

# ROPHS: Determine Real-Time Status of a Multi-Carriage Logistics Train at Airport

Shixiong Wang<sup>ID</sup>, Graduate Student Member, IEEE, Chongshou Li<sup>ID</sup>, and Andrew Lim<sup>ID</sup>

**Abstract**—Tracking ground support equipment (GSE) in a high accuracy manner is crucial for both airport safety and optimal management of airport assets but the related researches and products are scarce. Tracking a multi-carriage logistics train is obviously most challenging compared with other single-carriage GSE. In this paper, we design a real-time on-board positioning and heading system (ROPHS) to obtain the real-time status of a multi-carriage logistics train which consists of a powered leading vehicle and one or several non-powered trailing vehicles. The status includes: (a) the accurate positions and velocities of any points on this train, and (b) the alterable number and linking sequence of trailing vehicles at any time of a trip. Technically, the hardware of the system relies on real-time kinematic (RTK) to obtain geolocation of the leading vehicle, and on gyroscopes, magnetometers and accelerometers to obtain headings of all vehicles. A geometry based recurrence algorithm is afterwards presented to calculate the positions of any trailing vehicles. In the end, the multiple model based tracking algorithm is proposed to compute the precision-improved locations and real-time velocities of the whole train. Different from existing traditional GPS or RFID based positioning techniques, the proposed system can reach centimeter-level accuracy, which enables collisions detection, especially those latent collisions that cannot be easily monitored or foreseen by crews' visual inspection.

**Index Terms**—Airport transportation system, aircraft ground service, ground support equipment, real-time tracking, collision detection, multi-carriage logistics train.

## I. INTRODUCTION

ALTHOUGH the aviation industry has developed for more than 70 years, only the optimal scheduling and efficient management of airlines, aircraft, staffs and crews are widely studied. The optimal scheduling and management of airport ground support equipment (GSE) have not been comprehensively focused on in the past. The research on efficient operation of ground support equipment starts from around 2012 [1]. It is the robust increase of annual air traffic since 21st century that raises the importance of such optimal management of GSE to minimize the overheads and maximize the airport operational efficiency [1]. The GSE

Manuscript received 24 March 2020; revised 23 September 2020; accepted 27 January 2021. Date of publication 10 February 2021; date of current version 8 July 2022. This work was supported by the National Research Foundation of Singapore under Grant NRF-RSS2016-004. The Associate Editor for this article was Y. Song. (Corresponding author: Chongshou Li.)

Shixiong Wang and Andrew Lim are with the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 117576 (e-mail: s.wang@u.nus.edu; isealim@nus.edu.sg).

Chongshou Li is with the School of Information Science and Technology, Institute of Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China (e-mail: csliChina2020@outlook.com).

Digital Object Identifier 10.1109/TITS.2021.3055838

1558-0016 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See <https://www.ieee.org/publications/rights/index.html> for more information.

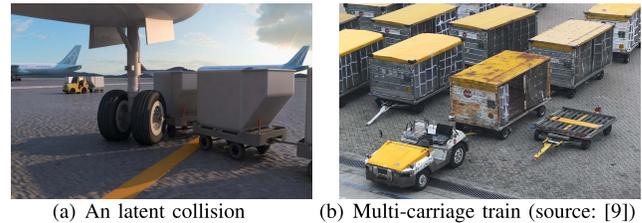


Fig. 1. Examples of collision and a cargo-transit train. (a) Collision between a train and an aircraft; (b) A real multi-carriage train operated at airport ramp.

efficient operations include the ramp safety and the optimal scheduling of GSE [1].

This paper is concerned with the ramp safety (i.e., aircraft ground service safety) problem that is important during airport managerial operations like GSE dispatching and/or scheduling [2]. One of the critical safety concerns is to avoid collisions between aircraft and ground support equipment (GSE). In order to make it, current approach uses visual scans of crews to guarantee a safe distance between aircraft and GSE [3]. This visual inspection is not reliable enough to detect all collisions and shocks imposed on aircraft, especially those caused by moving GSE like tank trucks, shuttles and cargo-transit trains, because the moving GSE likely strikes the aircraft in a short time without the notice of both pilots and vehicle drivers. These latent collisions may result in potential aviation risks and/or large amount of extra overheads. Fig. 1 (a) shows an example. A natural solution is to employ high accuracy tracking technologies and automatically detect collisions. However, this requires centimeter-level positioning accuracy of GSE but existing technologies like Global Positioning System (GPS) [3] and Radio-frequency identification (RFID) [4]–[6] cannot fulfill it. This issue widely exists at airports and is urgently expected to handle [2], [7], for instance, one of largest airports worldwide, Singapore Changi airport. For more information, see an open tender issued by the Civil Aviation Authority of Singapore in 2018 [8].

