Russell and Young (2018)]. They proposed to layer geofence to set warning boundaries, termination boundaries and hard boundaries, but did not propose a general geofence layering algorithm. UAS need trajectory prediction in flight to detect conflicts with geofence and other UAS in the same geofence and then to resolve conflicts autonomously [Seifert, Heller and Holzapfel (2018); Stevens, Rastgoftar and Atkins (2018); Fu, Liang, Zhang et al. (2019)]. Therefore, trajectory prediction is the basis of conflict detection.

Trajectory prediction methodologies can be divided into three categories: nominal, worst-case and probabilistic [Kuchar and Yang (2000)]. The nominal trajectory prediction simply projects current state of the aircraft to the future to obtain a single trajectory without considering any uncertainties. Although it is simple, the method can only be applied to cruise flight with small change of heading. Worst-case trajectory prediction takes into account all possible maneuvers of an aircraft to obtain all possible flight trajectories. It is generally applicable to trajectory prediction for Visual Flight Rule (VFR) flight with numerous maneuvers being performed. Probabilistic trajectory prediction takes into account the influence of uncertainties such as wind and navigation errors in trajectory prediction. It is the most widely used method for trajectory prediction, and it is also the method used in this paper. Hybrid estimation has been widely used in trajectory prediction because it can reduce the influence of noise. Furthermore, Interactive Multiple Model (IMM) [Bar-Shalom, Kirubarajan and Li (2001)] algorithm is extensively used because of its low complexity and small tracking error. However, since IMM regards mode transition as a Markov process with a fixed probability, it is only used in certain types of stochastic hybrid systems. In recent years, a hybrid estimation algorithm called State-dependent-transition hybrid estimation (SDTHE) [Seah and Hwang (2009)] has been developed and applied in many aerospace fields [Lee, Lee and Hwang (2016); Lee, Lee and Hwang (2016)]. SDTHE algorithm is also based on IMM algorithm framework, but it uses the information of guard conditions related to continuous states in stochastic hybrid systems, which greatly reduces the error of mode estimation. Compared with IMM algorithm, SDTHE algorithm can be applied to much broader range of stochastic hybrid systems. However, the SDTHE algorithm does not take into account the constraints of the target dynamics, which leads to a large error in trajectory prediction. Hence, the algorithm still has limitations.

Trajectory prediction based on hybrid estimation is only accurate in a short time, and prediction for long time will deviate from aircraft flight intent. Therefore, it is necessary to add intent inference to trajectory prediction. Trajectory prediction based on intent inference can be divided into two categories: one is based on the intent inference of pilot, which infers the operation of pilots according to different flight conditions of aircraft. It

uses discrete method in trajectory prediction [Rouse, Geddes and Curry (1987)]. The other is based on the position of aircraft where the real time position, heading and flight plan of the aircraft are obtained by ADS-B. This method can compute the possibility of aircraft going to different intent waypoints in real time, and then fly to the most likely intent waypoint. It uses a continuous research method. Since future UAS in low-altitude will operate autonomously without pilot, this paper adopts the second method based on intent inference. The typical method is Intent Inference Algorithms (IIA) [Krozel and Andrisani (2005, 2006)] proposed by Krozel. IIA can make correct intent inference based on aircraft dynamics, environmental information and flight plan. However, this method has a large time delay, and cannot get the real-time flight state and mode of aircraft. Moreover, the error of short-term trajectory prediction is very large. Hwang et al. use IMM algorithm and State-dependent-transition Hybrid Estimation (SDTHE) algorithm to predict trajectory based on intent inference. The proposed Intent Based Trajectory Prediction Based on IMM (IMM-IBTP) algorithm [Yepes, Hwang and Rotea (2007)] and Intent Based Trajectory Prediction Based on SDTHE (SDTHE-IBTP) algorithm [Hwang and Seah (2008)] combine the advantages of hybrid estimation algorithm which is more accurate in short-term prediction and IIA algorithm which has better performance in long-term prediction, and can get more accurate results. However, the along-track errors of these two algorithms increase gradually with time goes by, hence there is a large time delay in the long-term trajectory prediction. With the rapid increase of UAS in low-altitude airspace, it has become an important factor threatening the safety of low-altitude operation. However, few people have studied trajectory prediction method for UAS which is the basis of Sense and Avoid (SAA) technology in the presence of geofence. Compared with the manned aircraft, low-altitude UAS has smaller size and is flying autonomously, which requires higher accuracy for trajectory prediction. However, the accuracy of current trajectory prediction algorithm of manned aircraft cannot meet the requirements.

The remainder of this paper is structured as follows: Section 2 constructs the intent inference method for low-altitude UAS and proposes the geofence layering algorithm; Section 3 proposes the Constrained State-dependent-transition Hybrid Estimation (CSDTHE) algorithm for trajectory prediction; Section 4 adds Rauch-Tung-Striebel (RTS) backward smooth to the proposed Intent Based Trajectory Prediction based on Constrained State-dependent-transition Hybrid Estimation (CSDTHE-IBTP) algorithm to construct Intent Based Trajectory Prediction and Smooth based on Constrained State-dependent-transition Hybrid Estimation (CSDTHE-IBTPS) algorithm. The simulation results are analyzed in Section 5 and conclusions are summarized in Section 6.

2 Intent inference for UAS

The existing methods of intent inference for aircraft only aim at manned aircraft. And the sources of information in intent inference for UAS and manned aircraft are different. Therefore, intent inference for UAS needs to use information of emerging geofence system. The sketch of CSDTHE-IBTP algorithm presented in this paper is shown in Fig. 1. The algorithm utilizes the position, velocity and flight plan provided by ADS-B and the information of the airborne geofence system such as the type, location, tethering and permission of geofence.

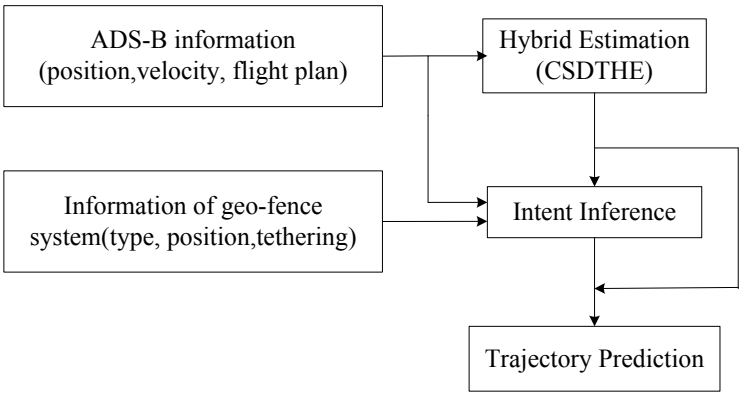


Figure 1: The sketch of CSDTHE-IBTP

The framework of CSDTHE-IBTP algorithm is:

Step 1. The state of UAS is estimated by CSDTHE according to the information provided by ADS-B.

Step 2. The intent of UAS is inferred by using ADS-B information, information of geofence system and results of hybrid estimation.

Step 3. Trajectory prediction is carried out by using results of hybrid estimation and intent inference.

We design several intent models in CSDTHE-IBTP algorithm which is divided into two categories: regulation intent and flight plan intent. Each intent includes information such as position, altitude and velocity. Examples of these two kinds of intents are shown in Tab. 1.

Table 1: Examples of intent

Regulation intent	Flight plan intent
Avoid other UAS	
Avoid keep-out geofence	Go to planned waypoint
Boundary maintaining with keep-in geofence	Return to planned waypoint
⋮	⋮

Each intent model I_l contains a series of waypoints, which are denoted by $\{W_l, W_l+1, ..., W_l+N_l\}$, where W_l is the first waypoint of intent model I_l, and the total number of waypoints is N_l+1. As shown in Fig. 2, the planned trajectory of UAS needs to cross keep-out geofence, which conflicts with the keep-out geofence. In Fig. 2, e_l denotes the unit vector of the UAS's intent heading pointing to intent waypoint. The green boundary around the periphery of keep-out geofence is layered geofence. The planned waypoint is a red one, and revised waypoint is a yellow one. If the UAS only uses flight plan intent, then the resulting trajectory is shown in Fig. 2(a), that is, the UAS

flies directly to the planned waypoint. If the UAS adopts the regulation intent, the corresponding trajectory is shown in Fig. 2(b), that is, the UAS will first avoid keep-out geofence, and then returns to planned waypoint. This paper assumes that UAS will adopt uniform flight rules when avoiding the geofence, i.e., clockwise avoidance, which can avoid the head-on conflict of UAS flying in opposite directions when avoiding geofence simultaneously. The revised waypoints about avoiding keep-out geofence are set at the vertex of the layered geofence, and the revised waypoint about avoiding keep-in geofence is the projection point of the planned waypoint on the nearest boundary of layered geofence. Based on our previous research in Fu et al. [Fu, Liang, Zhang et al. (2019)], geofence layering algorithm is as follows where scaling distance δ_b is set as follows:

$$\begin{cases} \delta_b > 0 & \text{for keep-out geofence} \\ \delta_b < 0 & \text{for keep-in geofence} \end{cases}$$

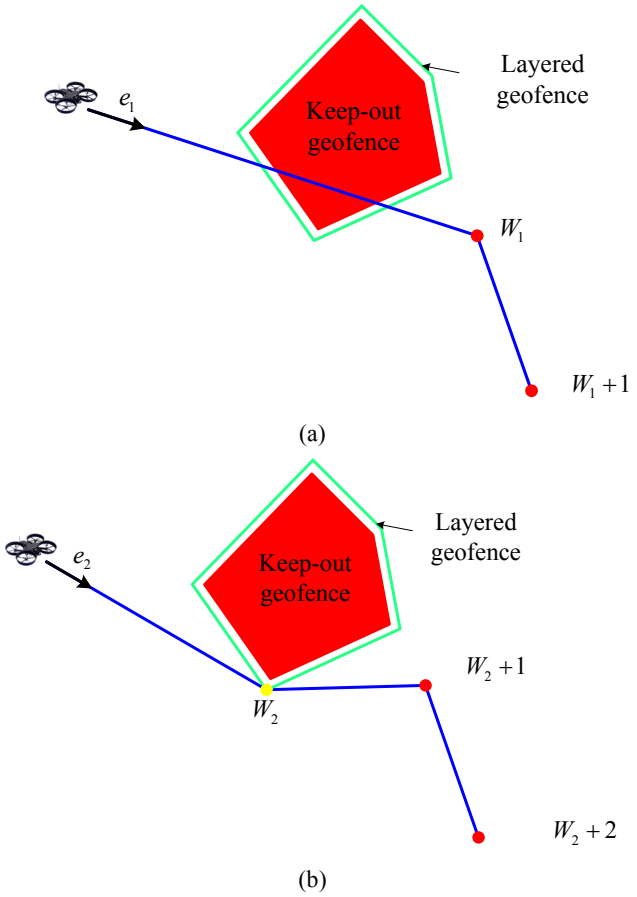


Figure 2: (a) Flying by flight plan intent only; (b) Flying by both flight plan intent and regulation intent

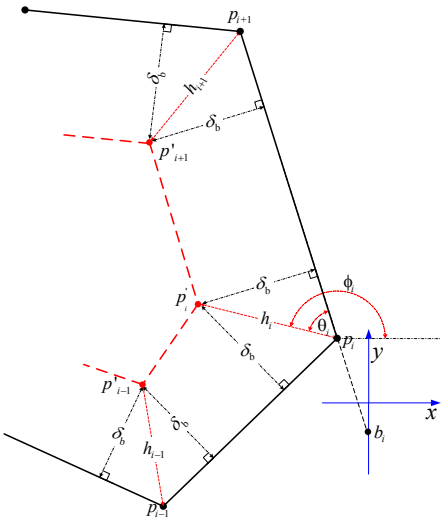


Figure 3: Geofence layering algorithm

Step 1. We set scaling distance δ_b, and get the geofence $g=(p_1,...,p_n)=[(x_1,y_1),(x_2,y_2),...,(x_n,y_n)]$, where p_i is the i th vertex of the geofence. Layered geofence g' initializes: $g'=g$.

