An Interval Type-2 Fuzzy Logic Based Map Matching Algorithm for Airport Ground Movements

Xinwei Wang, Alexander E.I. Brownlee, Michal Weiszer, John R. Woodward, Mahdi Mahfouf and Jun Chen

***Abstract*—Airports and their related operations have become the major bottlenecks to the entire air traffic management system, raising predictability, safety and environmental concerns. One of the underpinning techniques for digital and sustainable air transport is airport ground movement optimisation. Currently, real ground movement data is made freely available for the majority of aircraft at many airports. However, the recorded data is not accurate enough due to measurement errors and general uncertainties. In this paper, we aim to develop a new interval type-2 fuzzy logic based map matching algorithm, which can match each raw data point to the correct airport segment. To this aim, we first specifically design a set of interval type-2 Sugeno fuzzy rules and their associated rule weights, as well as the model output, based on preliminary experiments and sensitivity tests. Then, the fuzzy membership functions are fine-tuned by a particle swarm optimisation algorithm. Moreover, an extra checking step using the available data is further integrated to improve map matching accuracy. Using the real-world aircraft movement data at Hong Kong Airport, we compared the developed algorithm with other well-known map matching algorithms. Experimental results show that the designed interval type-2 fuzzy rules have the potential to handle map matching uncertainties, and the extra checking step can effectively improve map matching accuracy. The proposed algorithm is demonstrated to be robust and achieve the best map matching accuracy of over 96% without compromising the run time.**

***Index Terms*—ADS-B, airport ground movement, interval type-2 fuzzy logic, map matching**

I. Introduction

A recent Eurocontrol document has reported that the total volume of air traffic will grow up to over 16 million by 2040, which increases 53% compared to 2017 [1]. The early signs are that this trend of growth will return following the present pandemic [2]. As air traffic demand continues to increase, airports and their related operations have become the major bottlenecks to the entire air traffic management (ATM) system, raising predictability, safety and environmental concerns. As a response, airports, airlines, air traffic control services and aircraft manufacturers are investing heavily in automation and digitalisation in order to cope with the rising demand and constraints of the existing infrastructure [3]. One of the underpinning techniques for automating and digitalising air transport is airport ground movement (AGM) optimisation, which has received much attention in recent years [4], [5], [6], [7], [8], [9]. AGM optimisation is in essence a scheduling and routing problem, involving directing aircraft to their destinations in a safe and timely manner [10]. As a major bottleneck in the whole ATM system, the AGM optimisation also integrates with other airport operational problems, such as runway sequencing [11], [12], [13] and stand/gate allocation [14], [15].

Although AGM only constitutes a small portion of the flight, it contributes a disproportionally large amount of emissions and fuel cost for each aircraft. This is due to that the aircraft engine is supposed to work at the cruising speed, and thus operates inefficiently when an aircraft is moving at a low speed on the ground. According to a technical report from Honeywell [16], the aircraft fuel consumption during inefficient taxiing at congested airports accounts for 6% of the entire fuel cost for single-aisle aircraft in a short-haul flight, resulting in fuel cost 5 million tonnes worldwide.

To address the AGM optimisation problem, it is pivotal to learn from historical real-world ATM data. Koeners and Rademaker [17] pointed out the uncertainty of the aircraft taxiing speed is a major cause of uncertainty in airport ground operations. To tackle this uncertainty, they evaluated the airport traffic flow and speed distributions by using aircraft taxiing data from Schiphol airport. Ravizza $et\ al.$ [18] tested several statistical regression and machine learning approaches to accurately predicting taxiing times. Historical data from two major European airports, Stockholm-Arlanda Airport and Zurich Airport, were applied for cross-validation. Results indicate that a fuzzy rule-based system outperform other approaches in terms of prediction accuracy. Brownlee $et\ al.$ [7] utilised a set of real AGM data for one day of operations at Manchester Airport, and generated additional virtual AGM data to simulate taxi time uncertainties in AGM optimization. Note only when the provided historical AGM data is precise enough, the performance of the AGM optimisation under taxi time uncertainties can be improved.

Currently, Automatic Dependent Surveillance Broadcast (ADS-B), which broadcasts aircraft GPS position information, is widely used for air traffic monitoring [19]. EUROCONTROL and the American Federal Aviation Administration have mandated the deployment of the ADS-B system for 2020 as part of next generation air transportation systems. Since AGM is part of the ATM procedure, its corresponding data is also collected by ADS-B. However, the average positioning accu-

This work was funded by the UK Engineering and Physical Sciences Research Council [grants EP/N029577/2, EP/N029496/2, and EP/N029356/1]. (*Corresponding author*: Jun Chen)

Xinwei Wang, Michal Weiszer and Jun Chen are with School of Engineering and Materials Science, Queen Mary University of London, UK (e-mail:xinwei.wang@qmul.ac.uk; m.weiszer@qmul.ac.uk; jun.chen@qmul.ac.uk)

John R. Woodward are with School of Electronic Engineering and Computer Science, Queen Mary University of London, UK (e-mail:j.woodward@qmul.ac.uk)

Alexander E.I. Brownlee is with Division of Computing Science and Mathematics, University of Stirling, UK (e-mail:sbr@cs.stir.ac.uk)

Mahdi Mahfouf is with Department of Automatic Control and Systems Engineering, University of Sheffield, UK (e-mail:m.mahfouf@sheffield.ac.uk)

racy of ADS-B is 33 metres according to flight test results [20], indicating same position errors exist in the collected AGM data. Furthermore, freely available ADS-B data on tracker websites often relies on low-end volunteer's receivers, further compromising data consistency and frequency. However, the freely available ADS-B data is often the only source for research and development. As far as we are aware, this issue has only been considered in [21]. In this work, a new ground movement mapping tool is developed, including airport layout generation and a process of correcting the original AGM raw data, in which each raw data point is matched to correct airport segment through a linear transformation and a brute force serching method. This kind of snapping process falls into the domain of map matching (MM).

MM algorithms use inputs collected from positioning technologies (such as GPS and ADS-B) and plot this data on a high resolution road map, aiming to output enhanced positioning results [22]. An example is illustrated in Figure 1, where the input in red dots is the GPS data with noise and the output is the black line aligned to the road segment after the MM procedure.

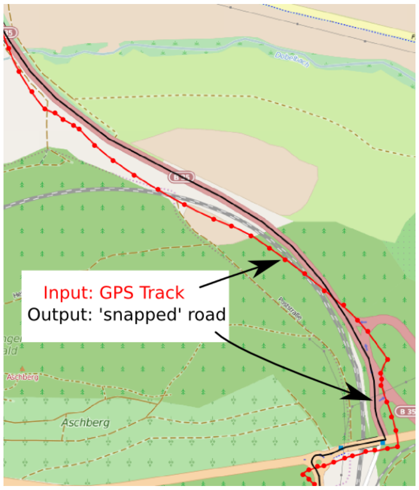


Fig. 1. An illustration of map matching [23].

Since the first application of the MM algorithm to car navigation [24], MM algorithms have been extensively studied, among which the fuzzy logic (FL) based MM algorithms demonstrate superior performance over many existing advanced MM algorithms [25], [26]. Besides, FL has advantages in providing a linguistic and an intuitive explanation via predefined IF-THEN rules. To better handle the uncertainties in the fuzzy logic system (FLS), the interval type-2 FLS (IT2FLS) has been further developed [27], and has been applied successfully to many different domains [28], [29]. However, the prospect of integrating IT2FLS into MM methods has not been hitherto investigated in the literature. Therefore, the purpose of this paper is twofold: the first is to address the necessity of the MM procedure for the AGM data from ADS-B, and the second is to demonstrate that the IT2FLS can be successfully applied in the context of the MM procedure for AGM, leading to improved accuracy.

The contributions of this paper are threefold: (1) To the best of our knowledge, MM for AGM has not been mathematically modelled and addressed before (except a preliminary research in [21]); we aim to formalise and clearly state the MM problem for AGM. (2) The IT2FLS is introduced for MM with the aim of better uncertainty handling. Specifically, we use the interval-valued fuzzy sets, which are a specific case of IT2FLS [30], [31]. The fuzzy rules with associated weights are designed and selected by preliminary experiments and sensitivity tests, and MF parameters are then tuned by using a particle swarm optimisation (PSO) algorithm. Moreover, an extra checking step with posterior information is integrated to further improve MM accuracy. (3) The proposed IT2FLS based MM framework is fine-tuned, and its superior performance in terms of MM accuracy is validated using real AGM data from ADS-B.

The remainder of this paper is structured as follows. In Section II, we provide an extensive literature review, including the MM algorithm and FLS. A brief problem description and the FLS modeling process are introduced in Section III. The developed IT2FL based MM algorithm for AGM is then discussed in Section IV. Section V contains a series of experimental results using the real world AGM data, which demonstrates the superior performance of the proposed MM algorithm. We conclude the work in Section VI.

II. Literature review

A. MM algorithms

MM algorithms have been intensively applied to road transport, and can be characterized into geometrical, topological, probabilistic and advanced MM algorithms [22]. Readers are referred to [22], [32] for more comprehensive MM surveys.

The first MM algorithm for car navigation was developed in [24], which is a geometrical algorithm on the basis of the road segment shape. Due to its simple structure, the algorithm is not accurate enough, especially at intersections and parallel roads. Velaga *et al.* [33] addressed several limitations in existing topological MM algorithms, and introduced an enhanced weight-based topological MM algorithms, in which two new weights for intersections and link connectivity were introduced. Experimental results demonstrate that the enhanced MM algorithm is superior to most existing topological MM algorithms.