To this end, in this paper, we propose and design a Real-time On-board Positioning and Heading System (ROPHS) to obtain the real-time and high accuracy status of all types of GSE, especially the multi-carriage logistics train. Such real-time and high-accuracy information obtained from ROPHS makes the collision detection problem efficiently solvable. Besides, worthy of mentioning, with the data-level support of ROPHS, the seamless aviation assets and vehicle optimization system becomes possible to develop [1]. For more information, see

our project website: <https://alim.algorithmexchange.com/caas/>. Since determining the status of a multi-carriage logistics train is the general case of that of a single-carriage GSE, we in this paper mainly discuss the problem for multi-carriage logistics trains. We treat the single-carriage GSE as a special multi-carriage train which only has one leading vehicle and no any trailing vehicles. Besides, this paper mainly focuses on defining the system structure, information streaming logic and tracking algorithms of ROPHS, paying less attention to specific hardware design (like electronic element/unit choosing, PCB layout design, battery deployment, etc.) because product developing is application-oriented and case-by-case.

The rest of the paper is organized as follows. Section II reviews the existing technologies regarding GSE locating and states the necessity of our research. Section III defines notations. In what follows, the system structure, hardware and software of ROPHS are discussed in Section IV, Section V, and Section VI, respectively. Section VII investigates the data preparation for collision detection. At last, experiment results are supplied in Section VIII and the conclusion in Section IX completes this paper.

## II. TECHNOLOGY REVIEW AND PROBLEM FORMULATION

We are concerned with maintaining real-time situational awareness of the multi-carriage train. The train consists of a leading and powered vehicle and several trailing vehicles which are non-powered but wheeled. A real cargo-transit train is displayed in Fig. 1 (b). In [3]–[6], traditional GPS and RFID based methods for GSE positioning are introduced. However, the original purposes of those are just to locate single-carriage GSE vehicles so that crews can easily find them which are randomly distributed on the airport ground. Obviously, the GSE tracking problem proposed in this paper has the following FIVE new challenges:

(a) Traditional GPS and RFID can only offer the positioning accuracy in meter level which cannot support the reliable collision detection task;

(b) If the leading vehicle and trailing vehicles are localized by GPS (or RFID) individually, we have no integrated information of a train. That is, we never know the forming structure of this train such as the number and linking sequence of trailing vehicles. Note that such information is necessary for dynamic and reliable GSE collision detection;

(c) The existing methods failed to take into account the fact that in one trip of a train, the number and linking sequence of trailing vehicles are not fixed. For example, at one time, the topology of a train could be “ $H-S_2-S_1-S_4-S_5$ ”, where  $H$  is for head (leading vehicle) and  $S$  is for slave (trailing vehicle). This train includes 4 trailing vehicles; at another time, the topology could alternatively be “ $H-S_2-S_3$ ”, which includes only 2 trailing vehicles. The subscripts (i.e., 1...5) are permanent IDs of trailing vehicles. This context means, in a trip, the train on its halfway releases some trailing vehicles (specifically,  $S_5$ ,  $S_4$  and  $S_1$ , one by one) and reconnects another trailing vehicle (i.e.,  $S_3$ ). This phenomenon is rather common in real airport operations. Obviously, the reported tracking technologies are not able to automatically monitor and update the changes of connected trailing vehicles;

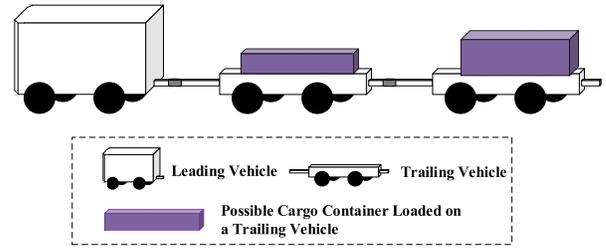


Fig. 2. 3D illustration of a multi-carriage cargo-transit train. The trailing vehicles are assumed to be dollies. Sensors cannot be deployed on the top surfaces of dollies because loaded containers are likely to cover the sensors and stop the sensors from normally working (cf. Fig. 8).

(d) The existing methods cannot solve the heading problem effectively. The popular geomagnetic field based methods cannot normally work in a messy electromagnetic environment due to the electromagnetic noise interference and/or other uncertainties [10]. Plus, the inertial units based methods may also fail to provide reliable data due to accumulated error introduced by the integrator unit. This means extra auxiliary correction mechanism must be designed to periodically calibrate the accumulated error;

(e) Sensors cannot be placed over some trailing vehicles (e.g., dollies) where the cargo container is loaded. It is illustrated by Fig. 2 (and also Fig. 8).

Unfortunately, as highlighted by the five new features above, the airport GSE tracking problem has substantial differences compared with other positioning and tracking problems in other industries like robotics [11], autonomous navigation and driving, automated agriculture [12], [13], etc., which denies the direct transplantation of methods from other industries. Without loss of generality, we take the SLAM (simultaneous localization and mapping, [11]) framework in robotics as an example to support the argument. Here are three reasons:

(a) First, let's discuss the tracking and navigation problem. Although SLAM is popular and nearly mature in robotics, it is intentionally developed for a robot to navigate itself in an unknown environment where the complete information (i.e., the map) of the environment is not available. In order to apply SLAM in an **unknown** environment, the first key step is to set enough number of (omnidirectionally) observable landmarks. Then the direction and range sensors like radar, camera, etc., are used by the robot to position itself. We should mention that as a completely open and known environment, airports have no need to adopt such self-exploration methods which are likely to consume much unnecessary time to train a feasible solution. Because we already have commercially mature and easy-to-use GNSS (global navigation satellite system, e.g., GPS, GLONASS, BEIDOU, etc.) based solutions to locate one vehicle. GNSS based solutions are more attractive for us in this problem because: 1) they do not require to set extra landmarks; 2) they are technically mature which means the solutions are with high reliability and time efficiency (i.e., quick convergence of reliable solution); 3) they are commercially mature which means the modules are cheap; 4) concerning the accuracy issue, the real-time kinematic (RTK, one kind of advanced GNSS

solution [14]) has centimeter level accuracy and precision which is at least no worse than other solutions like SLAM.