Step 2. The slope of each edge of geofence is given by:

$$m_i=\frac{y_{i+1}-y_i}{x_{i+1}-x_i} \tag{1}$$

Interception of each edge is given by:

$$b_i=y_i-x_i*m_i \tag{2}$$

And θ_i is the half-angle of interior angle, which is given by:

$$\theta_i=\frac{1}{2}\left(\arctan\frac{y_{i+1}-y_i}{x_{i+1}-x_i}-\arctan\frac{y_{i-1}-y_i}{x_{i-1}-x_i}\right) \tag{3}$$

Step 3. In Fig. 3, the distance h_i along the angular bisectors between the new and old edges spacing δ_b after scaling is computed by:

$$h_i=\frac{|\delta_b|}{\sin\theta_i} \tag{4}$$

Convert θ_i to $0\sim 2\pi$ and get:

$$\phi_i=\theta_i+\arctan\frac{y_{i+1}-y_i}{x_{i+1}-x_i} \tag{5}$$

Step 4. If $\delta_b<0$ (scaling inward), then:

$$\begin{aligned} x_i^{'} &= \cos\phi_i * h_i + x_i \\ y_i^{'} &= \sin\phi_i * h_i + y_i \end{aligned} \tag{6}$$

If $\delta_b > 0$ (scaling outward), then:

$$\begin{aligned} x_i^{'} &= -\cos\phi_i * h_i + x_i \\ y_i^{'} &= -\sin\phi_i * h_i + y_i \end{aligned} \tag{7}$$

Finally, the layered geofence is given by: $g' = (p_1^{'},...,p_n^{'}) = [(x_1^{'},y_1^{'}),(x_2^{'},y_2^{'})...,(x_n^{'},y_n^{'})]$.

During the flight of UAS, the intent correlation for each intent is computed respectively. Then it flies to the intent waypoint with the maximum intent correlation. For time k, the intent correlation of intent I_l is defined as:

$$\Lambda(I_l,k) = \kappa_1(I_l,k)\kappa_2(I_l,k) \tag{8}$$

where κ_1 is spatial intent correlation and κ_2 is temporal intent correlation.

When UAS is in Constant Velocity (CV) or Constant Acceleration (CA) mode, intent correlation κ_1^* indicates the difference between unit vector e_{ac} and e_l of UAS. κ_1^* is defined as:

$$\kappa_1^*(I_l,k) = N_1(\psi_l(k) - \psi_{ac}(k); 0, \sigma_s^2) \tag{9}$$

where ψ_l is the angle of e_l; ψ_{ac} is the angle of unit vector e_{ac} which denotes the current actual heading of UAS (see Fig. 4); σ_s is the design parameter and is set as $\sigma_s = 5\deg$.

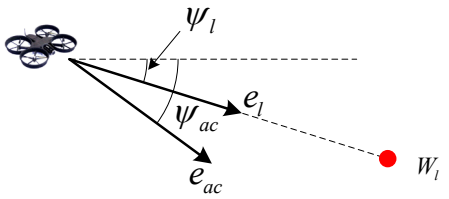


Figure 4: Definition of ψ_l and ψ_{ac}

Because κ_1^* is relevant to UAS's state, and state estimation has noise, there will be errors in κ_1^* at each moment. However, the intent inference of CV/CA mode needs to consider κ_1^* at each moment of linear dynamics in the past comprehensively, which may lead to wrong intent. Therefore, intent inference of CV/CA mode needs to reduce the influence of noise at the past moment on the basis of considering inferred intent at all times of the current linear phase. In this paper, fading memory filter is used to filter κ_1^* to get the intent κ_1 of CV/CA mode, which is given by:

$$\kappa_1(I_l,k) = \frac{1}{\gamma_k}\sum_{i=k_0}^{k} f^{k-k_0}\kappa_1^*(I_l,i),\ \gamma_k = \sum_{i=k_0}^{k} f^{k-k_0},\ 0 < f < 1 \tag{10}$$

where k_0 is the initial time of the current mode and f is fading memory factor, which is used to reduce the influence of inferred intent in the past time on the current intent inference exponentially.

When UAS is in Coordinated Turn (CT) mode, the spatial intent above can't get correct intent because the actual heading of UAS is always changing. In order to correct the wrong intent in CT mode, the change rate of κ_1^* is used to make intent inference of UAS in CT mode, that is:

$$\kappa_1(I_l,k)=\kappa_1^*(I_l,k)-\kappa_1^*(I_l,k-1) \tag{11}$$

Therefore, the spatial intent correlation κ_1 of UAS is defined as:

$$\kappa_1(I_l,k)=\begin{cases}\kappa_1^*(I_l,k)-\kappa_1^*(I_l,k-1) & \text{if flight mode = CT}\\ \dfrac{1}{\gamma_k}\displaystyle\sum_{i=k_0}^{k} f^{k-k_0}\kappa_1^*(I_l,i) & \text{if flight mode = CV or CA}\end{cases} \tag{12}$$

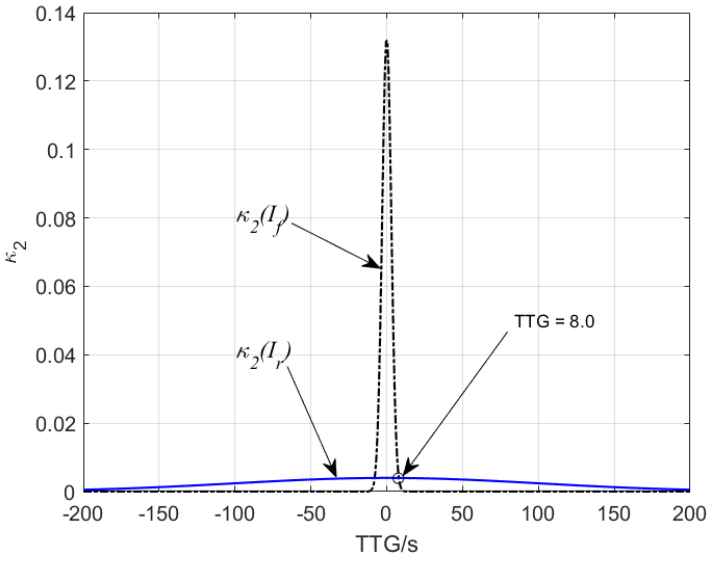


Figure 5: Comparison of different temporal intent correlations

If UAS arrives at different waypoints in different headings that vary greatly, then the spatial intent correlation of each intent model will have a great difference. In this scenario, the correct intent can be obtained. However, it is difficult to get the correct intent of UAS by using the spatial intent correlation only if the waypoints are approximately in a straight line. Therefore, this paper designs the temporal intent correlation κ_2:

$$\kappa_2(I_l,k)=\begin{cases}N_1(\mathrm{TTG}(W_{I_l});0,\sigma_{\mathrm{reg}}^2) & \text{if } I_l \text{ is regultion intent}\\ N_1(\mathrm{TTG}(W_{I_l});0,\sigma_{\mathrm{fp}}^2) & \text{if } I_l \text{ is flight plan intent}\end{cases} \tag{13}$$

where $\mathrm{TTG}(W_{I_l})$ is the time to arrive at W_{I_l}. As is known, UAS would choose the regulation intent when there are both regulation intent and flight plan intent at the same

time. This paper defines the safe time interval between UAS and geofence or other UAS as 8 seconds, which is used in Yang et al. [Yang, Yin and Shen (2017)], so we set $\sigma_{\text{reg}} = 100\,\text{s}$ and $\sigma_{\text{fp}} = 3\,\text{s}$. These two kinds of intent correlation is shown in Fig. 5, where I_f is the flight plan intent and I_r is the regulation intent. If $\text{TTG}(\text{W}_{I_l}) > 8.0\,\text{s}$, then $\kappa_2(I_f) < \kappa_2(I_r)$, that is, UAS prefers to choose the regulation intent, so as to ensure the safe time interval between UAS and geofence or other UAS; If $\text{TTG}(\text{W}_{I_l}) < 8.0\,\text{s}$, then $\kappa_2(I_f) > \kappa_2(I_r)$, UAS flies strictly by flight plan. Furthermore, the waypoint closer to the UAS's current position will be assigned higher weight based on κ_2.

Based on the principle of maximum intent correlation mentioned above, the inferred intent of UAS at time k th is as follows:

$$\hat{I}(k) = \arg\max_{I_l} \Lambda(I_l, k) \tag{14}$$

3 Trajectory prediction for UAS

The trajectory of UAS consists of a series of flight modes, including CV, CT and CA. Each flight mode has the characteristic of continuous state dynamics, and the mode transition between different flight modes depends on the continuous state of UAS. The dynamics of UAS can be regarded as a stochastic hybrid system, which has interactive continuous state and continuous-state- dependent modes. Since the dynamics of UAS is a linear system, we regard the UAS dynamic model as a stochastic linear hybrid system model.

3.1 Dynamic model of UAS

There exists a continuous state vector $\boldsymbol{x}(k) = [x_1(k), \cdots, x_n(k)]^T \in \boldsymbol{R}^n$, discrete mode $q(k) \in \Omega = \{1, 2, \cdots, r\}$, and measurement vector $\boldsymbol{z}(k) = [z_1(k), \cdots, z_p(k)]^T \in \boldsymbol{R}^p$, where k is discrete-time. For each mode $q(k)$, the dynamical equation of continuous state and measurement equation of UAS are as follows:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}_{q(k)}\boldsymbol{x}(k) + \boldsymbol{B}_{q(k)}\boldsymbol{w}_{q(k)}(k) \tag{15}$$

$$\boldsymbol{z}(k) = \boldsymbol{H}_{q(k)}\boldsymbol{x}(k) + \boldsymbol{v}_{q(k)}(k) \tag{16}$$

where $\boldsymbol{A}_{q(k)} \in \boldsymbol{R}^{n\times n}$, $\boldsymbol{B}_{q(k)} \in \boldsymbol{R}^{n\times m}$ and $\boldsymbol{H}_{q(k)} \in \boldsymbol{R}^{p\times n}$ are system, input and measurement matrices, and $\boldsymbol{w}_{q(k)}(k)$ and $\boldsymbol{v}_{q(k)}(k)$ are the process and measurement noise modeled as zero-mean white Gaussian noises with covariance $\boldsymbol{Q}_{q(k)}$ and $\boldsymbol{R}_{q(k)}$.