Greenfeld [34] proposed a topologically based MM procedure for the low quality GPS data. A weighted score was computed and the match was determined with the highest score. White *et al.* [35] provided two simple geometrical and two topological MM algorithms for comparisons. Although the proposed algorithms are conventional, the authors shed light on future directions of MM algorithms. Chen and Bierlaire [36] proposed a probabilistic method based on a smartphone measurement model which can calculate the likelihood of the observed smartphone data in transport networks. The output of this probabilistic method is a set of candidate road segments corresponding to the highest probabilities.

Considering the low frequency positioning data, Quddus and Washington [37] developed a new weight-based shortest path and vehicle trajectory aided MM algorithm. The well-known $A^\star$ algorithm was employed for shortest path searching, and two additional weights relating to the shortest path and vehicle trajectory were defined. Simulation results indicate that the developed algorithm is suitable for real-time transport application and services. Sharath $et\ al.$ [38] developed a dynamic two-dimensional MM algorithm considering dynamic weight coefficients and road width. The road segment was expressed as a grids matrix according to its center line, and the location identification was based on a weight score with respect to proximity, kinematic, turn-intent prediction and connectivity.

Krüger $et\ al.$ [38] presented a visual interactive MM algorithm, including preprocessing and matching processes. The parameters of the preprocessing step were optimized with the immediate visual feedback. Large-scale taxi trajectory data was employed to demonstrate the performance of this approach. A closed-loop MM framework for ground vehicle navigation using ambient signals of opportunity was developed in [39]. The proposed framework assumes the vehicle has a priori knowledge of its initial states, and estimates subsequent vehicle states using a particle filter.

FL based MM algorithms have been adopted by the majority of existing advanced MM approaches. To the best of our knowledge, Kim and Kim [25] were the first to introduce FL into the domain of MM. They proposed a novel fuzzy based algorithm, which can identify the exact road segment on which a car locates.

Fu $et\ al.$ [40] considered candidate segments which connect to the current road segment and their projection points are within the current positioning error radius detected by GPS. If more than one candidate identified, each candidate segment is checked by a fuzzy system considering the distance and heading differences. However, details of detecting intersections were not described in this work. Based on distance and heading difference, Jagadeesh$et\ al.$ [41] designed a simple fuzzy inference system to calculate the likehood of each road in the candidate set. Syed and Cannon [42] robustly identified the first segment depending on the initialisation conditions of GPS, and then applied a FLS to check whether the current GPS point is in the previous segment. A confidence level was further defined for MM locations, while detailed explanations were not provided.

Quddus $et\ al.$ [26] further developed a highly accurate MM algorithm based on FLS. Different from previous FL based MM algorithms [25], [40], [41], [42], this work considered an augmented GPS positioning system with data from vehicle sensors. Therefore, more information including vehicle speed, direction of the vehicle, heading of the vehicle and its difference could be used as antecedents of the FLS. Overall, 23 fuzzy rules were applied to ensure a superior performance compared to other existing MM algorithms. For more studies of FL for MM, the reader is referred to [43], [44], [45].

Despite abundant research into MM for road transport, MM for the AGM has not been fully investigated yet. Zimmerman [46] designed a guidance framework for aircraft taxiing with position lights, in which a MM method was applied to continuously compare the segment with the stored taxiway network. However, the utilised MM algorithm has not been explained in detail. A geometrical MM approach with airport sensor measurements was proposed in [47]. The airport layout constraints were considered to conduct MM with multi-sensor environments. Note the authors focus more on the measurement noise filtering, rather than the MM algorithm.

To make use of freely available data relating to real-world airport ground movements, Brownlee $et\ al.$ [21] developed an end-to-end tool, including airport layout generation, flight track information exploration and the corresponding MM process. The raw ground movement data was snapped to the correct segments by applying a linear transformation, so that the taxiing speed and routes can therefore be further analyzed. Note that this tool involves a brute force method for MM, leading to worse performance in terms of accuracy and efficiency in some scenarios. In a nutshell, a high-precision MM algorithm for AGM has not been proposed yet and we aim to fill this gap by using FLS.

B. Fuzzy logic system

FLS was first introduced in 1965 [48], and has been employed to a broad range of applications from control [49], intelligence system [50], energy industry [51] and the MM [26]. FLS can provide an intuitive, transparent and interpretable representation of complex systems through IF-THEN rules. However, such FLS has limited capabilities to directly handle data uncertainties [28]. More efficiently, to overcome this shortcoming, the type-2 FL, which can better address the uncertainties, was further developed in [27].

Compared to the conventional FLS (refer to T1FLS thereafter), type-2 FLS (T2FLS) provides more degrees of freedom, which can improve the accuracy and generalisation capabilities of models [29]. However, before conducting the defuzzification process similar to T1FLS, a type reduction process is required for T2FLS [52]. Unfortunately, this type reduction process is typically time consuming, since it involves enumerating all T1FLS embedded in the T2FLS. To address this issue, a unique IT2FLS was designed [53], [54]. The well-known Karnik-Mendel method and its variations were developed to provide an iterative fast type reduction process [52], [55]. To avoid algorithm iterations and further speed up the type reduction process, a new mathematical interpretation of the Karnik-Mendel procedure was derived, and has been demonstrated to operate type reduction without iterations [28], [56].

Given such superior performance, IT2FLS has been widely applied in autonomous mobile robots [57], control system [58], pattern recognition [59] and medical applications [60], [61]. In this work, we make use of a specific interval-valued fuzzy sets, which are a unique case of IT2FLS, to conduct MM for AGM, since it strikes a balance between the degree of freedom and the computational efficiency [61]. Meanwhile, a careful design and selection of IF-THEN rules has been also conducted to ensure the performance of the MM algorithm.

III. Problem description and IT2FLS modeling

A. Problem description

To make use of historical real-world AGM data, the capability to identify the historical location of an aircraft on a correct segment is of great importance. This is achieved by the MM algorithm, where the following two parts are considered as the inputs: the positioning data of aircraft near airport, and the corresponding airport map with high spatial resolution. Since the ADS-B data is publicly available and widely used in AGM [21], [7], [8], we choose it as the source of aircraft original positioning information. As for the airport map, the ground movement tool developed in [21] is adopted for taxiway network generation (see Figure 2 as an example). Therefore, the MM algorithm takes inputs from the ADS-B system (or coordinates obtained from other sources) and the map data including topology, links and nodes information. The output of the MM process is the corresponding segment for each location point from ADS-B (or other data sources). Then the obtained ground movement segments can be further applied to the AGM routing and scheduling problems [4], [9].

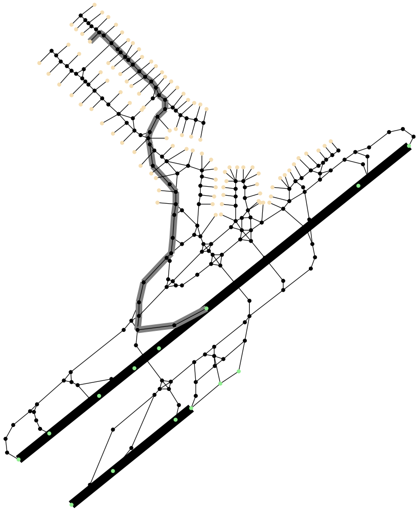


Fig. 2. Example of a generated airport taxiway network. Yellow, black, and green notes are gates, taxiways, and runways, taxiways. The ground movement route in gray is map matching result from the runway to the gate.

Identifying the initial node of ground movement is of great importance for the MM procedure. Fortunately, an aircraft typically starts or ends up at an airport gate during AGM. Therefore, only the first and last location data from ADS-B is checked for nearby gates using a search function, which will be explained in Section IV-A.

B. IT2FLS modeling

In this section, we first briefly introduce the T1FLS modeling process, and then extend it to IT2FLS for better addressing the MM procedure. FL is a superset of Boolean logic that represents the concept of partial truth [26]. As a conventional FLS, T1FLS consists of four parts: fuzzifier, inference, rule and defuzzifier.

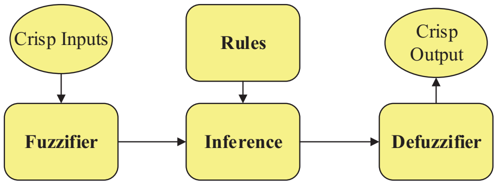


Fig. 3. The block diagram of the type-1 fuzzy logic system.

As shown in Figure 3, FLS establishes a mapping from an input space to an output space. In particular, the fuzzifier block takes a crisp input and generates a corresponding fuzzy mapping to capture more about linguistic uncertainties, namely antecedent membership functions (MFs). The inference part is specifically designed, where the fuzzified inputs are combined with rules, resulting in a fuzzified output, namely consequent MFs. Two types of inference block, Mamdani-type [62] and Sugeno-type [63], are mainly used for FLS. In this study, the Sugeno-type rules are selected for FLS construction. Clearly this fuzzified output cannot be directly used in practice, and a defuzzifier process is further required to generate a crisp output. Given an FLS with n inputs $\mathrm{x} \in \mathbb{R}^n$, c rules and fuzzified output $(y^1, y^2, \ldots, y^c) \in \mathbb{R}^c$, the ith rule can be expressed in the following IF-THEN linguistic form.

$$R^i : \text{IF x is } A^i, \text{THEN } y^i = MF^i(\mathrm{x}) \qquad (1)$$

where A^i is the antecedent MF of the ith rule and $MF^i(\mathrm{x})$ the consequent MF.

Although IF-THEN rules are transparent and interpretable, the subjectivity may be of great concern. This is because the words used in rules can mean different things to different people [28], and the definitions of MFs can vary from experts. To address this issue, T2FLS containing Type-2 Fuzzy MFs is introduced as an augmentation of the conventional T1FLS.