- (b) Second, let's discuss the collision detection problem. SLAM is theoretically based on the local view (not a global view) of a sensor like camera. The latent collisions among GSE and aircraft usually happen for the last trailing vehicle of the train (see Fig. 1). In this case, the leading vehicle (where the local-view sensor can be deployed overhead or on the side surfaces) cannot monitor whether the last trailing vehicle is striking an aircraft or not. And, the trailing vehicle itself has no local-view sensors to monitor the collisions. Therefore, we must have a global view of the relative positions of aircraft and trains to decide whether collisions happen.
- (c) Third, we discuss the train-structure monitoring problem. SLAM is not capable of monitoring the real-time number and linking sequence of the trailing vehicles of a train. Because it is not developed for the such purpose.

Therefore, we are expected to design ROPHS to address the new five challenges above, which is also the contribution of this paper. In highlights, the advantages of the proposed solution (over existing solutions in other industries) include:

- (a) It is the first integrated and comprehensive solution for airport GSE accurate tracking and collision detection that simultaneously handles all the new five challenges listed at the beginning of this section;
- (b) It tried best to use commercially mature modules to guarantee the high performances (e.g., quick convergence of algorithms) and robustness (i.e., reliability) while lowering the costs as much as possible. Taking the most expensive module used in the solution (viz., RTK board) as an example, it costs only about 150-200 dollars (USD) if purchased in bulk from UniStrong (<http://www.UniStrong.com>) while its positioning accuracy and precision is in centimeter level. Besides, GNSS based solution does not require extra supporting infrastructures (e.g., landmarks, signal bases etc.) in airports.

### III. NOTATIONS

A multi-carriage train is mathematically formulated as  $H-S_{i_{[1]}}-\dots-S_{i_{[k]}}-\dots-S_{i_{[n]}}$  with five requirements: (a) there is one and only one leading powered vehicle denoted as  $H$ ; (b) this train includes  $n \in \mathbb{N}$  non-powered trailing vehicles and  $n \geq 1$ ; (c) for trailing vehicle  $k$  ( $1 \leq k \leq n$ ),  $i_{[k]}$  is the permanent physical identifier (PID), for any  $1 \leq k < j \leq n$ ,  $i_{[k]} \neq i_{[j]}$ , and  $i_{[k]} \in \{1, \dots, N\}$  ( $n \leq N$ ), where  $N$  denotes the total number of trailing vehicles at an airport; (d) for trailing vehicle  $k$  ( $1 \leq k \leq n$ ),  $k$  is the logic identifier (LID) and  $k \in \{1, 2, \dots, n\}$ ; (e) the LID of the leading vehicle is 0.

Besides, given a train  $H-S_{i_{[1]}}-\dots-S_{i_{[k]}}-\dots-S_{i_{[n]}}$ , time points  $\{1, \dots, t, \dots, T\}$ , the **status of the train** at time  $t$  is defined as  $\{(x_j^t, y_j^t, z_j^t, v_j^t, \theta_j^t) | j = 0, 1, \dots, n\}$  where: (a)  $(x_j^t, y_j^t, z_j^t)$  is the position (in a global geodetic coordinate system we defined at the airport that we are working on),  $v_j^t$  is the instantaneous velocity at  $t$ ,  $\theta_j^t$  is the heading of the vehicle  $j$ , respectively; (b)  $j$  ( $0 \leq j \leq n$ ) is the vehicle LID; (c)  $j \in \{1, \dots, n\}$  and

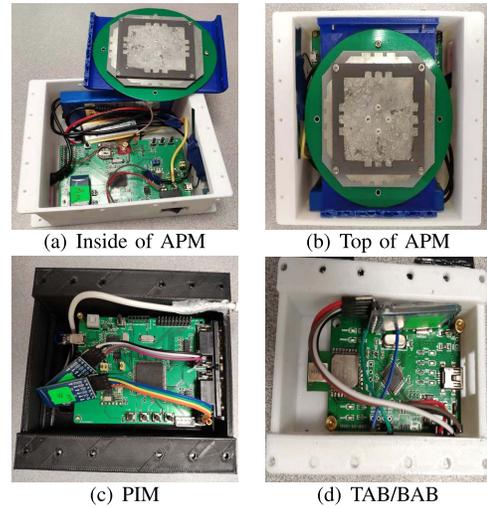


Fig. 3. Prototypes of ROPHS. Note that TAB and BAB are physically identical. They have different embedded code (i.e., different working mode).

$t \in \{1, \dots, t, \dots, T\}$ ,  $z_j^t$  is assumed to be a fixed constant  $h_j$  and the ROPHS only needs to compute  $(x_j^t, y_j^t)$ . Intuitively, the status of a train contains the position, heading, and velocity of any vehicle of this train.