The mode transition is modeled as a continuous state-dependent-transition model. It is determined by a series of guard conditions $G(i, j), i, j = 1, 2, \cdots, r$, which can be expressed by a linear function as follows:

$$G(i, j) = \{[\boldsymbol{x}^T \ \boldsymbol{\theta}^T]^T \mid L_{x,ij}\boldsymbol{x} + L_{\boldsymbol{\theta},ij}\boldsymbol{\theta} \le 0\} \tag{17}$$

where $L_{x,ij} \in \boldsymbol{R}^{l\times n}$ and $L_{\boldsymbol{\theta},ij} \in \boldsymbol{R}^{l\times s}$ are constant matrices, and $\boldsymbol{\theta} \in \boldsymbol{R}^s$ is a random vector describing uncertainty of system.

In actual flight, the Flight Mode Change Point (FMCP) of UAS is not deterministic due to various uncertainties (such as wind, navigation errors and so on). According to the central limit theorem, $\boldsymbol{\theta}$ obeys Gaussian distribution $\boldsymbol{\theta} \sim N(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \Sigma_\theta)$ with expectation $\bar{\boldsymbol{\theta}}$ and covariance Σ_θ. If $[\boldsymbol{x}^T \ \boldsymbol{\theta}^T]^T \in G(i, j)$, the mode will change from $q(k) = i$ to $q(k) = j$, that is, mode transition depends on the continuous state $\boldsymbol{x}(k)$. The stochastic hybrid system mentioned above just is a stochastic linear hybrid system.

The trajectory of UAS contains a series of flight modes:

CV mode:

$$\boldsymbol{x}(k+1) = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}(k) + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \boldsymbol{w}(k) \tag{18}$$

where T is sample time.

CT mode:

$$\boldsymbol{x}(k+1) = \begin{bmatrix} 1 & 0 & \frac{\sin(\omega T)}{\omega} & \frac{-(1-\cos(\omega T))}{\omega} \\ 0 & 1 & \frac{(1-\cos(\omega T))}{\omega} & \frac{\sin(\omega T)}{\omega} \\ 0 & 0 & \cos(\omega T) & -\sin(\omega T) \\ 0 & 0 & \sin(\omega T) & \cos(\omega T) \end{bmatrix} \boldsymbol{x}(k) + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \boldsymbol{w}(k) \tag{19}$$

where ω is turning rate of UAS, which denotes left turn when it's positive and denotes right turn when it's negative.

CA mode:

$$\boldsymbol{x}(k+1) = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}(k) + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \boldsymbol{u}(k) + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \boldsymbol{w}(k) \tag{20}$$

where $\boldsymbol{u}(k)=[a_\xi, a_\eta]^T$ is acceleration of UAS.

Assuming that the position of UAS can be obtained by monitoring system (such as radar, GPS, etc.), the measurement vector is as follows:

$$\boldsymbol{z}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{x}(k) + \boldsymbol{v}(k) \tag{21}$$

3.2 Continuous state dependent transition

In this paper, a general architecture of mode transition model for the flight of UAS is proposed. As shown in Fig. 6, the transition from mode i to mode j is denoted as C_{ij}.

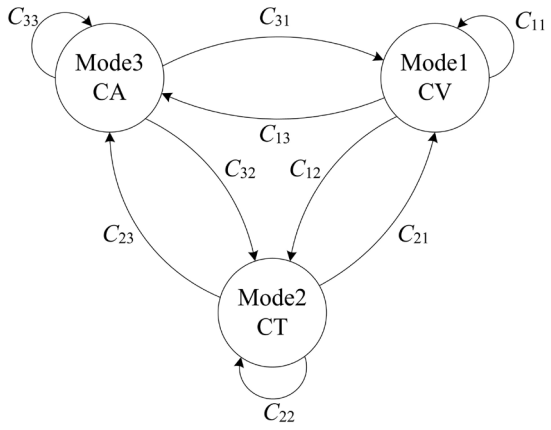


Figure 6: Flight mode transition

3.2.1 Relevant transition of CT mode

It corresponds to the mode transition from CV to CT when UAS starts to turn. Left-turn and right-turn are similar in terms of relevant guard conditions. Therefore, this paper takes left-turn as an example to describe the relevant conditions and parameter settings of relevant transition of CT mode. For each FMCP, the parameters used to define the guard conditions are shown in Fig. 7. $\boldsymbol{a}_i$ denotes the unit vector for heading of linear dynamics of UAS. d_s is the distance between the projection point of UAS's current position on the planned trajectory and waypoint W_l. d_1^* and d_2^* are random variables defining the distance between the UAS's starting turning position and ending turning position and waypoint W_l which are modeled as Gauss distribution with mean and covariance and are determined by the dimension of geofence and turning trajectory. Because of the uncertainties of UAS's navigation, the actual position of UAS at the beginning and end of the turn can only be determined randomly by the model.

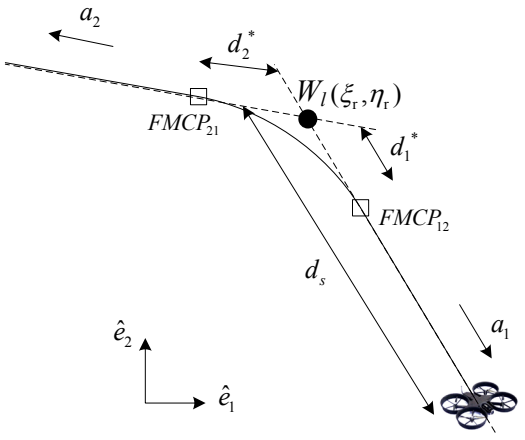


Figure 7: Mode transition (left turn)

d_s is defined as follow:

$$d_s = \boldsymbol{a}_i^T \begin{bmatrix} (\xi - \xi_{\mathrm{r}}) \\ (\eta - \eta_{\mathrm{r}}) \end{bmatrix} \tag{22}$$

where ξ and η are Cartesian coordinates of UAS; ξ_{r} and η_{r} are coordinates of waypoint W_l. If $d_s \le d_i^*$, the UAS begins to turn. Therefore, $d_s \le d_i^*$ is the transition condition, and its equivalent condition is given by:

$$L_i \boldsymbol{x} + L_{\theta} \boldsymbol{\theta}_i \le 0 \tag{23}$$

where $\boldsymbol{x} = [\xi \ \eta \ \dot{\xi} \ \dot{\eta}]^T$ is continuous state of UAS, $L_i = [\boldsymbol{a}_i^T \ 0 \ 0]$, $L_{\theta} = -1$ and $\boldsymbol{\theta}_i = d_i^* + \boldsymbol{a}_i^T [\xi_{\mathrm{r}} \ \eta_{\mathrm{r}}]^T$. The corresponding mode transition probability under the transition condition is $\Phi_l(L_i \boldsymbol{x}(k); \overline{\boldsymbol{\theta}}_i, \Sigma_{\theta_i})$, and the mean and covariance of $\boldsymbol{\theta}_i$ are $\overline{\boldsymbol{\theta}}_i$ and Σ_{θ_i}, respectively. $\Phi_l(L_i \boldsymbol{x}(k); \overline{\boldsymbol{\theta}}_i, \Sigma_{\theta_i})$ is a *l*-dimensional Gauss cumulative distribution function defined as $\Phi_l(\mu, \Sigma) \equiv p(\boldsymbol{y} \le 0)$. On the contrary, the mode transition probability of $\neg(d_s \le d_i^*)$ is $1 - \Phi_l(L_i \boldsymbol{x}(k); \overline{\boldsymbol{\theta}}_i, \Sigma_{\theta_i})$. If the UAS is in the middle of the turn where d_s reaches the minimum, i plus one, then the transition condition is changed to the one leaving the area. Guard conditions for turning are shown in Tab. 2. Moreover, the guard conditions for turning right are the same as for turning left.

Table 2: Guard conditions for turning at FMCP_{ij}

At the beginning of turn	At the end of turn
$C_{11} = C_{21} = C_{31} = \neg C_{12}$	$C_{12} = C_{22} = C_{32} = \neg C_{11}$
$C_{12} = C_{22} = C_{32} = L_1 \boldsymbol{x} + L_{\theta} \boldsymbol{\theta}_1 \le 0$	$C_{11} = C_{21} = C_{31} = L_2 \boldsymbol{x} + L_{\theta} \boldsymbol{\theta}_2 \le 0$
$C_{13} = C_{23} = C_{33} = \varnothing$	$C_{13} = C_{23} = C_{33} = \varnothing$
$L_{\theta} = -1$	$L_{\theta} = -1$
$\boldsymbol{\theta}_1 = d_1^* + \boldsymbol{a}_1^T [\xi_{\mathrm{r}} \ \eta_{\mathrm{r}}]^T$	$\boldsymbol{\theta}_2 = d_2^* + \boldsymbol{a}_2^T [\xi_{\mathrm{r}} \ \eta_{\mathrm{r}}]^T$

3.2.2 Relevant transition of CA mode

Relevant mode transition about CA involves both CV mode and CA mode, and UAS also performs mode transition near FMCP. For each FMCP, the parameters used to define the guard conditions are shown in Fig. 8, and relevant guard conditions is shown in Tab. 3.

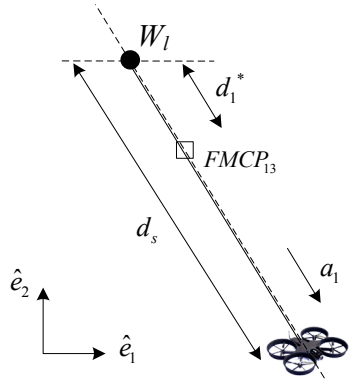


Figure 8: Mode transition from CV mode to CA mode

Table 3: Guard conditions for CA mode at $FMCP_{ij}$

Mode transition about CA mode
$C_{13} = C_{23} = C_{33} = \neg C_{13}$
$C_{13} = C_{23} = C_{33} = L_{\boldsymbol{x}}\boldsymbol{x} + L_{\boldsymbol{\theta}}\boldsymbol{\theta} \le 0$
$C_{12} = C_{22} = C_{32} = \varnothing$
$L_{\boldsymbol{\theta}} = -1$
$\boldsymbol{\theta}_1 = d_1^* + \boldsymbol{a}^T [\xi_r \quad \eta_r]^T$

3.3 Constraint model for UAS

The dynamics of UAS has the fixed flight mode, so we can add equality constraints to the dynamic model of UAS. Because the velocity constraint has faster detection times and fewer false alarms [Mann (2011)], it's used to replace the position constraint in order to ensure the flexibility of the model. Equality constraints are defined as:

$$D_{q(k)}\boldsymbol{x}(k) = \beta_{q(k)}(k) \tag{24}$$

where $D_{q(k)}(k)$ is a given matrix, and $\beta_{q(k)}(k)$ is usually a constant of zero or near zero.