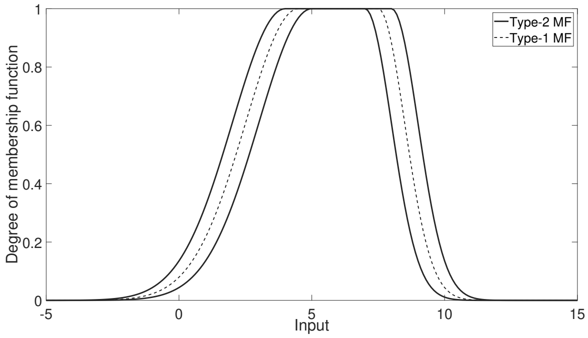


Fig. 4. Comparisons between type-1 and type-2 fuzzy membership functions.

As illustrated in Figure 4, comparing to type-1 MFs, type-2 MFs in T2FLS can provide an extra degree of freedom and typically improve the modeling accuracy. However, as shown in Figure 5, the disadvantage of using T2FLS comes from an additional time consuming type-reduction process before performing any defuzzification operation (see [52] for a detailed analysis). In light of this, a unique IT2FLS using interval type-2 fuzzy sets is designed to provide an iterative fast type reduction process [52], [55]. This process has been further

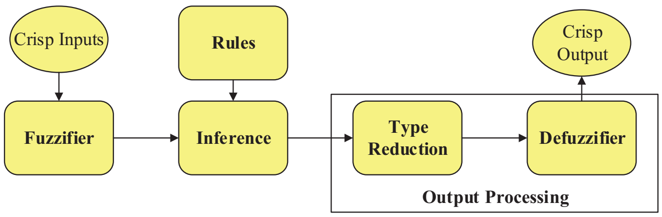


Fig. 5. The block diagram of the type-2 fuzzy logic system.

simplified through deriving a new mathematical interpretation of the Karnik-Mendel (KM) procedure without iterations [28], [56]. Detailed type reduction and defuzzification procedures are introduced as follows. We first compute centroids of a set of c fuzzy sets. The left and right end points y_l^i and y_l^i $(i = 1, \ldots, c)$ are computed using the KM algorithm. Notice that we only compute the end points once without KM iterations. Then we compute four boundary type-1 FLS centroids as

$$\begin{cases} y_l^{(0)}(\mathrm{x}) = \sum_{i=1}^{c} \underline{f}^i y_l^c \Big/ \sum_{i=1}^{c} \underline{f}^i, \quad y_r^{(0)}(\mathrm{x}) = \sum_{i=1}^{c} \overline{f}^i y_r^c \Big/ \sum_{i=1}^{c} \overline{f}^i \\ y_l^{(c)}(\mathrm{x}) = \sum_{i=1}^{c} \overline{f}^i y_l^c \Big/ \sum_{i=1}^{c} \overline{f}^i, \quad y_r^{(c)}(\mathrm{x}) = \sum_{i=1}^{c} \underline{f}^i y_r^c \Big/ \sum_{i=1}^{c} \underline{f}^i \end{cases} \tag{2}$$

where $\overline{f}^i$ and $\underline{f}^i$ are the upper and lower firing levels for the ith rule, respectively.

Based on the obtained four type-1 FLS centroids, we further compute four uncertainty bounds as

$$\begin{cases} \overline{y}_l(\mathrm{x}) = \min\{y_l^{(0)}(\mathrm{x}), y_l^{(c)}(\mathrm{x})\}, \quad \underline{y}_r(\mathrm{x}) = \max\{y_r^{(0)}(\mathrm{x}), y_r^{(c)}(\mathrm{x})\} \\ \underline{y}_l(\mathrm{x}) = \overline{y}_l(\mathrm{x}) - \left[\frac{\sum_{i=1}^{c}(\overline{f}^i - \underline{f}^i)}{\sum_{i=1}^{c}\overline{f}^i \sum_{i=1}^{c}\underline{f}^i} \cdot \frac{\sum_{i=1}^{c}\underline{f}^i(y_l^i - y_l^1)\sum_{i=1}^{c}\overline{f}^i(y_l^c - y_l^i)}{\sum_{i=1}^{c}\underline{f}^i(y_l^i - y_l^1) + \sum_{i=1}^{c}\overline{f}^i(y_l^c - y_l^i)} \right] \\ \underline{y}_r(\mathrm{x}) = \overline{y}_r(\mathrm{x}) - \left[\frac{\sum_{i=1}^{c}(\overline{f}^i - \underline{f}^i)}{\sum_{i=1}^{c}\overline{f}^i \sum_{i=1}^{c}\underline{f}^i} \cdot \frac{\sum_{i=1}^{c}\underline{f}^i(y_r^i - y_r^1)\sum_{i=1}^{c}\overline{f}^i(y_r^c - y_r^i)}{\sum_{i=1}^{c}\underline{f}^i(y_r^i - y_r^1) + \sum_{i=1}^{c}\overline{f}^i(y_r^c - y_r^i)} \right] \end{cases} \tag{3}$$

The actual lower and upper bounds of the output is now approximated as

$$[y_l(\mathrm{x}), y_r(\mathrm{x})] \approx [(\underline{y}_l(\mathrm{x}) + \overline{y}_l(\mathrm{x}))/2, (\underline{y}_r(\mathrm{x}) + \overline{y}_r(\mathrm{x}))/2] \tag{4}$$

The approximation can still maintain a high accuracy, which has been proven in [56]. The final defuzzified output is calculated as

$$y(\mathrm{x}) = (y_l(\mathrm{x}) + y_r(\mathrm{x}))/2 \tag{5}$$

More details and clarifications can be found in [56], [28].

IV. IT2FLS BASED MM ALGORITHM

The heart of MM algorithm is identifying the actual segment among the candidate segments for each location data point. In line with [26], we divide the identification process into two parts: the initial MM process and the subsequent MM process. Moreover, an extra checking step, which is similar to curve to curve analysis [34], is developed to fine-tune the proposed MM algorithm. Note this extra checking step is not capable of real-time MM applications, as it includes posterior information, i.e., the collected next movement positions of the aircraft.

A. Initial MM process

At the beginning of the MM process, we need to identify the initial segment that matches the first location data point. As mentioned in Section III-A, an aircraft normally departures or arrives at an airport gate, making the initial segment identification relatively easier compared to road transport. Thus we only need to check the first and last location points data from ADS-B: if the first point is validated to be closer to an airport gate segment, then the subsequent MM process is initiated for the remaining ADS-B data; otherwise the last point should be closer to a gate segment, and the subsequent MM process is evoked after reversing the order of the entire ADS-B data.

Algorithm 1 Initial MM process

```
Input: aircraft data from ADS-B, taxiway network information and radius of the search neighborhood;
Output: initial gate segment;
1: Select the first few location points data from ADS-B;
2: Search gate segments within circular neighborhood;
3: if exist candidate gate segments then
4:     return the gate segment with the smallest distance to the three location points;
5: else
6:     Reverse the aircraft data from ADS-B and go to line 1;
7: end if
```

The detailed initial MM process is provided in Algorithm 1. We first select the first few AGM points (e.g., the first three points) data from ADS-B, and search for the nearest airport gates. Here we define a search neighborhood within a circular radius (e.g., 100 metres) of the first AGM point, and then list the candidate gate segments within the circle. The gate is determined with the smallest distance to the first few location points. Notice if there is no candidate gate segments within the circle, we reverse the entire ADS-B data and conduct the gate search process again.

B. Subsequent MM process

Based on the matched initial gate segment, the subsequent MM process then iterates through location points from ADS-B in a chronological order to find their corresponding segments. Similar to the initial MM process, a search function for nearby segments is designed. Given the taxiway network normally consists of more than one thousand segments, iterating through all segments for each point would be time consuming. Instead, we set a circle around the current location and only search potential segments within or intersect with the circle. The number of candidate segments is further decreased by selecting the nearest k nodes to the current location point and the corresponding segments that connect to those nodes. This process is illustrated in Figure 6, where segments are highlighted using a circle of the current location point.

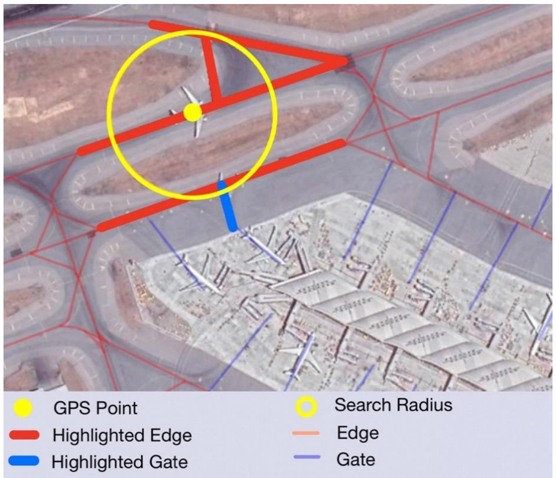


Fig. 6. Illustration of the initial MM process. The raduis of the circle is 100 metres.

Each highlighted segment is checked whether the current location point belongs to the segment, and a belongingness possibility is calculated using the developed IT2FLS. As the previous MM method for AGM [21] struggles with junctions and dense networks, we establish two separate IT2FLSs: one for the current segment, and the other for whether the current aircraft has left the previous segment and matches with another segment (potentially at a junction). The purpose of the first IT2FLS is to match the subsequent location along the current segment, unless the aircraft has passed the segment. The second IT2FLS is to identify a new segment among the candidates at a junction for the previous non-matched segment. After identifying the new segment by the second IT2FLS process, the first IT2FLS process restarts, matching the subsequent data points to the new segment.