### IV. SYSTEM STRUCTURE

The system mainly involves of two physical parts: (a) active positioning module (APM) for the leading vehicle, and (b) passive information module (PIM) for the trailing vehicle(s). The APM is designed to obtain the real-time position and heading of the leading vehicle, while each PIM is used to obtain the real-time heading of each trailing vehicle. APM and PIMs communicate with each other to share the information for further data fusion. The tag accessory board (TAB) and base accessory board (BAB) are designed to identify the real-time number and linking sequence of the trailing vehicles of a train. TAB(s) and BAB(s) are accessories of APM and PIM(s). Prototypes are given in Fig. 3.

As for the software part, the communication logic of these physical components, a geometry based recurrence positioning algorithm and an advanced moving-object tracking algorithm is developed to determine the real-time status of the whole train. In summary, the diagram of ROPHS is given in Fig. 4.

*Remark 1:* In this paper, we technically differentiate two terms: positioning and tracking. Positioning emphasizes the process that determines the position of a point based on direct measurements from sensors at a time instant, while tracking stresses the process that continuously positioning and simultaneously improving the precision of positioning by integrating successive measurements and other mechanisms. For more information, refer to [15], [16].

### V. HARDWARE DESIGN

In this section, we discuss the hardware design, implementation and installation of the ROPHS. Although the communication logic among hardware is of software part, we clarify it in the end of this section to improve the readability.

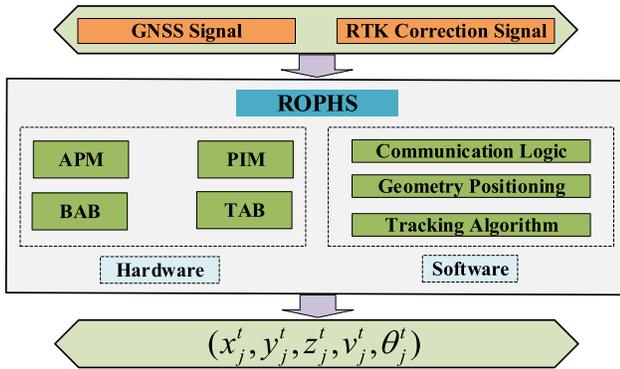


Fig. 4. Diagram of ROPHS. Its hardware consists of APM, PIM, BAB, and TAB. The software part is for data fusion and information integration. The inputs of the system are the GNSS signal and RTK correction signal while the output is the real-time status of the train. The geolocation is technically based on RTK which requires GNSS signal and RTK correction signal.

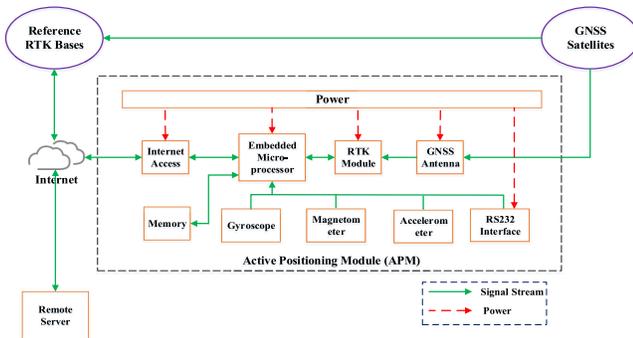


Fig. 5. Schematic of APM.

#### A. APM

The APM's schematic is shown in Fig. 5. This module relies on RTK [14] to determine the accurate position of the point on the leading vehicle where APM is placed. In order to obtain the heading of the leading vehicle and afterwards jointly determine the status of the whole train, information from gyroscope, magnetometer and accelerometer (GMA) is integrated to obtain the robust heading of the leading vehicle. The APM shares information of the whole train with a remote server via Internet. The remote server is the integrated aviation assets management system. The GNSS antenna of APM receives the GNSS signals from satellites which can be BEIDOU, GPS, GLONASS, GALILEO *et al.* The RTK correction signals received by APM are transmitted from the reference RTK base station via Internet. The RTK module calculates the current accurate and precise position of APM based on the GNSS signals and RTK correction signals. The RS232 interface is for wired communication with the corresponding TAB and/or BAB. The embedded micro-processor computes the real-time status of the leading vehicle as well as that of the whole train based on: (a) all information collected by sensors, (b) the geometry based positioning algorithm, and (c) the multiple model based tracking algorithm.

#### B. PIM

Fig. 6 (a) gives the schematic of PIM which is used to compute the heading(s) of trailing vehicle(s). The PIM

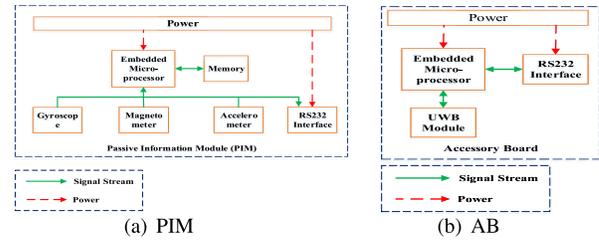


Fig. 6. Schematic of PIM and AB.

contains an embedded microprocessor, a memory, an inertial unit (digital gyroscope), a magnetometer, an accelerometer, a RS232 interface, and a battery. The GMA is for heading. The PIM is connected wiredly to the corresponding BAB and TAB by a RS232 interface. There is no RTK in PIM to actively obtain the geolocation of the trailing vehicle. This is because of two main reasons: (a) RTK and GNSS antenna are overwhelmingly expensive compared to GMA. Integrating a RTK module into every PIM causes unnecessary costs; (b) Dollies cannot accept the installation of GNSS antenna over its top surfaces (see Fig. 2). Alternatively, the location(s) of trailing vehicle(s) are determined by the geometry based positioning algorithm and the multiple model based tracking algorithm with information from both APM and PIM(s).