Equality constraints can take two different forms by different flight modes of UAS. When UAS is traveling in a straight line, velocity can be constrained to be orthogonal to the unit vector $\boldsymbol{b}_i$ which is perpendicular to the straight line, that is, to constrain the UAS's velocity to always point to the current intent waypoint. Explicitly,

$$[0 \;\; 0 \;\; \boldsymbol{b}_i^T]\boldsymbol{x}(k) = 0 \tag{25}$$

The other form of constraint is the equality constraint in the CT mode. The turning trajectory model in CT mode is an arc which is tangent to both the former and the latter straight line segment, as shown in Fig. 9. The corresponding velocity constraint is to keep the UAS's velocity tangent to the arc, that is, the UAS's velocity is always orthogonal to the gradient of the arc. The circle where the arc is located is:

$$h_j = (\xi - \xi_c)^2 + (\eta - \eta_c)^2 = R_j^2 \tag{26}$$

and the gradient of the arc is:

$$\nabla h_j = [2(\xi - \xi_c) \;\; 2(\eta - \eta_c)] \tag{27}$$

then the equality constraint is given by:

$$[0 \;\; 0 \;\; \nabla h_j^T]\boldsymbol{x}(k) = 0 \tag{28}$$

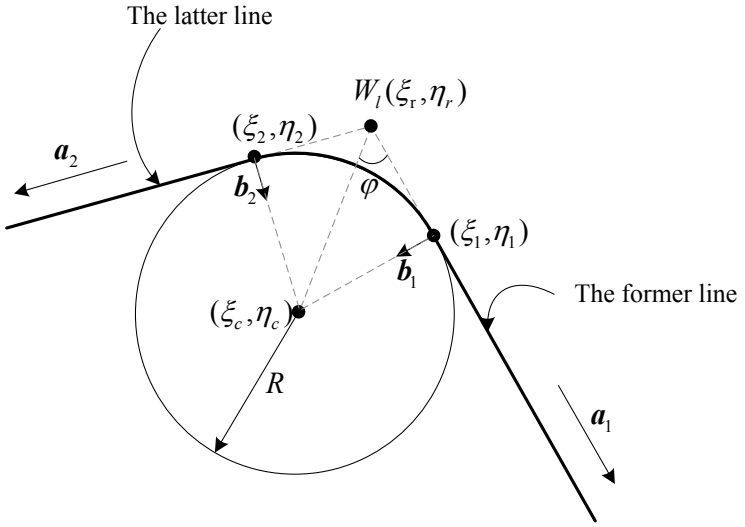


Figure 9: Arc between the former and the latter straight line segment

In Fig. 9, the turning circle is tangent to both the former and the latter straight lines. The radius R of the turning circle is given and relevant to the flight performance of the UAS. Then:

$$\begin{cases} \boldsymbol{a}_1^T[(\xi_1 - \xi_r) \;\; (\eta_1 - \eta_r)]^T = R\sin\varphi \\ \boldsymbol{a}_2^T[(\xi_2 - \xi_r) \;\; (\eta_2 - \eta_r)]^T = R\sin\varphi \\ a_{1\eta}(\xi_1 - \xi_r) - a_{1\xi}(\eta_1 - \eta_r) = 0 \\ a_{2\eta}(\xi_2 - \xi_r) - a_{2\xi}(\eta_2 - \eta_r) = 0 \end{cases} \tag{29}$$

where (ξ_1, η_1) is tangent point between the former straight line and the circle, and (ξ_2, η_2) is tangent point between the latter straight line and the circle, and φ is the half-angle between the former line and latter straight line. $a_{1\xi}$ and $a_{1\eta}$ are the two components of $\boldsymbol{a}_1$ on two coordinate axes. $a_{2\xi}$ and $a_{2\eta}$ are the two components of $\boldsymbol{a}_2$ on two coordinate axes. By formula (29), the beginning point (ξ_1, η_1) entering CT mode and end point (ξ_2, η_2) exiting CT mode of UAS can be determined.

3.4 Constrained state-dependent-transition hybrid estimation

In this paper, CSDTHE algorithm for constrained stochastic linear hybrid system is proposed. The algorithm includes a series of constrained Kalman filters (CKF) [Simon and Chia (2000)], each of which corresponds to the continuous state of UAS and has state constraints in each mode. $\boldsymbol{Z}_1^k \equiv \{z(1), z(2), ..., z(k)\}$ denotes the set of measurements up to time k, and mode probability $m^i(k)$ is defined as the probability that UAS will be in mode i when the state is $z(k)$ at time k. We suppose the probability of the mode i at time $k-1$ is given as:

$$m^i(k-1) \equiv p(q(k-1) = i \mid \boldsymbol{Z}_1^{k-1}), i = 1, 2, ..., r \tag{30}$$

Then we get the Gauss distribution of probability distribution function of continuous state as follows:

$$p(\boldsymbol{x}(k-1) \mid q(k-1) = i, \boldsymbol{Z}_1^{k-1}) = N(\boldsymbol{x}(k-1); \hat{\boldsymbol{x}}^i(k-1), \boldsymbol{P}^i(k-1)) \tag{31}$$

where $\hat{\boldsymbol{x}}^i(k-1)$ and $\boldsymbol{P}^i(k-1)$ are the mean and covariance of state vector $\boldsymbol{x}(k-1)$ predicted by the i th CKF at time $k-1$ respectively. Afterwards, $m^i(k)$ and $p(\boldsymbol{x}(k) \mid q(k) = i, \boldsymbol{Z}_1^k)$ can be computed by $z(k)$. The flow chart of CSDTHE is shown in Fig. 10.

1) Mixing

Firstly, we compute the mixing mode probability as:

$$\begin{aligned} m^{i|j}(k) &= p(q(k-1) = i \mid q(k) = j, \boldsymbol{Z}_1^{k-1}) \\ &= \frac{p(q(k) = j \mid q(k-1) = i, \boldsymbol{Z}_1^{k-1}) p(q(k-1) = i \mid \boldsymbol{Z}_1^{k-1})}{p(q(k) = j \mid \boldsymbol{Z}_1^{k-1})} \\ &= \frac{\pi_{ij}(k-1) m^i(k-1)}{\sum_{l=1}^{r} \pi_{lj}(k-1) m^l(k-1)} \end{aligned} \tag{32}$$

where $\pi_{ij}(k-1) := p(q(k) = j \mid q(k-1) = i, \boldsymbol{Z}_1^{k-1})$ is mode transition probability, which is computed as:

$$\begin{aligned} \pi_{ij}(k-1) &:= \int_{R^n} p(q(k) = j \mid q(k-1) = i, \boldsymbol{x}(k-1) = \boldsymbol{x}, \boldsymbol{Z}_1^{k-1}) \times p(\boldsymbol{x}(k-1) = \boldsymbol{x} \mid q(k-1) = i, \boldsymbol{Z}_1^{k-1}) d\boldsymbol{x} \\ &= \int_{R^n} \Phi_l(L_{x,ij}\boldsymbol{x} + L_{\theta,ij}\bar{\boldsymbol{\theta}}, L_{\theta,ij}\Sigma_\theta L_{\theta,ij}^T) N(\boldsymbol{x}; \hat{\boldsymbol{x}}^i(k-1), \hat{\boldsymbol{P}}^i(k-1)) d\boldsymbol{x} \\ &= \Phi_l(L_{x,ij}\hat{\boldsymbol{x}}^i(k-1) + L_{\theta,ij}\bar{\boldsymbol{\theta}}, L_{\theta,ij}\Sigma_\theta L_{\theta,ij}^T + L_{x,ij}\boldsymbol{P}^i(k-1)L_{x,ij}^T) \end{aligned} \tag{33}$$

Using the mixing probability, the initial conditions of the CKF matching mode j are obtained as follows:

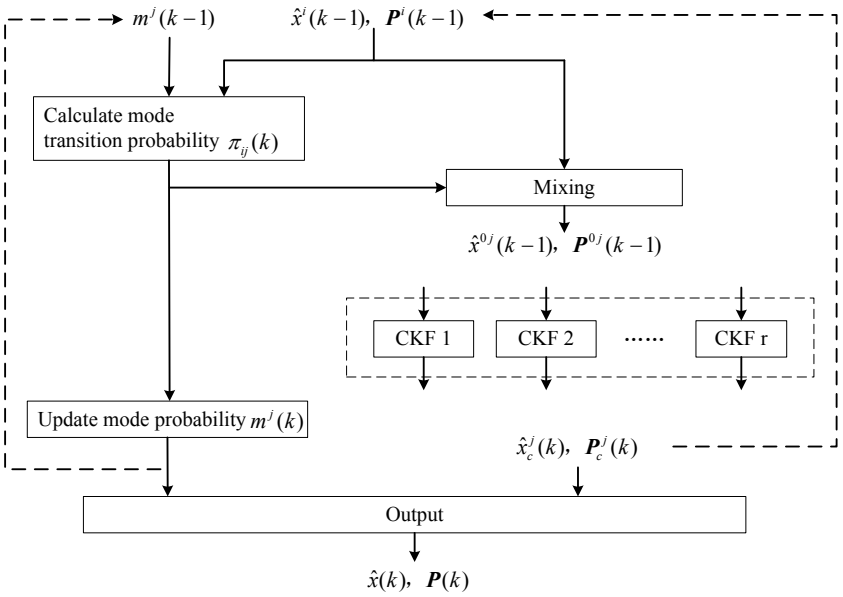


Figure 10: Flow chart of CSDTHE

$$\begin{aligned}\hat{\boldsymbol{x}}^{0j}(k-1) &= \sum_{i=1}^{r} m^{i|j}(k)\hat{\boldsymbol{x}}^i(k-1)\\ \boldsymbol{P}^{0j}(k-1) &= \sum_{i=1}^{r} m^{i|j}(k)\{\boldsymbol{P}^i(k-1)+[\hat{\boldsymbol{x}}^i(k-1)-\hat{\boldsymbol{x}}^{0j}(k-1)][\hat{\boldsymbol{x}}^i(k-1)-\hat{\boldsymbol{x}}^{0j}(k-1)]^T\}\end{aligned} \tag{34}$$

2) Mode-conditioned Estimation (CKF)

The state constraints of UAS need to be included in trajectory prediction algorithm, so this paper uses CKF to incorporate the state constraints of UAS by constraining the prior distribution of traditional Kalman filter. For mode j, the prediction equations are as follows:

$$\begin{aligned}\hat{\boldsymbol{x}}^j(k\,|\,k-1) &= \boldsymbol{A}_j\hat{\boldsymbol{x}}^{0j}(k-1)\\ \boldsymbol{P}^j(k\,|\,k-1) &= \boldsymbol{A}_j\boldsymbol{P}^{0j}(k-1)\boldsymbol{A}_j^T+\boldsymbol{B}_j\boldsymbol{Q}_j\boldsymbol{B}_j^T\end{aligned} \tag{35}$$