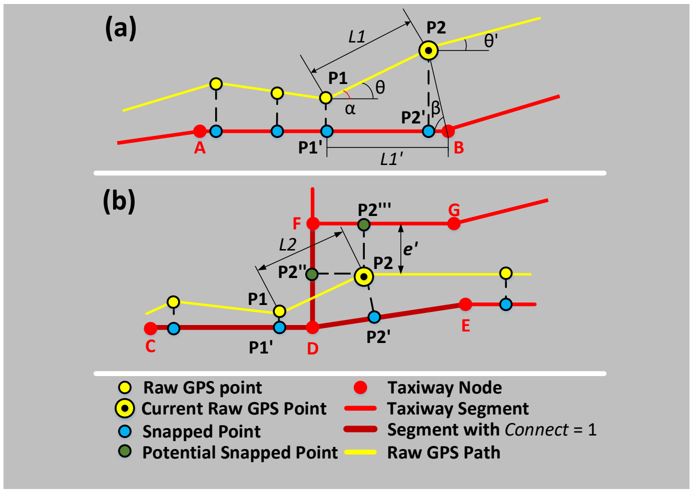


Fig. 7. Illustration of input parameters of the subsequent MM process. (a) along a segment and (b) at a junction.

1) *Inputs of the FLS:*

As illustrated in Figure 7, the related input parameters for the two IT2FLSs are introduced in this section. According to Figure 7(a), the inputs for subsequent MM process along a segment (the first IT2FLS) are (1) the angle α, which is between the previous trajectory and the current segment, (2) the angle β, which is between the current segment and the segment from the current raw GPS location point to the end node of the current segment, (3) the difference δ which is between the distance $L1'$ and the distance $L1$. Here $L1'$ is calculated from the previous matched point to the end of the segment, and $L1$ is from the previous raw GPS point to the current point, and (4) the heading angle increment $H = |\theta' - \theta|$, where $|\cdot|$ is the absolute value function. The heading is calculated by using the previous, current and consequent raw GPS location points. The distances between two latitude-longitude location points can be calculated by the Haversine method [64]:

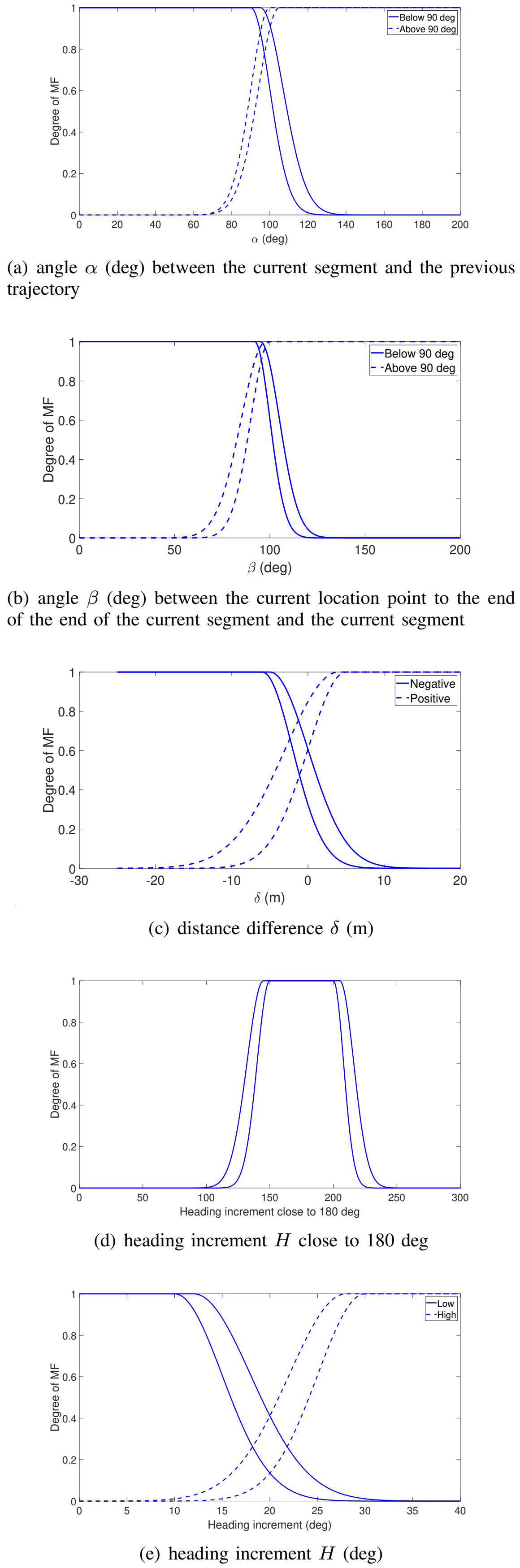
$$d = 2R\sin^{-1}\sqrt{\sin^2(\frac{\varphi_2 - \varphi_1}{2} + \cos\varphi_1 \cos\varphi_2 \sin^2(\frac{\lambda_1 - \lambda_2}{2})} \tag{6}$$

where d, R, $(\varphi_i, \lambda_i)(i = 1, 2)$ are the distance from one point to the subsequent one, the radius of the Earth, the latitudes and longitudes of the raw GPS location points, respectively.

As for the subsequent MM process at a junction (the second IT2FLS), the input variables are defined in Figure 7 (b) as (1) the connectivity of the segments $Connect$, i.e., $Connect = 1$ when segments directly connect to the previous segment node; otherwise $Connect = 0$, (2) the distance error Δ, which is the difference between the shortest path distance from the previous matched point to the candidate matched point and the distance between the previous and the current raw GPS points, and (3) the Haversine distance e, which represents the distance from the current raw GPS location point to the candidate segment. These inputs are calculated for every nearby segments given each raw GPS point. Using the segment FG in Figure 7 (b) as an example, its connectivity $Connect = 0$ since it does not directly connect to the current segment CD. Its distance error $\Delta = |L2 - L2'|$, where $L2' = P1'D + DF + FP2'''$, and its Haversine distance is e'.

2) *IT2FLS along a segment:*

After clearly defining the input parameters, we design the IT2FLS along a segment (FLS-1) as follows. The inputs are fuzzified with the use of S-/bell-shaped and gaussian MFs for the inference block. These MFs are shown in Figure 8, resulting in a value between 0 and 1. Note the specific parameters of these MFs are determined by a training process, which will be described in the next section. Initially the rules using T1FLS in [26] are employed for the FLS-1, however, undesirable results with a lower MM accuracy are obtained. This is because the rules in [26] rely heavily on gyro-rate reading as well as dilution of precision information, both of which are not provided from the original ADS-B data. Therefore, preliminary experiments and sensitivity tests similar to that in [26] were conducted to first design fuzzy rules, determine the associated rule weights, and the Sugeno fuzzy model output. The generated nine rules for the FLS-1 are shown below. Note that the MF parameters are subject to further optimisation using a PSO algorithm. The above choice is largely due to the fact that optimising MF parameters using PSO is already time-consuming (up to one week for 100 PSO iterations using a high-performance



(a) angle α (deg) between the current segment and the previous trajectory

(b) angle β (deg) between the current location point to the end of the end of the current segment and the current segment

(c) distance difference δ (m)

(d) heading increment H close to 180 deg

(e) heading increment H (deg)

Fig. 8. MFs of FLS-1 along a segment.

computing cluster).

- Rule 1: If (α is below 90°) and (β is below 90°), $Out1$ = Z3 and W = 0.3
- Rule 2: If (δ is positive) and (α is above 90°), $Out1$ = Z1 and W = 0.3
- Rule 3: If (δ is positive) and (β is above 90°), $Out1$ = Z1 and W = 0.3
- Rule 4: If (H is low) and (α is below 90°) and (β is below 90°), $Out1$ = Z3 and W = 1
- Rule 5: If (H is low) and (δ is positive) and (α is above 90°), $Out1$ = Z1 and W = 1
- Rule 6: If (H is low) and (δ is positive) and (β is above 90°), $Out1$ = Z1 and W = 1
- Rule 7: If (α is below 90°) and (β is below 90°), $Out1$ = Z3 and W = 0.3
- Rule 8: If (H is high) and (α is below 90°), $Out1$ = Z1 and W = 1
- Rule 9: If (H is close to 180°) and (α is below 90°), $Out1$ = Z3 and W = 1

where we employ a zero-order Sugeno fuzzy model with three constant output $Out1$: Z1=10 (low), Z2 = 50 (average) and Z3 = 100 (high). Here W is the rule weight importance.

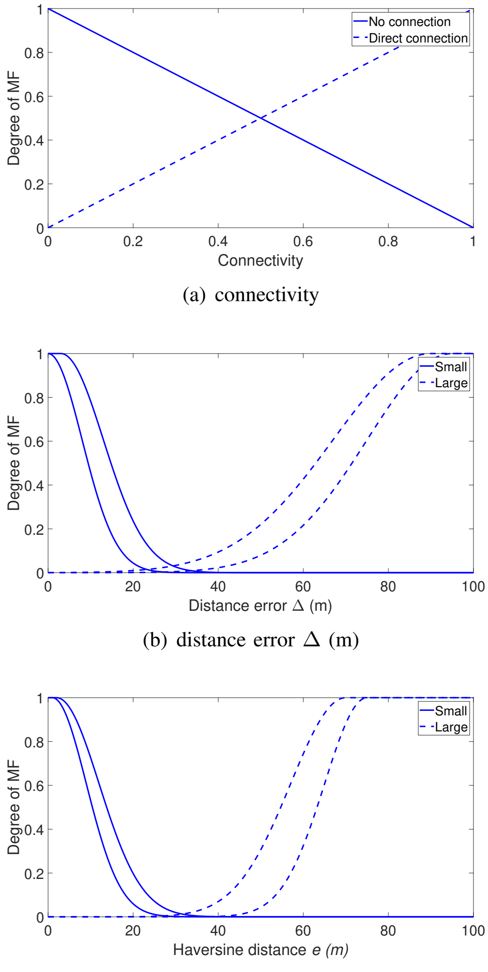
The value of $Out1$ determines whether the current location point matches with the current segment. If $Out1$ is greater than a certain threshold value, then the location point snaps to current segment (for instance, P2 would snap to the same segment P1 matches in Figure 7(a)). In contrast, if $Out1$ is no more than the threshold, it indicates that the current location point has passed current segment. The FLS-1 terminates and the IT2FLS at a junction is then evoked to identify the new segment.