#### C. BAB and TAB

The schematic of accessory board (AB) is given in Fig. 6 (b). It consists of a microprocessor, a UWB module, a battery and a RS232 interface. The AB can work as either BAB mode or TAB mode. The BAB-TAB pair is mainly used for ranging. The ranging mechanism works under symmetric-double-sided two-way time-of-arrival (SDS-TW-TOA) protocol based on IEEE Standard 802.15.4a [17]. Thus, one BAB and another one TAB are required to pair with each other. The UWB module in BAB acts as base (also known as anchor) node while the UWB module in TAB acts as tag node. RS232 is used to communicate with corresponding APM or PIM.

Now we illustrate how the BAB and TAB pair can determine the number and the linking sequence of a train. Generally, a BAB could sense all TABs within a limited region, for example, a circle with radius of 10 metres centred at the BAB. However, the TAB and BAB pair with each other only when the distance between them is the smallest. An illustration is given in Fig. 7, where a train,  $H-S_1-S_2$ , is given and there are five trailing vehicles around but not linked to it. We take the BAB on  $S_1$  as an example and see how it pairs with the closest TAB on  $S_2$ . This BAB could sense 7 TABs which have been connected by dotted lines in Fig. 7 (a). However, only the distance between BAB of  $S_1$  and TAB of  $S_2$  is the smallest and they are valid to build a UWB pair. Once the pair is established,  $S_2$  is connected to  $S_1$  and a sub-linking sequence  $S_1-S_2$  is generated. Note that when  $S_1$  and  $S_2$  are mechanically linked, the TAB on  $S_2$  is even closer to BAB on  $S_1$  than the TAB on  $S_1$ . This is because the width of trailing vehicle is greater than the width of space between two adjacent vehicles. Even if the train is running at airport, the above rational still holds. By this wireless ranging mechanism and

TABLE I  
RECOMMENDED COMMERCIAL MODELS

| Module    | Suggested Model | Produced by | Performances (real field tested) |  | Business Website               |
|-----------|-----------------|-------------|----------------------------------|--|--------------------------------|
|           |                 |             | Accuracy                         | Precision                                  |                                |
| RTK Board | P327            | Hemisphere  | 0                                | 0.7 cm <sup>2</sup> (variance)             | www.hemispheregnss.com         |
| GMA       | WT101           | Wit-motion  | 0 (calibrated)                   | 0.0025 deg <sup>2</sup> (variance)         | www.wit-motion.com/english.php |
| UWB       | DWM1000         | Decawave    | 0 (calibrated)                   | 2.5 cm <sup>2</sup> for ranging (variance) | www.decawave.com               |
| BLE 5.0   | HC-42           | HC Tech     | 0 (calibrated)                   | 0.11 m <sup>2</sup> for ranging (variance) | www.hc01.com                   |

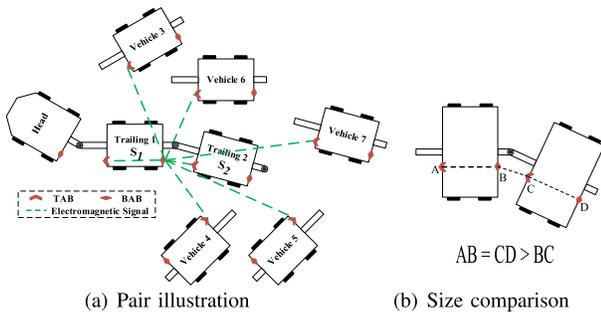


Fig. 7. Pair of BAB and TAB determining linking sequence.

*periodic* checks of validity of UWB pairs, we can identify the real-time number and linking sequence of the whole train. Note again that during airport ground handling, the linking sequence is not fixed so that wired connection between two adjacent vehicles is not applicable.

When the BAB-TAB pair is established, the heading information from PIM is transmitted by this UWB electromagnetic channel to its preceding PIM or APM for data fusion. Since the UWB ranging mechanism is based on SDS-TW-TOA message packages between the UWB pairs, the communication information which contains the heading could also be coded in the SDS-TW-TOA message packages. Note that the SDS-TW-TOA protocol relies on the IEEE Standard 802.15.4a. [17].