For $\boldsymbol{z}(k)$, prior mean and covariance can update as follows:

$$\begin{aligned}\hat{\boldsymbol{x}}^j(k) &= \hat{\boldsymbol{x}}^j(k\,|\,k-1)+\boldsymbol{K}(k)(\boldsymbol{z}(k)-\boldsymbol{H}_j\hat{\boldsymbol{x}}^j(k\,|\,k-1))\\ \boldsymbol{P}^j(k) &= (\mathbf{I}-\boldsymbol{K}(k)\boldsymbol{H}_j)\boldsymbol{P}^j(k\,|\,k-1)\end{aligned} \tag{36}$$

where

$$\boldsymbol{K}(k) = \boldsymbol{P}^j(k\,|\,k-1)\boldsymbol{H}_j^T(\boldsymbol{H}_j\boldsymbol{P}^j(k\,|\,k-1)\boldsymbol{H}_j^T+\boldsymbol{R}_j)^{-1} \tag{37}$$

In this paper, constraints are imposed by projecting unconstrained state estimation onto the constraint surface. The state constraints of UAS are equality constraints, which are equivalent to solving the following minimum problems:

$$\hat{\boldsymbol{x}}_c^j(k) = \arg\min_{\boldsymbol{x}}\{[\boldsymbol{x}-\hat{\boldsymbol{x}}^j(k)]^T\boldsymbol{W}[\boldsymbol{x}-\hat{\boldsymbol{x}}^j(k)]:\boldsymbol{D}_j\boldsymbol{x}=\beta_j\} \tag{38}$$

where $\boldsymbol{W}$ is a positive definite weighting matrix. The constrained optimization problem can be solved by Lagrange multiplier method. The cost function is defined as:

$$\Psi(\boldsymbol{x}) = [\boldsymbol{x} - \hat{\boldsymbol{x}}^j(k)]^T \boldsymbol{W} [\boldsymbol{x} - \hat{\boldsymbol{x}}^j(k)] + 2\boldsymbol{\lambda}^T (\boldsymbol{D}_j \boldsymbol{x} - \beta_j) \tag{39}$$

where $\boldsymbol{\lambda}$ is a Lagrange multiplier vector. We take the partial derivative and set it to zero to get:

$$\begin{aligned} \frac{\partial \Psi}{\partial \boldsymbol{x}} &= 2[\boldsymbol{x} - \hat{\boldsymbol{x}}^j(k)]^T \boldsymbol{W} + 2\lambda^T \boldsymbol{D}_j = 0 \\ \frac{\partial \Psi}{\partial \boldsymbol{\lambda}} &= \boldsymbol{D}_j \boldsymbol{x} - \beta_j = 0 \end{aligned} \tag{40}$$

After rearranging, we can get:

$$\hat{\boldsymbol{x}}_c^j(k) = \hat{\boldsymbol{x}}^j(k) - \boldsymbol{J}_j (\boldsymbol{D}_j \hat{\boldsymbol{x}}^j(k) - \beta_j) \tag{41}$$

where $\boldsymbol{J}_j$ is defined as:

$$\boldsymbol{J}_j \equiv \boldsymbol{W}^{-1} \boldsymbol{D}_j^T (\boldsymbol{D}_j \boldsymbol{W}^{-1} \boldsymbol{D}_j^T)^{-1} \tag{42}$$

The corresponding prior covariance is as follows：

$$\boldsymbol{P}_c^j(k) = (\boldsymbol{I} - \boldsymbol{J}_j \boldsymbol{D}_j)(\boldsymbol{A}_j \boldsymbol{P}^j(k) \boldsymbol{A}_j^T + \boldsymbol{B}_j \boldsymbol{Q}^j(k) \boldsymbol{B}_j^T)(\boldsymbol{I} - \boldsymbol{J}_j \boldsymbol{D}_j)^T \tag{43}$$

3) Mode probability update

Mode probability which is updated by Bayes rule is given by:

$$\begin{aligned} m^j(k) &= p(q(k) = j \mid \boldsymbol{Z}_1^k) \\ &= \frac{1}{c} p(\boldsymbol{z}(k) \mid q(k) = j, \boldsymbol{Z}_1^{k-1}) p(q(k) = j \mid \boldsymbol{Z}_1^{k-1}) \end{aligned} \tag{44}$$

where c is a normalization constant. Mode- conditioned likelihood function is given by:

$$p(\boldsymbol{z}(k) \mid q(k) = j, \boldsymbol{Z}_1^{k-1}) = N(\boldsymbol{z}(k) - \boldsymbol{H}^j \hat{\boldsymbol{x}}_j(k \mid k-1); 0, \boldsymbol{S}^j(k)) \tag{45}$$

where

$$\boldsymbol{S}^j(k) = \boldsymbol{H}^j(k) \boldsymbol{P}^j(k \mid k-1) \boldsymbol{H}^j(k)^T + \boldsymbol{R}^j \tag{46}$$

Prior mode probability is given by:

$$p(q(k) = j \mid \boldsymbol{Z}_1^{k-1}) = \sum_{i=1}^{r} \pi_{ij}(k-1) m^i(k-1) \tag{47}$$

4) Output

Applying mode-conditioned estimation $\hat{\boldsymbol{x}}_c^j(k)$, corresponding covariance $\boldsymbol{P}_c^j(k)$ and mode probability $m^j(k)$, we can get continuous state estimation $\hat{\boldsymbol{x}}(k)$ and covariance $\boldsymbol{P}(k)$ which are given by:

$$\begin{aligned} \hat{\boldsymbol{x}}(k) &= \sum_{j=1}^{r} \hat{\boldsymbol{x}}_c^j(k) m^j(k) \\ \boldsymbol{P}(k) &= \sum_{j=1}^{r} \{\boldsymbol{P}_c^j(k) + [\hat{\boldsymbol{x}}_c^j(k) - \hat{\boldsymbol{x}}(k)][\hat{\boldsymbol{x}}_c^j(k) - \hat{\boldsymbol{x}}(k)]^T\} m^j(k) \end{aligned} \tag{48}$$

5) Iterative Computation

Repeat Steps 1-4 to propagate state estimation to the future and we can get predicted trajectory.

4 Smoothing for predicted trajectory

The CSDTHE-IBTP algorithm enables the predicted trajectory to follow the trajectory determined by the waypoints and achieve a better mode matching, which can reduce the cross-track error of trajectory prediction. In order to reduce the cross-track error of trajectory prediction, a smoothing algorithm for predicted trajectory is proposed in this paper. At the initial time of trajectory prediction, the waypoint W_i would be allocated a control arrival time (CTA) t_i. Once the predicting time t reaches t_i, a pseudo-measurement $\tilde{z}_{t_i}$ is generated to correct predicted trajectory. The measurement model is given by:

$$\tilde{z}_{t_i} = [x \;\; y \;\; \sqrt{\dot{x}^2 + \dot{y}^2}]_{t_i} + \tilde{v}_{t_i} \tag{49}$$

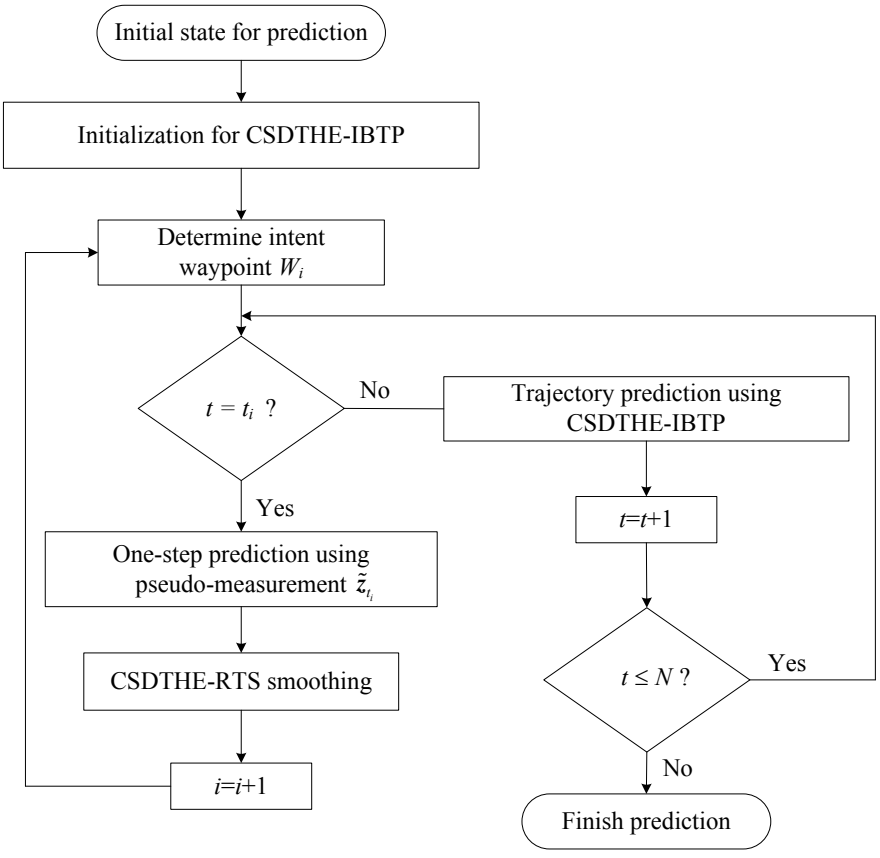


Figure 11: Flow chart of CSDTHE-IBTPS

where $\tilde{\boldsymbol{z}}_{t_i}$ is determined by the position and expected velocity of waypoint W_i, and $\tilde{\boldsymbol{v}}_{t_i}$ is additive noise which is modeled as Gauss distribution with zero-mean and covariance $\tilde{\boldsymbol{R}}_{t_i}$. $\tilde{\boldsymbol{R}}_{t_i}$ determines the confidence of waypoint information, that is how confident UAS will fulfill its intent. Explicitly, the smaller $\tilde{\boldsymbol{R}}_{t_i}$ is, the more confident it will be. One-step prediction can be made to obtain the predicted trajectory at time t_i based on pseudo-measurement. Since the pseudo-measurement model is nonlinear, it is necessary to replace CKF in CSDTHE algorithm with UKF [Julier and Uhlmann (2004)] in one-step prediction.

The flow chart of CSDTHE-IBTPS algorithm is shown in Fig. 11, and the total prediction time is T_N. The flow of CSDTHE-IBTPS algorithm is as follows:

Step 1. Intent inference is performed according to the initial state of UAS at the beginning of trajectory prediction, and intent waypoint W_i is determined.

Step 2. It judges whether the predicting time t is equal to CTA t_i. If it is equal, skip to Step 3, otherwise, skip to Step 4.

Step 3. The pseudo-measurements $\tilde{\boldsymbol{z}}_{t_i}$ are generated and UKF is used to make one-step prediction. Then the CSDTHE-RTS algorithm based on CSDTHE algorithm is used to smooth the predicted trajectory from t_{i-1} to t_i. Then $i = i+1$ and it determines the next intent waypoint W_i and returns to Step 2.