3) IT2FLS at a junction: IT2FLS at a junction (FLS-2) applies a simpler system consisting of three inputs and six fuzzy rules. The MFs of the inputs are plotted in Figure 9, and the generated fuzzy rules are as follows. Note the specific parameters of these MFs are determined after a training process in the next section, and the rules are generated using the same sensitivity test in the above subsection.

- Rule 1: If ($Connect$ is low), $Out2$ = Z1 and W = 0.5
- Rule 2: If ($Connect$ is high), $Out2$ = Z3 and W = 0.5
- Rule 3: If (Δ is small), $Out2$ = Z3 and W = 0.1
- Rule 4: If (Δ is large), $Out2$ = Z1 and W = 0.1
- Rule 5: If (e is small), $Out2$ = Z3 and W = 1
- Rule 6: If (e is large), $Out2$ = Z1 and W = 1

where $Out2$ is the possibility belonging to the new segment. Thus the candidate segment with the largest crisp output value after type reduction and defuzzification will be selected for matching. According to preliminary experiments, the distance error Δ from the ADS-B data is relatively large, leading to incorrect matching results. Meanwhile, the input $Connect$ aggravates the errors in the condition that the AGM data has large intervals between raw GPS location points, as the corresponding rule prefers to select a closer segment to the current one. These issues have been addressed through increasing the weights of the Haversine distance rules (Rules 5 and 6 in the FLS-2) during the sensitivity test, and the performance of the FLS-2 is thus ensured.

However, assigning a higher weight to the Haversine distance e could lead to map matching errors, e.g., a location point could be matched to a wrong segment when the point is near a junction. We address this issue by introducing an extra step for FLS-2, after identifying a new segment for the current location point. Similarly to the curve-to-curve analysis [34], we apply the FLS-2 on the two next nodes, and calculate the



(a) connectivity

(b) distance error Δ (m)

(c) Haversine distance e (m)

Fig. 9. MFs of FLS-2 at a junction.

Dijkstra shortest path [65] from the previous point to the next two points, as shown in Figure 10.

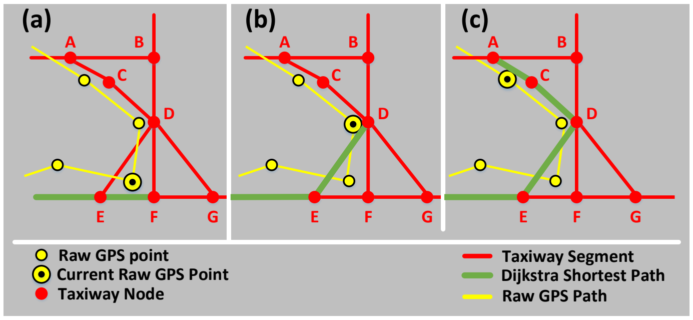


Fig. 10. FLS-2 chooses the most likely segments for the current, next and second next points, using a shortest path to connect the segments of the next two points. (a) the current point does not match to the correct segment, (b) the next (second) point correctly matches the segment, (c) the third point correctly matches the segment including the second point, therefore the current point re-matches the correct segment after the extra check step.

Three shortest paths corresponding to the three subfigures in Figure 10 are considered as follows: whether the first path is included in the second path. If yes, keep the current map matching result; otherwise, check whether the first path (Figure 10(a)) and the second path (Figure 10(b)) are included in the third one (Figure 10(c)), respectively. If the first path matches the third one or both paths are not included in the third one, keep the current map matching result; if the second path matches the third one (corresponding to the situation in Figure 10), update the current point map matching results with the second path. We could consider more subsequent points in this check step. However, sparser data may yield much more errors, since the reliance on the Dijkstra shortest path could cause undesired shortcuts.

V. Experiments

To validate the performance of the proposed MM algorithm, we first briefly introduce the collected AGM dataset as well as the experimental setup, and then the proposed MM algorithm is validated and compared with existing methods.

A. Dataset and experimental setup

In this work, we utilise the data from Hong Kong Airport (HKG), one of the top ten busiest airports in the world. The layout of HKG is illustrated in Figure 11, where the red, blue and green segments represent taxiways, gates and runways, respectively.

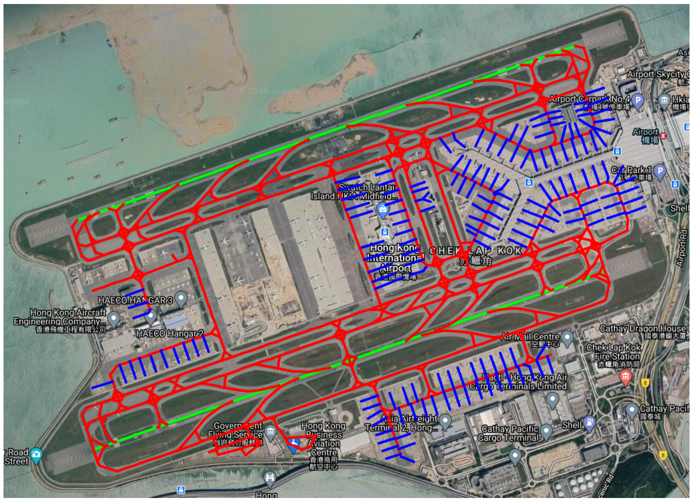

Hong Kong International Airport
CHEK LAP KOK
HAECO HANGAR 3
HAECO Hangar 2
Hong Kong Aircraft Engineering Company
Airport Skycity
Cathay Dragon House
Cathay Pacific Cargo Terminal
Government Flying Service
Hong Kong Business Aviation Centre
Cathay Pacific
Shell

Fig. 11. The layout of Hong Kong Airport.

The real-world ADS-B aircraft ground movement data at HKG is collected from a freely-available website FlightRadar24, following the tools described in [66] (available at https://github.com/gm-tools/gm-tools). The website FlightRadar24 gathers airborne flight tracks from automatic dependent ADS-B messages transmitted by aircraft, including the latitude, longitude and altitude information with 10-metre resolution. The sampling frequency is from 5 to 10 seconds. As we focus on the aircraft ground movements at HKG, only the tracks within 5km of HKG are collected. Notice that the airport towing vehicle tracks are also included in the original ADS-B messages, and we have removed this irrelevant data when collecting the aircraft movements. In the end, we collected

130 aircraft ground movements from 15th January 2017, and manually identified the corresponding actual movement trajectories. Overall, the collected aircraft movements contain 11,119 nodes to be matched. The identified actual movement trajectories (in total 7,956 segments) are to be used for testing the performance of the proposed MM algorithm (denoted as IT2FLS+ thereafter). We also compare IT2FLS+ with other five MM algorithms, including the exhaustive method (denoted as Exhaustive) which applies a brute method to match a route [21], the IT2FLS model without the extra checking step for FLS-2 in Section IV-B3 (IT2FLS), the T1FLS model with/without extra checking step for FLS-2 (T1FLS+/T1FLS), and the FLS (denoted as Q-FLS) developed in [26]. The Q-FLS method is modified to exclude the original rules using gyro-rate reading and dilution of precision, as the collected aircraft movements dataset does not contain such information. Note that only the MM methods without the extra checking step are capable of real-time applications, since the extra checking step includes posterior information, i.e., the collected next movement positions of the aircraft.

By randomly dividing the 130 aircraft ground movements into 10 groups, all the MM models (except the exhaustive method) are further trained and tested using particle swarm optimization [67] based on 10-fold cross validation. During the training process, we aim to optimise the MF parameters with maximal average map matching accuracy AccA, which is defined in the next section. The MM algorithms are implemented in Python, and run on Queen Mary's Apocrita high-performance computing facility.

B. Results and discussions

The overall comparison results of the MM algorithms are illustrated in Figure 12, including MM accuracy for training and testing dataset results (denoted as Train and Test) respectively. Detailed MM results for each training/testing subset are shown in the appendix from Tables II to VI. Moreover, the MM accuracy results with initial parameters setting without training (denoted as Pre-Train) are included. Here, four metrics are used to measure accuracy over different MM algorithms. AccD, AccE and AccT denote MM accuracy over the entire distance, the number of segments, and the number of turning segments that are correctly identified respectively; AccA represents average MM accuracy based on AccD, AccE and AccT. For instance, assume that one actual aircraft movement trajectory has in total 100 segments with 20 turning segments, and its distance is 5km. Taking the corresponding ADS-B aircraft movement data as inputs, the MM method generates a snapped trajectory, in which 95 out of 100 segments, 18 of 20 turning segments (which have more than two directly connected segments), and 4.8 of 5km are correctly mapped. Then AccD, AccE and AccT of the MM process are calculated as $95/100 = 95\%, 18/20 = 90\%, 4.8/5 = 96\%$. Consequently, average accuracy AccA is $(95\% + 90\% + 97\%)/3 = 93.7\%$. The MM results from Q-FLS are separately shown in Figure 12(a), as its MM accuracy never exceeds 80%, which is clearly worse than other MM algorithms. Note that there are no MM accuracy results for T1FLS+ and IT2FLS+ over Pre-Train and Train in Figure 12(b) and Figure 12(c), as these two algorithms directly apply the same rules and parameters from T1FLS/IT2FLS, and have an additional extra checking step as described in Section IV-B3..