*Remark 2:* In our prototype (see Fig. 3), we adopted the UWB ranging and communication pairs in BAB and TAB. However, the UWB ranging and communication pairs in BAB and TAB could be replaced by BLE (Bluetooth Low Energy) 5.0 or later BLE versions. Since we do not care more about the ranging accuracy and only focus on the smallest distance amongst all available ranges to build a pair (see Fig. 7), the BLE ranging protocol (i.e., iBeacon [18]) could also be considered for lower energy consumption. Although the one-time ranging accuracy of BLE is low, usually in meter level, we can improve the ranging accuracy with multiple successive range measurements via filtering methods like the Kalman filter [19]. In our field test, we set the sampling time of BLE as 0.1 seconds and we use the canonical exponential smoothing method (coefficient: 0.5) to denoise the range measurements. We found that although the ranging error of the BLE 5.0 is large (error range after filtering:  $\pm 0.65$ m), the linking sequence of trailing vehicles could be correctly identified. The BLE 5.0 module that we recommend is listed in Table I.

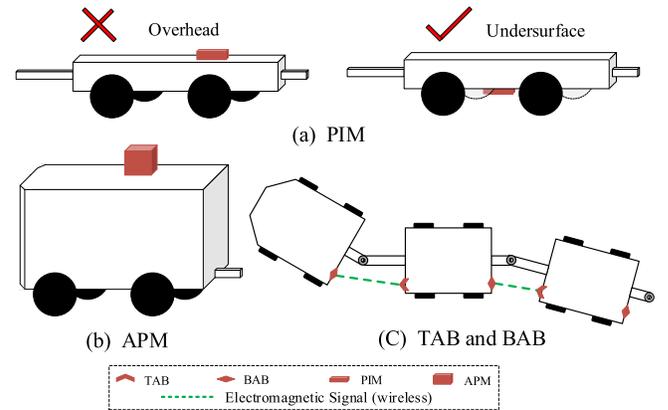


Fig. 8. Placements of components of ROPHS on the train.

#### D. Installation

In this section, we discuss how to physically install the components of ROPHS on the train. Fig. 8 (a), (b) suggest a possible placement of PIM and APM, respectively. The APM requires open enough area to receive sufficient satellites signals. The PIM should be placed undersurface below the trailing vehicles. Because some trailing vehicles like dollies in an airport cannot accept an overhead deployment. For example, we can see the loaded container may occupy the top surface of a dolly in Fig. 2. The BAB is placed in the rear of vehicles, including both leading and trailing vehicles. The TAB is deployed in the front of trailing vehicle(s). They are illustrated in Fig. 8 (c). The overall placement scheme and all communication channels of the ROPHS system is showed in Fig. 9. Note that the information transmitting channels between APM and BAB, PIM and TAB, and PIM and BAB are all wired (in order to guarantee the communication reliability) while the connection between BAB and TAB is wireless.

#### E. Communication Logic

In this subsection, the communication logic of the ROPHS is presented. Given a train  $H-S_{i[1]} \dots -S_{i[n]}$  and time  $t$ , we first present the logic for the trailing vehicle  $i_{[j]}$  ( $j = \{1, \dots, n\}$ ). The PIM on vehicle  $i_{[j]}$  first gets its heading  $\theta_j^t$  and PID  $i_{[j]}$ . Then there are two cases:

- If it is the last trailing vehicle linked on the train (i.e.,  $j = n$ ), PIM sends this information to the associated TAB via RS232 which is a wired channel. The TAB forwards the received to paired BAB on vehicle  $i_{[j-1]}$  by the wireless electromagnetic channel;

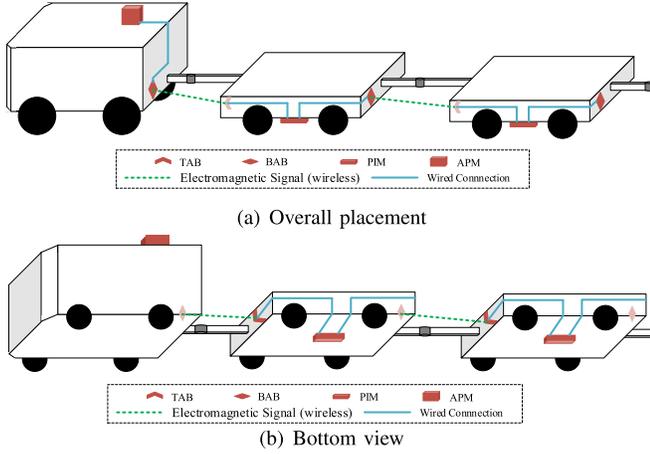


Fig. 9. Overall placement of the ROPHS on the train.

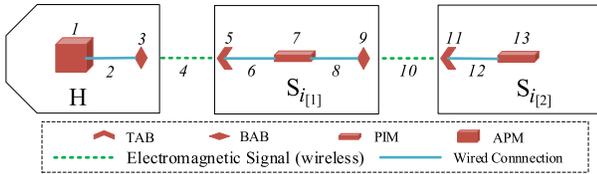


Fig. 10. Labels for communication channel and logic presentation.

(b) If it is not the last vehicle (i.e.,  $1 \leq j < n$ ), the PIM on vehicle  $i_{[j]}$  gets information  $\{(\theta_k^t, i_{[k]}) | k = j + 1, \dots, n\}$  from BAB via RS232. Then PIM generates a new message package  $\{(\theta_k^t, i_{[k]}) | k = j, \dots, n\}$  (which adds the information of itself) and sends to BAB on vehicle  $i_{[j-1]}$  similarly as Case (a).