Step 4. The CSDTHE-IBTP algorithm is used to predict one-step trajectory based on results of hybrid estimation and intent inference.

Step 5. The predicting time t is added a time step T to determine whether the prediction time t is larger than T_N. If it is larger than T_N, the CSDTHE-IBTPS algorithm terminates, otherwise, returns to Step 2.

In this paper, RTS backward smoothing is added to CSDTHE algorithm proposed above, and then CSDTHE-RTS algorithm is constructed. CSDTHE-RTS algorithm can smooth the predicted trajectory according to the reverse recursive computation of the former n step prediction results. It's simpler and more rigorous than the one proposed in Nadarajah et al. [Nadarajah, Tharmarasa, Mcdonald, et al. (2012)]. One cycle of CSDTHE-RTS algorithm is as follows:

1) Smoothed mode probability

Posterior backward mode transition probability is given by:

$$\begin{aligned}\hat{\pi}_{ij}^{k+1,k} &\triangleq p\{m^j(k) \mid m^i(k+1), \boldsymbol{Z}_1^n\} \\ &= p\{m^j(k) \mid m^i(k+1), \boldsymbol{Z}_1^k\} \\ &= \frac{p\{m^i(k+1) \mid m^j(k), \boldsymbol{Z}_1^k\} p\{m^j(k) \mid \boldsymbol{Z}_1^k\}}{\sum_j p\{m^i(k+1) \mid m^j(k), \boldsymbol{Z}_1^k\} p\{m^j(k) \mid \boldsymbol{Z}_1^k\}} \\ &= \frac{\pi_{ji}(k) m^j(k \mid k)}{\sum_j \pi_{ji}(k) m^j(k \mid k)}\end{aligned} \tag{50}$$

where the first equation uses hypothesis in Nadarajah et al. [Nadarajah, Tharmarasa, Mcdonald et al. (2012)], that is, $m^j(k)$ is independent of the all measurement $\boldsymbol{Z}_{k+1}^n$ in future when $m^j(k+1)$ is given. The fourth equation uses of the hypothesis by analogy with IMM algorithm, that is $p\{m^i(k+1)\mid m^j(k),\boldsymbol{Z}_1^k\}=\pi_{ji}(k)$. Mode transition probability $\pi_{ji}(k)$ can be computed as follows:

$$\begin{aligned}\pi_{ij}(k) &:= \int_{R^n} p(q(k)=j\mid q(k+1)=i,\boldsymbol{x}(k+1)=\boldsymbol{x},\boldsymbol{Z}_1^{k+1})\times p(\boldsymbol{x}(k+1)=\boldsymbol{x}\mid q(k+1)=i,\boldsymbol{Z}_1^{k+1})d\boldsymbol{x}\\ &= \int_{R^n}\Phi_l(L_{x,ij}\boldsymbol{x}+L_{\theta,ij}\overline{\boldsymbol{\theta}},L_{\theta,ij}\Sigma_\theta L_{\theta,ij}^T)N(\boldsymbol{x};\hat{\boldsymbol{x}}^i(k+1\mid n),\hat{\boldsymbol{P}}^i(k+1\mid n))d\boldsymbol{x}\\ &= \Phi_l(L_{x,ij}\hat{\boldsymbol{x}}^i(k+1\mid n)+L_{\theta,ij}\overline{\boldsymbol{\theta}},L_{\theta,ij}\Sigma_\theta L_{\theta,ij}^T+L_{x,ij}\hat{\boldsymbol{P}}^i(k+1\mid n)L_{x,ij}^T)\end{aligned} \tag{51}$$

Smoothed mode probability $m^j(k\mid n)$ is given by:

$$\begin{aligned}m^j(k\mid n) &= \sum_i p\{m^j(k)\mid m^i(k+1),\boldsymbol{Z}_1^n\}p\{m^i(k+1)\mid\boldsymbol{Z}_1^n\}\\ &= \sum_i \hat{\pi}_{ij}^{k+1,k}m^i(k+1\mid n)\end{aligned} \tag{52}$$

where $m^i(k+1\mid n)\triangleq E[m^i(k+1)\mid\boldsymbol{Z}_1^n]$ can be obtained by the previous circle.

2) Smoothed mode-conditioned estimation

Posterior forward mode transition probability is given by:

$$\begin{aligned}\hat{\pi}_{ij}^{k,k+1} &\triangleq p\{m^i(k+1)\mid m^j(k),\boldsymbol{Z}_1^n\}\\ &= \frac{p\{m^j(k)\mid m^i(k+1),\boldsymbol{Z}_1^n\}p\{m^i(k+1)\mid\boldsymbol{Z}_1^n\}}{\sum_i p\{m^j(k)\mid m^i(k+1),\boldsymbol{Z}_1^n\}p\{m^i(k+1)\mid\boldsymbol{Z}_1^n\}}\\ &= \frac{\hat{\pi}_{ij}^{k+1,k}m^i(k+1\mid n)}{\sum_i\hat{\pi}_{ij}^{k+1,k}m^i(k+1\mid n)}\end{aligned} \tag{53}$$

Mode conditional smoothing:

$$\begin{aligned}\hat{\boldsymbol{x}}^j(k\mid n) &\triangleq E[\boldsymbol{x}(k)\mid m^j(k),\boldsymbol{Z}_1^n]\\ &= E\Big[E[\boldsymbol{x}(k)\mid m^j(k),m^i(k+1),\boldsymbol{Z}_1^n]\mid m^j(k),\boldsymbol{Z}_1^n\Big]\\ &= \sum_i\hat{\boldsymbol{x}}^{j\mid i}(k\mid n)\hat{\pi}_{ji}^{k,k+1}\end{aligned} \tag{54}$$

$$\boldsymbol{P}^j(k\mid n)=\sum_i\pi_{ji}^{k,k+1}[\boldsymbol{P}^{j\mid i}(k\mid n)+(\hat{\boldsymbol{x}}^{j\mid i}(k\mid n)-\hat{\boldsymbol{x}}^j(k\mid n))(\hat{\boldsymbol{x}}^{j\mid i}(k\mid n)-\hat{\boldsymbol{x}}^j(k\mid n))^T] \tag{55}$$

where $\hat{\boldsymbol{x}}^{j\mid i}(k\mid n)\triangleq E[\boldsymbol{x}(k)\mid m^j(k),m^i(k+1),\boldsymbol{Z}_1^n]$.

For simplicity, some approximations is made as follows:

$$\begin{aligned}&f(\boldsymbol{x}(k+1)\mid m^i(k+1),\boldsymbol{Z}_1^n)\approx N(\boldsymbol{x}(k+1);\hat{\boldsymbol{x}}^i(k+1\mid n),\boldsymbol{P}^i(k+1\mid n))\\ &f(\boldsymbol{x}(k)\mid m^i(k),\boldsymbol{Z}_1^k)\approx N(\boldsymbol{x}(k);\hat{\boldsymbol{x}}^i(k\mid k),\boldsymbol{P}^i(k\mid k))\end{aligned} \tag{56}$$

$\hat{\boldsymbol{x}}^{j\mid i}(k\mid n)$ can be obtained by Kalman backward smooth which is given by:

$$\begin{aligned}
&\hat{\boldsymbol{x}}^{j|i}(k\,|\,n) = \hat{\boldsymbol{x}}^{j}(k\,|\,k) + \boldsymbol{K}^{j}(k)(\hat{\boldsymbol{x}}^{i}(k+1\,|\,n) - \hat{\boldsymbol{x}}^{i|j}(k+1\,|\,k)) \\
&\boldsymbol{P}^{j|i}(k\,|\,n) = \boldsymbol{P}^{j}(k\,|\,k) + \boldsymbol{K}^{j}(k)(\boldsymbol{P}^{i}(k+1\,|\,n) - \boldsymbol{P}^{i|j}(k+1\,|\,k))(\boldsymbol{K}^{j}(k))^{T} \\
&\boldsymbol{K}^{j}(k) = \boldsymbol{P}^{j}(k\,|\,k)(\boldsymbol{A}^{i}(k))^{T}(\boldsymbol{P}^{i|j}(k+1\,|\,k))^{-1}
\end{aligned} \tag{57}$$

where $\hat{\boldsymbol{x}}^{i|j}(k+1\,|\,k) \triangleq E[\boldsymbol{x}(k+1)\,|\,m^{j}(k), m^{i}(k+1), \boldsymbol{Z}_{1}^{k}]$ and $\boldsymbol{P}^{i|j}(k+1\,|\,k)$ is error covariance matrix.

3) Smoothed overall estimation

$$\begin{aligned}
&\hat{\boldsymbol{x}}(k\,|\,n) = \sum_{i} m^{i}(k\,|\,n)\hat{\boldsymbol{x}}^{i}(k\,|\,n) \\
&\boldsymbol{P}(k\,|\,n) = \sum_{i} m^{i}(k\,|\,n)[\boldsymbol{P}^{i}(k\,|\,n) + (\hat{\boldsymbol{x}}^{i}(k\,|\,n) - \hat{\boldsymbol{x}}(k\,|\,n))(\hat{\boldsymbol{x}}^{i}(k\,|\,n) - \hat{\boldsymbol{x}}(k\,|\,n))^{T}]
\end{aligned} \tag{58}$$

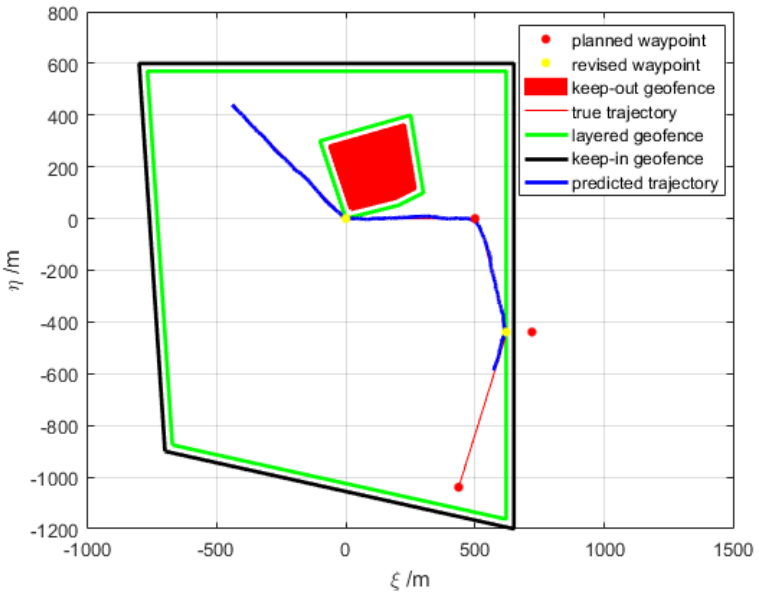
5 Simulation

In the simulation, there are both regulation intent and flight plan intent. Regulation intent has two categories: avoiding keep-out geofence and avoiding keep-in geofence. It is assumed that UAS can acquire measurements and flight plan (planned waypoints) through ADS-B system during flight, and location of the geofence can be obtained through the airborne geofence system. The scaling distance δ_b of geofence is set as ± 30 m, and predicting time step is $T = 0.5$ s. Total time of simulation is $T_N = 222$ s, and default velocity and turning rate of UAS is 10 m/s and 0.04π rad/s, respectively. The process noise covariance ($\boldsymbol{Q}_{CV}$ and $\boldsymbol{Q}_{CT}$), measurement noise covariance $\boldsymbol{R}$ and initial state ($\boldsymbol{x}_0$ and $\boldsymbol{P}_0$) of each mode are set as follows:

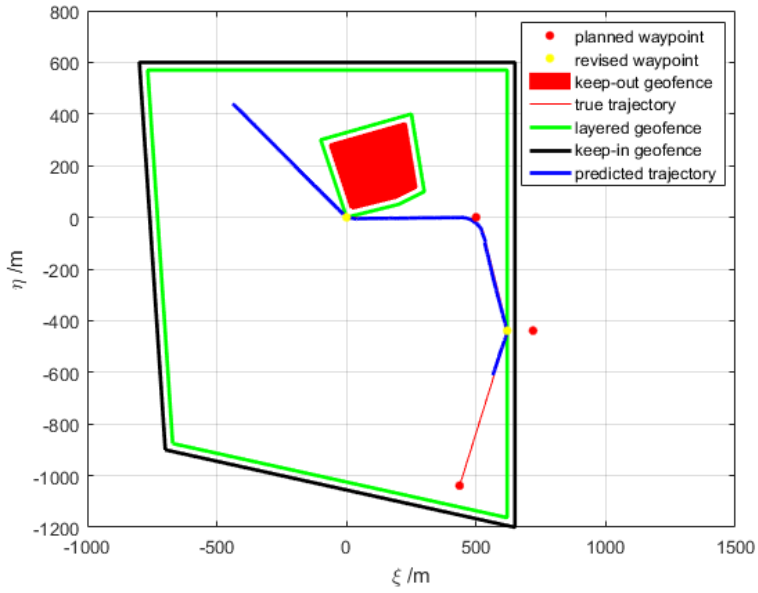
$$\boldsymbol{Q}_{\mathrm{CV}} = \begin{bmatrix} 0.003^2 & 0 \\ 0 & 0.003^2 \end{bmatrix},\ \boldsymbol{Q}_{\mathrm{CT}} = \begin{bmatrix} 0.001^2 & 0 \\ 0 & 0.001^2 \end{bmatrix},\ \boldsymbol{R} = \begin{bmatrix} 3^2 & 0 \\ 0 & 3^2 \end{bmatrix},$$

$$\tilde{\boldsymbol{R}}_{t_i} = diag([5^2 \quad 5^2 \quad 0.1^2]),\ \boldsymbol{x}_0 = \begin{bmatrix} -44 & 0 & 440 & 7 & -7 \end{bmatrix},\ \boldsymbol{P}_0 = \mathrm{diag}\left(\begin{bmatrix} 1^2 & 1^2 & 0.01^2 & 0.01^2 \end{bmatrix}\right)$$

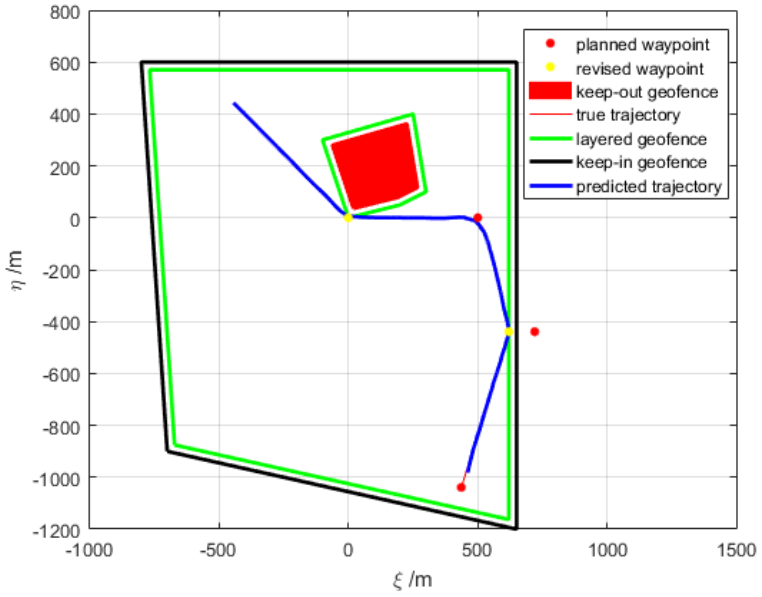
In Fig. 12(a), it's the trajectory prediction result of SDTHE-IBTP algorithm proposed in Hwang et al. [Hwang and Seah (2008)]. It can be seen that SDTHE-IBTP algorithm can get the revised waypoints according to regulation intent. The predicted trajectory can avoid keep-out geofence and keep-in geofence. Obviously, there is a notable error in the prediction results of the SDTHE-IBTP algorithm without constraints. In Fig. 12(b), it's the trajectory prediction result of CSDTHE-IBTP algorithm. Compared with SDTHE-IBTP algorithm, the predicted trajectory of CSDTHE-IBTP algorithm with equality constraints is smoother, which effectively reduces the cross-track error of the predicted trajectory. However, it is obvious that the predicted trajectory of CSDTHE-IBTP algorithm still has a large time delay. Fig. 12(c) is the prediction result of CSDTHE-IBTPS algorithm. It can be seen that the cross-track error of CSDTHE-IBTPS algorithm is approximately equivalent to CSDTHE-IBTPS algorithm. However, due to the smoothing based on pseudo- measurements, the time delay of trajectory prediction is greatly reduced.


planned waypoint
revised waypoint
keep-out geofence
true trajectory
layered geofence
keep-in geofence
predicted trajectory
800
600
400
200
0
-200
-400
-600
-800
-1000
-1200
-1000
-500
0
500
1000
1500
η /m
ξ /m

(a)


planned waypoint
revised waypoint
keep-out geofence
true trajectory
layered geofence
keep-in geofence
predicted trajectory
800
600
400
200
0
-200
-400
-600
-800
-1000
-1200
-1000
-500
0
500
1000
1500
η /m
ξ /m

(b)



(c)

Figure 12: (a) Trajectory prediction of SDTHE-IBTP; (b) Trajectory prediction of CSDTHE-IBTP; (c) Trajectory prediction of CSDTHE-IBTPS

The intent inference results of CSDTHE-IBTP algorithm and CSDTHE-IBTPS algorithm in trajectory prediction in Fig. 12 are shown in Fig. 13. In the prediction of CSDTHE-IBTP algorithm at first waypoint, intent inference has a delay of 6 seconds compared with real intent. And the longer the prediction time is, the longer the time delay between intent inference and real intent is, and the time delay has reached 40 s in the third waypoint prediction. CSDTHE-IBTPS algorithm has the same delay as CSDTHE-IBTPS algorithm in the prediction at first waypoint because there is no correction of pseudo-measurement. In the prediction of next two waypoints, the delay of intent inference is reduced because of the correction of the pseudo-measurements provided by the waypoints and smoothing of CSDTHE-RTS. At the third waypoint, the delay is only 2 s. It can be seen that the CSDTHE-IBTPS algorithm eliminates the disadvantage of the CSDTHE-IBTP algorithm that the time delay of intent inference increases with prediction time.

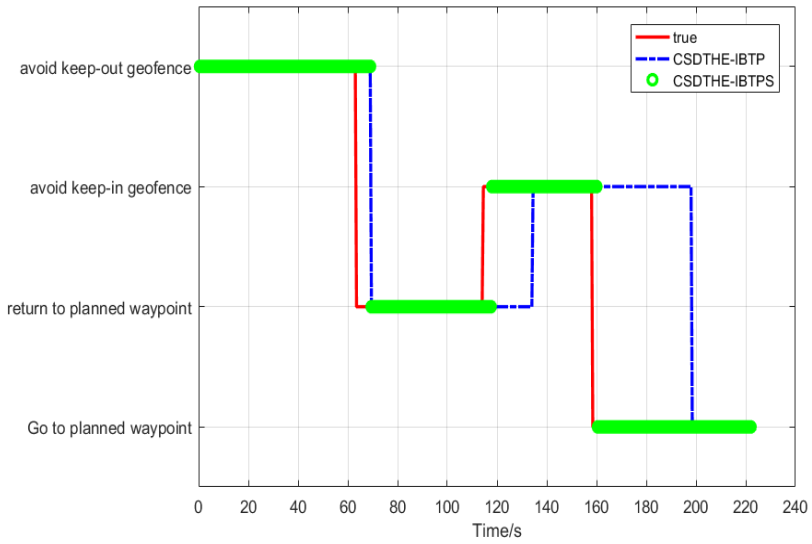
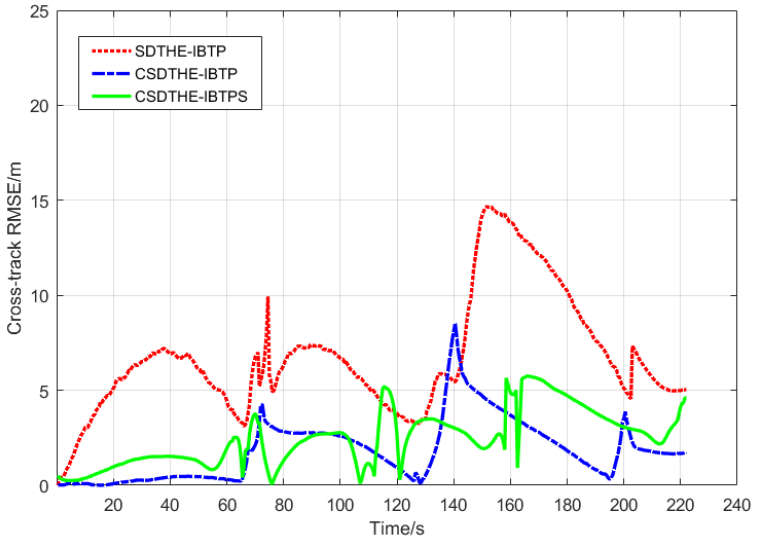