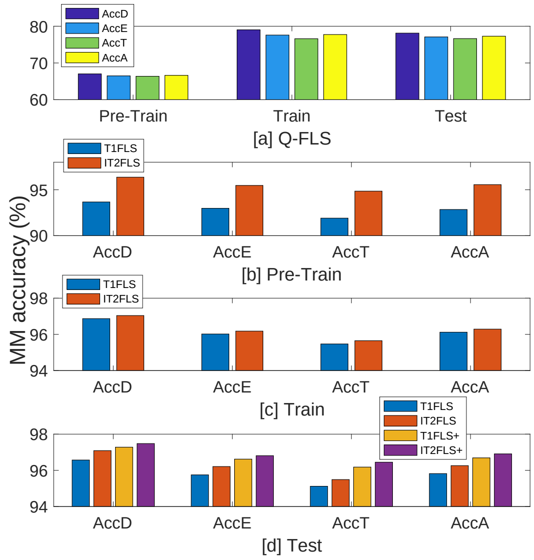


Fig. 12. Comparison results of MM algorithms.

As shown in Figure 12(b), IT2FLS achieves a higher MM accuracy over 95% without training. This could be due to the generalisation capability of the IT2FLS model, which employs more parameters to handle the MM process; the proposed IT2FLS method is then insensitive to the MF parameters to some extent. T1FLS also performs well with specifically designed rules. In contrast, Q-FLS initially has the worst MM accuracy, as the applied rules may not effectively for MM. Although the MM accuracy of Q-FLS increases over 10% after training process, Q-FLS still has the lowest MM accuracy.

After the training process, IT2FLS+ clearly achieves the best MM performance in terms of all four metrics. The overall MM accuracy AccA goes up to 96.91%. When excluding the extra checking step, the IT2FLS+ model degrades to IT2FLS, and its overall accuracy AccA decreases to 96.26% as well. We also observe that T1FLS and T1FLS+ both perform worse than corresponding IT2FLS and IT2FLS+, respectively.

The MM accuracy improvement in terms of AccD and AccE using IT2FLS+ over T1FLS+ appears relatively minor. This is because of the newly developed set of fuzzy rules, which are specifically designed for the airport map matching problem. Based on these fuzzy rules, T1FLS+ already achieves a high MM accuracy. Therefore, the room for IT2FLS+ to massively increase the accuracy with respect to these two metrics is limited. However, IT2FLS+ still renders advantages over T1FLS+ in the following two aspects. Firstly, IT2FLS+ offers more generalisation capacity due to IT2FLS, indicating it has the potential to better handle unseen airport ground movements. Secondly, the seemingly minor accuracy improvement with respect to AccE and AccD using IT2FLS+ can result in a significant increase in the number of correctly matched segments compared to T1FLS+. For instance, by using IT2FLS+, the AccE increases by $96.81\% - 96.62\% = 0.19\%$. After translating this improvement into the increased number of correctly matched segments using the collected ground movement data in [8], for Hong Kong Airport, which has 33,095 movements in 36 days, the significance of such improvement becomes evident. As each movement includes 60 segments on average, this indicates the number of correctly matched segments could be significantly increased by $33095 \times 60 \times 0.19\% = 3772$ when using IT2FLS+ compared to T1FLS+.

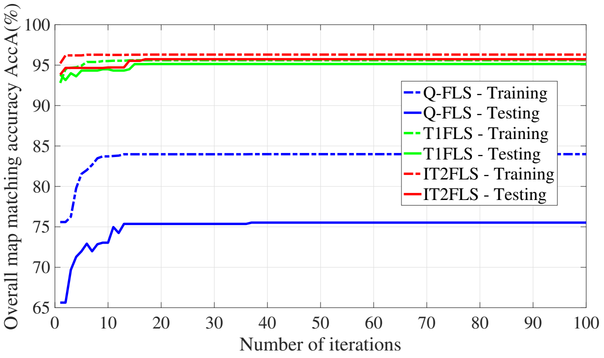


Fig. 13. Illustration of the training and testing curves for Q-FLS, T1FLS and IT2FLS.

Q-FLS has the worst performance with less than 80% MM accuracy. This is reasonable, as its original fuzzy logic rules are designed for road transport, and only part of them can be applied in the aircraft movement MM process. The performance gap can be also found during the training process. Taking the 7th subgroup in the dataset as the testing set, we illustrate its training and testing curves of Q-FLS and T1FLS in Figure 13. At the beginning of the training, the performance difference of Q-FLS between the training and testing data is about 10%, and its MM prediction is less than 85%. As for T1FLS and IT2FLS, the MM accuracy difference between the training and testing data slightly fluctuates within 1%, and has a higher MM accuracy at early training stages. These observations demonstrate that the specifically designed rules in this work have superior performances even without optimising the parameters of the MFs.

As the MM Exhaustive method taken from [21] does not belong to data-driven models, the results are separately presented here: for the whole 130 flight trajectories, among 8 of which fail to match a route. For the remaining trajectories, the MM accuracy of AccD, AccE and AccT is 91.64%, 90.87% and 89.73% respectively, which is lower than that of our proposed MM approach.

To further analyse the MM results, we look into the performance differences with respect to metrics AccD, AccE and AccT. The MM accuracy over the entire distance AccD consistently has the highest value across all MM algorithms, followed by AccE and AccT. This can be explained by one MM example illustrated in Figure 14 using IT2FLS+. Clearly the snapped segments in cyan successfully match almost the entire aircraft movement trajectory, except one turning segment within the black circle. Since certain input variables of MM model have drastic changes (e.g, the heading angle could vary up to 90 degrees for turning segments) and there are more optional segments to match around the turning-segment areas, the turning segments are indeed difficult to match. Thus AccT has the lowest MM accuracy, while AccE considering all the segments is greater than AccT. Meanwhile, the turning segments typically are shorter than other segments, therefore the AccD, which accounts MM accuracy over the movement distance, is larger than AccE. Nevertheless, the MM accuracy AccT of IT2FLS+ still achieves a high-precision level with 96.45%, correctly matching most of the turning segments.

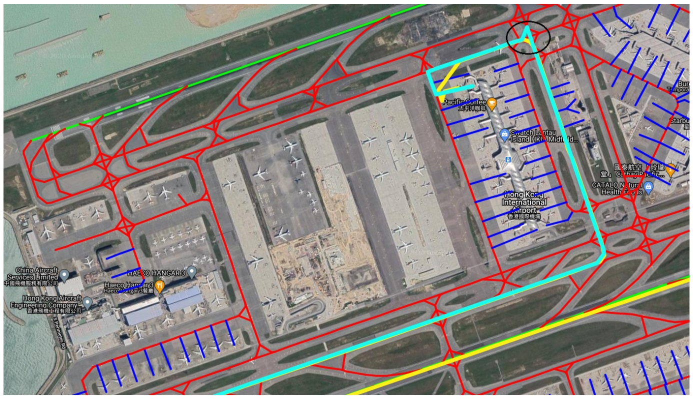


Fig. 14. An example of MM results using IT2FLS+. Yellow and cyan lines represent the collected aircraft movement to be mapped and corresponding snapped trajectories, respectively. The MM results in cyan correctly match the majority of the aircraft movement trajectory, except an incorrect turning segment within the black circle.

TABLE I
MM COMPUTATIONAL TIME. THE THREE ROWS INDICATE THE RUN TIMES (IN SECONDS) FOR THE ENTIRE DATASET, EACH TRAJECTORY AND SINGLE POINT IN AVERAGE, RESPECTIVELY.

Method	Exhaustive	Q-FLS	T1FLS	T1FLS+	IT2FLS	IT2FLS+
Total	322,030	311.84	512.40	1037.80	564.07	1095.07
Trajectory	2,477	2.40	3.94	7.98	4.34	8.42
Point	31	0.03	0.05	0.09	0.05	0.10

The computational time for the six MM algorithms are listed in Table I. Note that runs with the exhaustive approach [21] were conducted on a separate infrastructure[1]; nevertheless, it clearly has the largest computational time. Q-FLS takes the least run time, as it runs with the least fuzzy logic rules and does not consider the extra checking step. Recall that an simplified type reduction process, which can significantly increase the computational efficiency, has been introduced when implementing the IT2FLS model [28]. Therefore, the run time of IT2FLS only slightly increases compared to that of IT1FLS. Integrating with the extra checking step, T1FLS/IT2FLS is upgraded to T1FLS+/IT2FLS+, while the run time is doubled. However, the time increase is acceptable, as the average run time for single point along a movement trajectory is no more than 0.10 seconds. The MM algorithms with a real-time manner are promising to underpin online air traffic control

[1] the Archie-WeSt HPC; where each core is a Intel Xeon Gold 6138 CPU @2.0GHz. While direct comparisons are thus not possible, the FLS approaches compare very favourably.

for tactical decisions, e.g., AGM conflict resolution, trajectory prediction and automated lighting control.

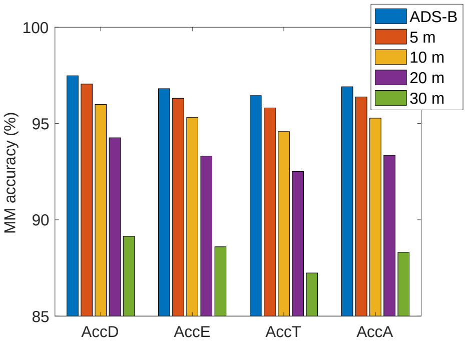


Fig. 15. MM robustness performance of IT2FLS+.