For the leading vehicle  $H$ , the APM first receives headings and PIDs,  $\{(\theta_k^t, i_{[k]}) | k = 1, \dots, n\}$ , of all trailing vehicles from the connected BAB. It also obtains location and heading  $(x_0^t, y_0^t, \theta_0^t)$  of the leading vehicle from sensors. Based on them, the APM uses a geometry based positioning algorithm and multiple model based tracking algorithm to compute velocities  $v_j^t$  ( $j = 0, \dots, n$ ) and precision-improved locations of the whole train, including the leading vehicle and all trailing vehicles  $(\hat{x}_j^t, \hat{y}_j^t)$  ( $j = 0, \dots, n$ ). Then the APM sends train's status  $\{(\hat{x}_j^t, \hat{y}_j^t, h_j, v_j^t, \theta_j^t) | j = 0, 1, \dots, n\}$  ( $h_j$  is constant and the height of vehicle  $j$ ) at time period  $t$  to remote server via Internet for higher-level functions like collision detection and optimal scheduling. Algorithm 1 details the communication logic of the train  $H-S_{i_{[1]}}-S_{i_{[2]}}$  step by step. Fig. 10 provides a visual illustration and displays the label of each component mentioned in the algorithm.

#### F. Recommended Chips

In order to help inspired readers to reproduce our solution, we share some reliable commercial models of RTK, GMA, UWB, and BLE, listed in Table I. Table I means the fluctuation of position measurements would be limited within 5cm ( $\pm 3 \times \sqrt{0.7\text{cm}^2} = \pm 2.5\text{cm}$ ), while the fluctuation of heading measurements would be within 0.3deg ( $\pm 3 \times \sqrt{0.0025\text{deg}^2} = \pm 0.15\text{deg}$ ). Note that here we use the “ $3\sigma$ ” rule, where  $\sigma$  denotes the standard deviation.

#### Algorithm 1 Communication Logic of ROPHS

- 1: **Note:** Arrows define the information streams. The message packages are above arrows, while the communication channels are below arrows. See also Fig. 10 for illustration.
- 2: **Initialize:**  $t \leftarrow 1$
- 3: **repeat**
- 4: PIM 13 gets heading  $\theta_2^t$
- 5: Send to TAB 11: PIM 13  $\xrightarrow[\text{RS232 12}]{(\theta_2^t, i_{[2]})}$  TAB 11
- 6: Relay to BAB 9: TAB 11  $\xrightarrow[\text{UWB channel 10}]{(\theta_2^t, i_{[2]})}$  BAB 9
- 7: Forward to PIM 7: BAB 9  $\xrightarrow[\text{RS232 8}]{(\theta_2^t, i_{[2]})}$  PIM 7
- 8: PIM 7 gets heading  $\theta_1^t$
- 9: Send to TAB 5: PIM 7  $\xrightarrow[\text{RS232 6}]{\{(\theta_j^t, i_{[j]}) | j=1,2\}}$  TAB 5
- 10: Relay to BAB 3: TAB 5  $\xrightarrow[\text{UWB channel 4}]{\{(\theta_j^t, i_{[j]}) | j=1,2\}}$  BAB 3
- 11: Forward to APM 1: BAB 3  $\xrightarrow[\text{RS232 2}]{\{(\theta_j^t, i_{[j]}) | j=1,2\}}$  APM 1
- 12: APM 1 obtains  $\theta_0^t$  and  $(x_0^t, y_0^t)$
- 13: APM 1 calls positioning algorithm and tracking algorithm to compute velocities  $(\hat{v}_0^t, \hat{v}_1^t, \hat{v}_2^t)$  and precision-improved locations  $\{(\hat{x}_j^t, \hat{y}_j^t) | j = 0, 1, 2\}$
- 14: Transmit to server: APM 1  $\xrightarrow[\text{Internet}]{\{(\hat{x}_j^t, \hat{y}_j^t, h_j, v_j^t, \theta_j^t) | j=0,1,2\}}$  remote server
- 15:  $t \leftarrow t + 1$
- 16: **until**  $t$  is the last time period

## VI. SOFTWARE DESIGN

In this section, we proceed to software design of the ROPHS. It involves the geometry based positioning algorithm, the multiple model based tracking algorithm, and the heading calibration method. The main task of software is to produce real-time status of the train based on raw information generated by hardware. We use the train  $H-S_1-S_2$  as running example in this section.

#### A. Geodetic Coordinate Systems

The geodetic coordinate systems (GCS) and parameters about the physical size of the train are illustrated in Fig. 11. In Fig. 11 (a), there is a global geodetic coordinate system (GGEC)  $x_e-O_e-y_e$  and  $O_e$  is the origin chosen on the airport ground. The APM and PIM are placed on vehicles along their axes of symmetry. At time  $t$ , for vehicle  $j$  ( $j = 0, 1, 2$ ), we define local geodetic coordinate system (LGCS)  $x_j^t-O_j^t-y_j^t$  and  $O_j^t = (x_j^t, y_j^t)'$  is a two dimensional column vector. Here  $x_j^t$  and  $y_j^t$  are the coordinates in  $x_e$  and  $y_e$  axes respectively, and symbol  $'$  is the transpose operator. Point  $O_j^t$  is the location of APM ( $j = 0$ ) or PIM ( $j > 0$ ) at time point  $t$ ; the heading  $\theta_j^t$  is defined as the angle between  $y_j^t$  axis of LGCS and  $y_e$  axis of GGEC. Note that the heights of GSE vehicles are considered as constants. Thus, coordinate system here only consists of two dimensions. In addition, we present several geometry parameters about the physical size of the train in Fig. 11 (b).