Figure 13: Comparison of inferred intent between CSDTHE-IBTP and CSDTHE-IBTPS

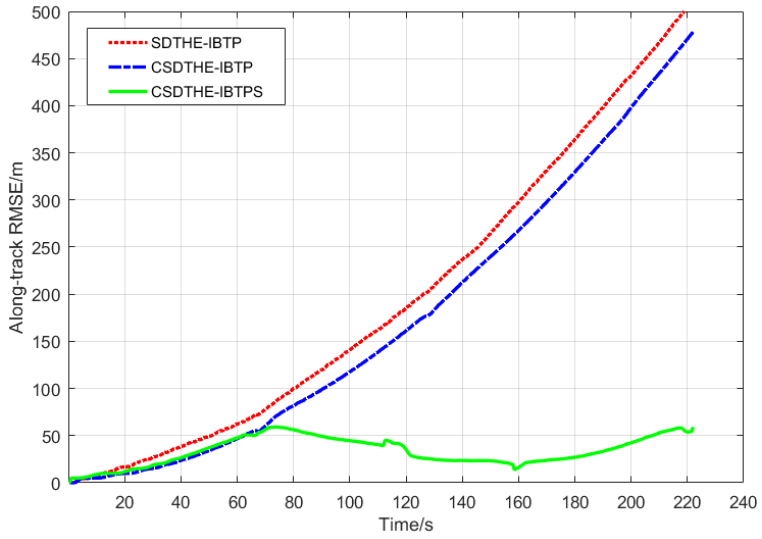
In this paper, the SDTHE-IBTP, CSDTHE-IBTP and CSDTHE-IBTPS algorithm are used to simulate the scenarios in Fig. 12 100 times by Monte Carlo simulations, and the Root Mean Square Error (RMSE) of cross-track and the RMSE of along-track are computed, respectively, as shown in Fig. 14. As can be seen from Fig. 14(a), the cross-track RMSE of SDTHE-IBTP algorithm is larger than that of CSDTHE-IBTP algorithm and CSDTHE-IBTPS algorithm. The cross-track RMSE of CSDTHE-IBTP algorithm is effectively reduced compared with SDTHE-IBTP algorithm because of the equality constraints on velocity in trajectory prediction. For different modes, the cross-track RMSE of CV/CA mode is smaller (see 72 s-128 s, 140 s-195 s and 200 s-222 s). Because the velocity is constrained to the direction of the intent waypoint, the cross-track RMSE decreases continuously. Fig. 14(a) also shows that the RMSE of the cross-track of CT mode is dependent on the turning angle. For the same CT mode, the RMSE of the cross-track increases with the increase of the turning angle; for different CT modes, the larger the turning angle of the whole process is, the larger the maximum value of the RMSE of the cross-track is. Because CSDTHE-RTS algorithm is used for smoothing, the maximum value of cross-track RMSE of CSDTHE-IBTPS algorithm in CT mode is lower than that of CSDTHE-IBTP algorithm, and the cross-track RMSE in CV/CA mode is approximately unchanged.

In Fig. 14(b), it shows that the performance of CSDTHE-IBTP algorithm is not significantly improved compared with SDTHE-IBTP algorithm in terms of along-track RMSE Because of the application of pseudo-measurements at each waypoint, the CSDTHE-IBTPS algorithm eliminates the time-cumulative characteristics of along-track RMSE in the CSDTHE-IBTP algorithm in trajectory prediction after the first waypoint. That is, the along-track RMSE predicted by CSDTHE-IBTPS algorithm will not increase with time, as shown in Fig. 14(b). But before reaching the first waypoint, the RMSE of the two algorithms are basically the same because the information of the waypoints is not used. To sum up, it is not difficult to find that the performance of CSDTHE-IBTP

algorithm is basically the same as CSDTHE-IBTPS algorithm when the number of predicted waypoints is 1. When the number of predicted waypoints is larger than 1, the accuracy of trajectory prediction of CSDTHE-IBTPS algorithm is greatly improved compared with CSDTHE-IBTP algorithm.



（a）Cross-track RMSE



（b）Along-track RMSE

Figure 14: Comparison of RMSE of different algorithms

6 Conclusions

In this paper, the problem of trajectory prediction for low-altitude UAS is studied. Due to the rapid increase of low-altitude UAS, requirements for higher accuracy of trajectory prediction of low-altitude UAS are put forward, and the characteristics of confined airspace caused by the emergence of geofence make the existing trajectory prediction algorithms difficult to meet the trajectory prediction requirements of low-altitude UAS. To solve this problem, this paper proposes a CSDTHE-IBTPS algorithm with higher accuracy.

CSDTHE-IBTPS algorithm can reduce the along-track RMSE and cross-track RMSE by combining the spatial and temporal information of UAS's intent the algorithm consists of two parts: 1) CSDTHE prediction based on intent inference. In the CSDTHE algorithm, the continuous state-dependent-transition model is adopted. The parameters of mode transition are defined by using FMCP points, and the conditions of mode transition are designed. Equality constraint is imposed on the velocity of UAS in two stages: straight line and turning stage phase. Using information of ADS-B and emerging geofence system, the intent inference is carried out, and a geofence layering algorithm is proposed, so that the predicted trajectory of UAS is consistent with the intent. 2) CSDTHE-RTS smoothing for predicted trajectory. At the CTA of the intent waypoint, the algorithm generates pseudo-measurement based on the information of the waypoint and uses the proposed CSDTHE-RTS algorithm to smooth the predicted trajectory. The simulation results show that the algorithm can eliminate the time-cumulative effect of along-track RMSE in trajectory prediction. It improves the accuracy of trajectory prediction and the real-time of intent inference greatly compared with the existing algorithms. Furthermore, the proposed algorithm can provide theoretical reference for UTM system.

The future research of trajectory prediction for low-altitude UAS can focus on the dynamic geofence of which the location is time-varying, so that the information of geofence in the trajectory prediction algorithm is more correspond to the actual low-altitude environment.

Data availability: The results.mat data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgment: This work was financially supported by the Major Program of National Natural Science Foundation of China; the National Natural Science Foundation of China [Grant No. 61703427].

References

ArduPilot (2019): Mission planner configuration and tuning.

http://ardupilot.org/planner/docs/mission-planner-configuration-and-tuning.html?highlight=geofence

Aweiss, A. S.; Owens, B. D.; Rios, J. L.; Homola, J. R.; Mohlenbrink, C. P. (2018): Unmanned aircraft systems (UAS) traffic management (UTM) national campaign Ⅱ. *Proceedings of the 2018 AIAA Information System*, pp. 1-16.

Bar-Shalom, Y.; Kirubarajan, T.; Li, X. R. (2001): Estimation with applications to tracking and navigation. *Wiley*, pp. 476-477.

Cho, J. W.; Yoon, Y. J. (2018): How to assess the capacity of urban airspace: a topological approach using keep-in and keep-out geofence. *Transportation Research Part C- Emerging Technologies*, vol. 92, no. 1, pp. 137-149.

Dill, E. T.; Russell, V. G.; Young, S. S. (2018): Safeguard: flight test results of an on-board system designed to assure conformance to geospatial limitations. *Proceedings of 2018 Digital Avionics Systems Conference*.

Dill, E. T.; Young, S. D.; Hayhurst, K. J. (2016): SAFEGUARD: an assured safety net technology for UAS. *Proceedings of 2016 the Digital Avionics Systems Conference*.

DJI (2019): Limited flight zone query. https://www.dji.com/cn/flysafe/geo-map.

Fu, Q. X.; Liang, X. L.; Zhang, J. Q.; He, L. L.; Zhou, W. Y. (2019): Design and implementation of autonomous flight unmanned aircraft system geo-fence algorithm. *Journal of Xi'an Jiaotong University*, vol. 53, no. 5, pp. 167-175.

Hwang, I.; Seah, C. (2008): Intent based probabilistic conflict detection for the next generation air transportation system. *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2040-2059.

Julier, S. J.; Uhlmann, J. K. (2004): Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401-422.

Kopardekar, P.; Rios, J.; Prevot, T.; Johnson, M.; Jung, J. et al. (2016): Unmanned aircraft system traffic management (UTM) concept of operations. *Proceedings of the 16th AIAA Aviation Technology, Integration, and Operations Conference*, pp. 1-16.

Krozel, J.; Andrisani, D. (2005): Intent inference and strategic path prediction. *Proceedings of the AIAA Guidance, Navigation, and Control Conference*.

Krozel, J.; Andrisani, D. (2006): Intent inference with path prediction. *AIAA Journal of Guidance, Control and Dynamics*, vol. 29, no. 2, pp. 225-236.

Kuchar, J.; Yang, L. (2000): A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 178-189.

Lee, J.; Lee, S.; Hwang, I. (2016): Hybrid system modeling and estimation for estimated time of arrival prediction in terminal airspace. *AIAA Journal of Guidance, Control and Dynamics*, vol. 39, no. 4, pp. 903-910.

Lee, S.; Lee, J.; Hwang, I. (2016): Maneuvering spacecraft tracking via state-dependent adaptive estimation. *AIAA Journal of Guidance, Control and Dynamics*, vol. 39, no. 9, pp. 2034-2043.

Mann, G. W. (2011): *Aircraft Taxiway Conformance Monitoring with Constrained Stochastic Linear Hybrid Systems (Ph.D. Thesis)*, Department of Aeronautics and Astronautics, Purdue University. West Lafayette, IN.

Nadarajah, N.; Tharmarasa, R.; Mcdonald, M.; Kirubarajan, T. (2012): IMM forward filtering and backward smoothing for maneuvering target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2673-2078.

Ni, D. H.; Yu, G. Z.; Rathinam, S. (2017): Unmanned aircraft system and its applications in transportation. *Journal of Advanced Transportation*, vol. 2017, pp. 1-2.

Rouse, W. B.; Geddes, N. D.; Curry, R. E. (1987): An architecture for intelligent interfaces: outline of an approach to supporting operators of complex systems. *Human Computer Interaction*, vol. 3, no. 2, pp. 87-122.

Seah, C. E.; Hwang, I. (2009): State estimation for stochastic linear hybrid systems with continuous-state-dependent-transitions: an IMM approach. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 1, pp. 376-392.

Seiferth, D.; Heller, M.; Holzapfel, F. (2018): Automatic safe area detection for novel unmanned air vehicle. *Proceedings of 2018 4th International Conference on Control, Automation and Robotics*.

Simon, D.; Chia, T. L. (2000): Kalman filtering with state equality constraints. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 128-136.

Stevens, M. N.; Rastgoftar, H.; Atkins, E. M. (2018): Geo-fence boundary violation detection in 3D using triangle weight characterization with adjacency. *Journal of Intelligent & Robotic Systems*, vol. 92, no. 1, pp. 1-12.

Usach, H.; Vila, J. A.; Torens, C.; Adolf, F. (2018): Architectural design of a safe mission manager for unmanned aircraft systems. *Journal of Systems Architecture*, vol. 90, no. 2, pp. 94-108.

Yepes, J.; Hwang, I.; Rotea, M. (2007): New algorithm for aircraft intent inference and trajectory prediction. *AIAA Journal of Guidance, Control and Dynamics*, vol. 30, no. 2, pp. 370-382.

Yang, J.; Yin, D.; Shen, L. C. (2017): Reciprocal geometric conflict resolution on unmanned aerial vehicles by heading control. *AIAA Journal of Guidance, Control and Dynamics*, vol. 40, no. 10, pp. 2511-2523.

Young, S. D.; Dill, E. T.; Hayhurst, K. J.; Gilabert, R. V. (2017): Safeguard: progress and test results for a reliable independent on-board safety net for UAS. *Proceedings of 2017 Digital Avionics Systems Conference*.