To test the robustness of the proposed MM algorithm IT2FLS+, we further add extra positioning errors in the real-world raw trajectory data collected from ADS-B. Specifically, five different positioning error conditions are considered. The baseline condition uses the real-world ADS-B data (denoted as ADS-B). Four additional conditions include extra positioning errors randomly distributed from 0 to Δ_e (Δ_e = 5, 10, 20 and 30) metres, which are respectively added to each raw point data. The MM results are illustrated in Figure 15 in terms of the four accuracy indicators. Clearly, the MM performance of IT2FLS+ has a decreasing trend along with increasing positioning errors. However, the decrease is reasonably slow. For instance, average MM accuracy AccA maintains a high value of 93.35% with additional 20-metre errors. Although a sharp decrease is observed between 20 to 30 metres errors, the MM results still have a good performance with AccA at 88.31%. The proposed MM algorithm IT2FLS+ is verified to be robust thanks to its type-2 fuzzy rules and the extra checking step.

VI. Conclusions

A new interval type-2 fuzzy logic based map matching algorithm for airport ground movement has been proposed in this work. We specifically design IF-THEN rules for the interval type-2 fuzzy logic system (IT2FLS), which employs more parameters to better address the map matching problem. Moreover, an extra checking step with posterior information is integrated into the IT2FLS to further improve the map matching accuracy. The proposed algorithm is tested with real-world aircraft movement data at Hong Kong Airport. Experimental results indicate that both of the IT2FLS and the designed additional checking step lead to better map matching performance, while the computation is relatively cheap and capable of online applications. The proposed map matching algorithm can be of great value to underpin trajectory-based airport ground operations [4].

Future research for map matching algorithms for airport ground movement can be oriented in three directions. 1) The fuzzy rules are first designed with associated rule weights and output values, followed by an efficient global optimisation method to simultaneously design fuzzy rules and their associated parameters. 2) Current experiments are conducted at a single airport, due to the time-consuming process of dataset labelling, i.e., identifying the actual aircraft movement trajectories. It would be of great importance to establish a benchmark for map matching testing across different airports. 3) The proposed algorithm is applied for matching airport ground trajectories. It could also provide quantitative insights into other road transport map matching problems.

Appendix

Detailed MM results for each training/testing subset are shown from Tables II to VI.

TABLE II
MM results with Q-FLS.

	Train			
	AccD (%)	AccE (%)	AccT (%)	AccA (%)
1	80.52	78.92	78.19	79.21
2	73.91	74.08	72.03	73.34
3	76.69	76.14	73.78	75.54
4	85.64	83.21	82.02	83.62
5	78.86	77.00	76.46	77.44
6	81.18	78.83	78.86	79.62
7	85.85	83.76	82.36	83.99
8	78.59	76.56	76.72	77.29
9	79.27	77.42	75.88	77.53
10	69.76	70.10	69.70	69.85
mean	79.03	77.60	76.60	77.74
std	4.89	4.01	4.05	4.31
	Test			
	AccD (%)	AccE (%)	AccT (%)	AccA (%)
1	85.11	82.77	82.80	83.56
2	71.79	72.22	70.27	71.43
3	71.30	72.24	69.85	71.13
4	80.67	79.80	79.49	79.99
5	78.98	76.82	75.44	77.08
6	81.66	81.06	80.67	81.13
7	76.12	74.33	76.12	75.52
8	78.90	76.35	76.90	77.38
9	81.66	80.70	78.94	80.43
10	75.12	74.65	75.89	75.22
mean	78.13	77.09	76.64	77.29
std	4.49	3.80	4.17	4.15

TABLE III
MM results with T1FLS+.

	Test			
	AccD (%)	AccE (%)	AccT (%)	AccA (%)
1	97.38	96.04	95.33	96.25
2	95.42	95.34	95.68	95.48
3	98.52	98.26	97.73	98.17
4	95.93	95.07	94.27	95.09
5	98.26	98.08	97.69	98.01
6	97.52	96.62	95.85	96.66
7	97.40	96.24	95.92	96.52
8	96.36	96.01	96.03	96.13
9	98.22	97.32	96.64	97.39
10	97.75	97.22	96.64	97.20
mean	97.28	96.62	96.18	96.69
std	1.04	1.08	1.05	1.06

Acknowledgement

This work was funded by the UK Engineering and Physical Sciences Research Council [grants EP/N029577/2,

TABLE IV
MM RESULTS WITH IT2FLS+.

	Test			
	AccD (%)	AccE (%)	AccT (%)	AccA (%)
1	97.46	96.27	95.63	96.45
2	96.43	96.35	96.68	96.49
3	98.56	98.26	97.73	98.18
4	97.65	96.92	96.50	97.03
5	98.26	98.08	97.69	98.01
6	97.01	95.93	95.53	96.16
7	97.01	95.93	95.53	96.16
8	96.17	95.16	94.95	95.43
9	98.44	97.45	96.81	97.57
10	97.75	97.22	96.64	97.20
mean	97.48	96.81	96.45	96.91
std	0.87	1.05	0.96	0.96

TABLE V
MM RESULTS WITH T1FLS.

	Train			
	AccD (%)	AccE (%)	AccT (%)	AccA (%)
1	96.47	95.55	94.85	95.62
2	97.71	96.85	96.35	96.97
3	96.22	95.42	94.73	95.46
4	96.45	95.71	95.06	95.74
5	96.77	95.94	95.41	96.04
6	96.73	95.86	95.29	95.96
7	96.41	95.50	94.90	95.60
8	97.34	96.44	95.94	96.57
9	97.33	96.52	96.11	96.65
10	97.27	96.42	96.01	96.57
mean	96.87	96.02	95.47	96.12
std	0.51	0.50	0.59	0.53
	Test			
	AccD (%)	AccE (%)	AccT (%)	AccA (%)
1	96.79	95.46	94.31	95.52
2	94.78	94.05	94.51	94.45
3	96.72	97.09	96.21	96.67
4	96.23	95.32	94.59	95.38
5	97.73	97.32	96.74	97.26
6	97.17	96.11	95.20	96.16
7	96.06	94.99	94.36	95.14
8	95.63	94.30	93.86	94.60
9	97.45	96.11	95.39	96.32
10	97.19	96.75	96.05	96.66
mean	96.57	95.75	95.12	95.82
std	0.91	1.12	0.96	1.00

TABLE VI
MM RESULTS WITH IT2FLS.

	Train			
	AccD (%)	AccE (%)	AccT (%)	AccA (%)
1	96.55	95.69	95.02	95.75
2	97.56	96.75	96.25	96.85
3	96.64	95.78	95.20	95.88
4	96.74	95.91	95.34	96.00
5	96.93	96.04	95.54	96.17
6	96.63	95.69	95.12	95.81
7	97.07	96.20	95.64	96.30
8	97.33	96.44	95.94	96.57
9	97.37	96.52	96.11	96.67
10	97.56	96.74	96.33	96.88
mean	97.04	96.18	95.65	96.29
std	0.40	0.42	0.48	0.43
	Test			
	AccD (%)	AccE (%)	AccT (%)	AccA (%)
1	96.79	95.46	94.31	95.52
2	97.80	97.16	96.68	97.21
3	98.15	98.03	97.42	97.87
4	97.61	96.92	96.18	96.90
5	97.73	97.33	96.74	97.27
6	96.81	95.80	94.81	95.81
7	95.64	94.37	93.59	94.53
8	95.64	94.30	93.86	94.60
9	97.45	96.11	95.40	96.32
10	97.28	96.60	95.85	96.58
mean	97.09	96.21	95.49	96.26
std	0.87	1.24	1.31	1.14

EP/N029496/2, and EP/N029356/1]. Thanks are also extended to undergraduate student Chris Claudet for his assistance in this work.

References

[1] EUROCONTROL, “European aviation in 2040: Challenges of growth,” 2018, available at https://www.eurocontrol.int/publication/challenges-growth-2018.

[2] A. I. Czerny, X. Fu, Z. Lei, and T. H. Oum, “Post pandemic aviation market recovery: Experience and lessons from china,” *Journal of Air Transport Management*, vol. 90, p. 101971, 2021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0969699720305548

[3] ATI, “Aerospace Technology Institute - accelerating ambition - technology strategy,” 2019, https://www.ati.org.uk/wp-content/uploads/2021/08/ati-technology-strategy.pdf.

[4] J. Chen, M. Weiszer, P. Stewart, and M. Shabani, “Toward a more realistic, cost-effective, and greener ground movement through active routing – Part I: Optimal speed profile generation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1196–1209, 2015.

[5] J. Chen, M. Weiszer, G. Locatelli, S. Ravizza, J. A. Atkin, P. Stewart, and E. K. Burke, “Toward a more realistic, cost-effective, and greener ground movement through active routing: A multiobjective shortest path approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3524–3540, 2016.

[6] C. Evertse and H. Visser, “Real-time airport surface movement planning: Minimizing aircraft emissions,” *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 224–241, 2017.

[7] A. E. Brownlee, M. Weiszer, J. Chen, S. Ravizza, J. R. Woodward, and E. K. Burke, “A fuzzy approach to addressing uncertainty in airport ground movement optimisation,” *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 150–175, 2018.

[8] X. Wang, A. E. Brownlee, J. R. Woodward, M. Weiszer, M. Mahfouf, and J. Chen, “Aircraft taxi time prediction: Feature importance and their implications,” *Transportation Research Part C: Emerging Technologies*, vol. 124, p. 102892, 2021.

[9] X. Wang, A. E. Brownlee, M. Weiszer, J. R. Woodward, M. Mahfouf, and J. Chen, “A chance-constrained programming model for airport ground movement optimisation with taxi time uncertainties,” *Transportation Research Part C: Emerging Technologies*, vol. 132, p. 103382, 2021.

[10] J. A. Atkin, E. K. Burke, and S. Ravizza, “The airport ground movement problem: Past and current research and future directions,” in *Proceedings of the 4th International Conference on Research in Air Transportation (ICRAT), Budapest, Hungary*, 2010, pp. 131–138.

[11] G. Hancerliogullari, G. Rabadi, A. H. Al-Salem, and M. Kharbeche, “Greedy algorithms and metaheuristics for a multiple runway combined arrival-departure aircraft sequencing problem,” *Journal of Air Transport Management*, vol. 32, pp. 39–48, 2013.

[12] U. Benlic, A. E. Brownlee, and E. K. Burke, “Heuristic search for the coupled runway sequencing and taxiway routing problem,” *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 333–355, 2016.

[13] G. De Maere, J. A. Atkin, and E. K. Burke, “Pruning rules for optimal runway sequencing,” *Transportation Science*, vol. 52, no. 4, pp. 898–916, 2017.

[14] G.-S. Jo, J.-J. Jung, and C.-Y. Yang, “Expert system for scheduling in an airline gate allocation,” *Expert systems with applications*, vol. 13, no. 4, pp. 275–282, 1997.

[15] J. Guépet, R. Acuna-Agost, O. Briant, and J.-P. Gayon, “Exact and heuristic approaches to the airport stand allocation problem,” *European Journal of Operational Research*, vol. 246, no. 2, pp. 597–608, 2015.

[16] E. D. Ganev, “Electric drives for electric green taxiing systems: Ex-

amining and evaluating the electric drive system," *IEEE Electrification Magazine*, vol. 5, no. 4, pp. 10–24, 2017.

[17] J. Koeners and R. Rademaker, "Creating a simulation environment to analyze benefits of real-time taxi flow optimization using actual data," in *AIAA Modeling and Simulation Technologies Conference*, 2011, p. 6372.

[18] S. Ravizza, J. Chen, J. A. Atkin, P. Stewart, and E. K. Burke, "Aircraft taxi time prediction: comparisons and insights," *Applied Soft Computing*, vol. 14, pp. 397–406, 2014.

[19] M. Strohmeier, M. Schafer, V. Lenders, and I. Martinovic, "Realities and challenges of nextgen air traffic management: the case of ADS-B," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 111–118, 2014.

[20] J. Zhang, L. Wei, and Z. Yanbo, "Study of ADS-B data evaluation," *Chinese Journal of Aeronautics*, vol. 24, no. 4, pp. 461–466, 2011.

[21] A. E. Brownlee, J. Atkin, J. Woodward, U. Benlic, and E. Burke, "Airport ground movement: real world data sets and approaches to handling uncertainty," *Proceedings of the Practice and Theory of Automated Timetabling*, 2014.

[22] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 15, no. 5, pp. 312–328, 2007.

[23] GraphHopper Community, www.graphhopper.com/maps/, accessed June, 2020.

[24] J. S. Kim, J. H. Lee, T. H. Kang, W. Y. Lee, and Y. G. Kim, "Node based map matching algorithm for car navigation system," in *International Symposium on Automotive Technology & Automation*, 1996.

[25] S. Kim and J.-H. Kim, "Adaptive fuzzy-network-based C-measure map-matching algorithm for car navigation system," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 2, pp. 432–441, 2001.

[26] M. A. Quddus, R. B. Noland, and W. Y. Ochieng, "A high accuracy fuzzy logic based map matching algorithm for road transport," *Journal of Intelligent Transportation Systems*, vol. 10, no. 3, pp. 103–115, 2006.

[27] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning – I," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[28] J. M. Mendel, "Type-2 fuzzy sets and systems: an overview," *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 20–29, 2007.

[29] ——, "Uncertain rule-based fuzzy systems," in *Introduction and New Directions*. Springer, 2017, p. 684.

[30] H. B. Sola, J. Fernandez, H. Hagras, F. Herrera, M. Pagola, and E. Barrenechea, "Interval type-2 fuzzy sets are generalization of interval-valued fuzzy sets: Toward a wider view on their relationship," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1876–1882, 2014.

[31] J. M. Mendel, H. Hagras, H. Bustince, and F. Herrera, "Comments on "interval type-2 fuzzy sets are generalization of interval-valued fuzzy sets: Towards a wide view on their relationship"," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 1, pp. 249–250, 2015.

[32] M. Hashemi and H. A. Karimi, "A critical review of real-time map-matching algorithms: Current issues and future directions," *Computers, Environment and Urban Systems*, vol. 48, pp. 153–165, 2014.

[33] N. R. Velaga, M. A. Quddus, and A. L. Bristow, "Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 672–683, 2009.

[34] J. S. Greenfeld, "Matching GPS observations to locations on a digital map," in *81th Annual Meeting of the Transportation Research Board*, vol. 1, no. 3, 2002, pp. 164–173.

[35] C. E. White, D. Bernstein, and A. L. Kornhauser, "Some map matching algorithms for personal navigation assistants," *Transportation Research Part C: Emerging Technologies*, vol. 8, no. 1-6, pp. 91–108, 2000.

[36] J. Chen and M. Bierlaire, "Probabilistic multimodal map matching with rich smartphone data," *Journal of Intelligent Transportation Systems*, vol. 19, no. 2, pp. 134–148, 2015.

[37] M. Quddus and S. Washington, "Shortest path and vehicle trajectory aided map-matching for low frequency GPS data," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 328–339, 2015.

[38] R. Krüger, G. Simeonov, F. Beck, and T. Ertl, "Visual interactive map matching," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 6, pp. 1881–1892, 2018.

[39] M. Maaref and Z. M. Kassas, "A closed-loop map-matching approach for ground vehicle navigation in gnss-denied environments using signals of opportunity," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[40] M. Fu, J. Li, and M. Wang, "A hybrid map matching algorithm based on fuzzy comprehensive judgment," in *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2004, pp. 613–617.

[41] G. Jagadeesh, T. Srikanthan, and X. Zhang, "A map matching method for GPS based real-time vehicle location," *The Journal of Navigation*, vol. 57, no. 3, pp. 429–440, 2004.

[42] S. Syed and M. E. Cannon, "Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons," in *ION National Technical Meeting*, no. 1, 2004, pp. 26–28.

[43] M. Ren and H. A. Karimi, "A chain-code-based map matching algorithm for wheelchair navigation," *Transactions in GIS*, vol. 13, no. 2, pp. 197–214, 2009.

[44] ——, "A fuzzy logic map matching for wheelchair navigation," *GPS Solutions*, vol. 16, no. 3, pp. 273–282, 2012.

[45] J. Tang, S. Zhang, Y. Zou, and F. Liu, "An adaptive map-matching algorithm based on hierarchical fuzzy system from vehicular GPS data," *PloS One*, vol. 12, no. 12, p. e0188796, 2017.

[46] R. Zimmerman, "Method for guiding aircraft on taxiways," 2002, US Patent 6,411,890.

[47] J. G. Herrero, J. B. Portas, and J. C. Corredera, "Use of map information for tracking targets on airport surface," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 2, pp. 675–693, 2003.

[48] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[49] P. M. Larsen, "Industrial applications of fuzzy logic control," *International Journal of Man-Machine Studies*, vol. 12, no. 1, pp. 3–10, 1980.

[50] R. R. Yager and L. A. Zadeh, *An introduction to fuzzy logic applications in intelligent systems*. Springer Science & Business Media, 2012, vol. 165.

[51] L. Suganthi, S. Iniyan, and A. A. Samuel, "Applications of fuzzy logic in renewable energy systems–a review," *Renewable and Sustainable Energy Reviews*, vol. 48, pp. 585–607, 2015.

[52] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, no. 1-4, pp. 195–220, 2001.

[53] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Transactions on Fuzzy systems*, vol. 8, no. 5, pp. 535–550, 2000.

[54] J. M. Mendel, R. I. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 808–821, 2006.

[55] D. Wu, "Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 80–99, 2012.

[56] H. Wu and J. M. Mendel, "Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 5, pp. 622–639, 2002.

[57] H. A. Hagras, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 524–539, 2004.

[58] M.-Y. Hsiao, T.-H. S. Li, J.-Z. Lee, C.-H. Chao, and S.-H. Tsai, "Design of interval type-2 fuzzy sliding-mode controller," *Information Sciences*, vol. 178, no. 6, pp. 1696–1716, 2008.

[59] B.-I. Choi and F. C.-H. Rhee, "Interval type-2 fuzzy membership function generation methods for pattern recognition," *Information Sciences*, vol. 179, no. 13, pp. 2102–2122, 2009.

[60] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, "Medical data classification using interval type-2 fuzzy logic system and wavelets," *Applied Soft Computing*, vol. 30, pp. 812–822, 2015.

[61] O. Obajemu, M. Mahfouf, and J. W. Catto, "A new fuzzy modeling framework for integrated risk prognosis and therapy of bladder cancer patients," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 3, pp. 1565–1577, 2017.

[62] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[63] M. Sugeno, *Industrial applications of fuzzy control*. Elsevier, 1985.

[64] G. Van Brummelen, *Heavenly mathematics: The forgotten art of spherical trigonometry*. Princeton University Press, 2012.

[65] D. E. Knuth, "A generalization of Dijkstra's algorithm," *Information Processing Letters*, vol. 6, no. 1, pp. 1–5, 1977.

[66] A. Brownlee, J. Atkin, J. Woodward, and E. Burke, "Methods and sources for underpinning airport ground movement decision support systems," University of Stirling, Tech. Rep., 2020. [Online]. Available: http://hdl.handle.net/1893/30962

[67] M. Clerc, *Particle swarm optimization*. John Wiley & Sons, 2010, vol. 93.