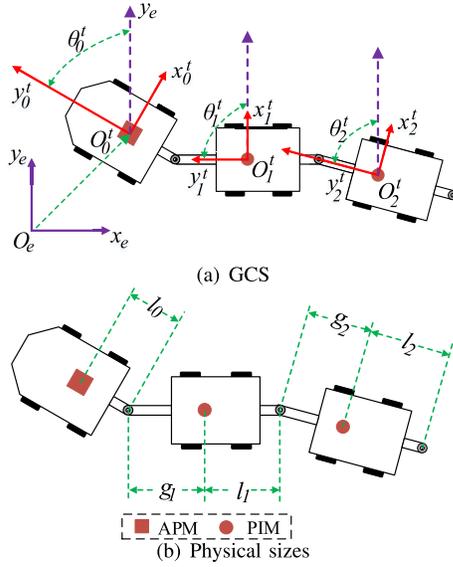


Fig. 11. Geodetic coordinate system and key physical sizes of the train.

We assume that these parameters are constant and do not vary over time.

### B. Geometry Based Positioning Algorithm

The positioning algorithm determines the real-time positions of the trailing vehicles. Considering  $H$ - $S_1$ - $S_2$  at time point  $t$ , the APM generates its longitude and latitude which can be transformed into the GGEC, i.e.,  $x_e$ - $O_e$ - $y_e$  [20]. We also note that headings  $\theta_j^t$  ( $j = 0, 1, 2$ ) can be obtained from APM ( $j = 0$ ) and PIM  $j > 0$ . For trailing vehicle  $j$ , the position of its PIM  $O_j^t$  is computed by the following Lemma.

*Lemma 1:* Given a train  $H$ - $S_1$ - $S_2$ - $\dots$ - $S_j$ , time point  $t$  ( $t \in \{1, \dots, T\}$ ), two dimensional column vector  $O_j^t$ , heading  $\theta_j^t$ , geometry parameters  $l_j$  and  $g_j$  defined in Fig. 11 (b), the following equation holds.

$$\begin{aligned} O_j^t &= O_{j-1}^t + Q'(\theta_{j-1}^t) \cdot [L_{j-1} + Q'(\Delta\theta_j^t) \cdot G_j] \\ &= O_0^t + \sum_{k=0}^{j-1} \{Q'(\theta_k^t) \cdot [L_k + Q'(\Delta\theta_{k+1}^t) \cdot G_{k+1}]\}, \quad (1) \end{aligned}$$

where  $\Delta\theta_j^t := \theta_j^t - \theta_{j-1}^t$ ,  $L_j := (0, -l_j)'$ ,  $G_j := (0, -g_j)'$  and

$$Q(\theta) := \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}.$$

*Proof:* This is by the geometry relations. Note that  $j = 1, 2$  for the running example in this section.  $\square$

Note that we use the points where we deploy the APM and PIM(s) to represent the positions of the leading vehicle and trailing vehicle(s), respectively.

### C. Multiple Model Based Tracking Algorithm

This subsection discusses how to improve the positioning precision and estimate the velocities of the train. The dynamics models that possibly match the running dynamics of a train well at an airport are: (a) constant velocity (CV) model, (b) constant acceleration (CA) model, (c) Singer model, and (d)

current statistics (CS) model. For more on these models, see [15]. The CV and CS model are appropriate for slow moving pattern and straight-line trajectories while the CA and Singer model are suitable for relatively high maneuvering and curve trajectories. Note that the running speed of GSE at an airport is limited (not greater than  $25\text{km/h}$ ) [7]. Since the train at an airport usually runs in straight lines and occasionally maneuvers, the CV model and CS model should fit the dynamics well in most of the time.

For vehicle  $j$  ( $j = 0, 1, 2$ ), we now illustrate how to track it and compute velocity. The tracking system is modelled by the Markov jump linear system [21], [22] as (2) and (3)

$$\begin{cases} X_{\mathcal{M}}^{t+1} = \Phi_{\mathcal{M}} X_{\mathcal{M}}^t + G_{\mathcal{M}} W^t \\ Y^t = H_{\mathcal{M}} X_{\mathcal{M}}^t + V^t, \end{cases} \quad (2)$$

where  $\mathcal{M}$  is model label and  $\mathcal{M} \in \{\text{CV}, \text{CA}, \text{Singer}\}$ ,  $t$  is discrete time point,  $W^t$  and  $V^t$  are process noise vector and measurement noise vector with proper dimensions, respectively, and  $Y^t$  denotes the measurement vector. If model  $\mathcal{M}$  is CS, the equation becomes

$$\begin{cases} X_{\text{CS}}^{t+1} = \Phi_{\text{CS}} X_{\text{CS}}^t + (G_{\text{CA}} - G_{\text{CS}}) \bar{A}^t + G_{\text{CS}} W^t \\ Y^t = H_{\text{CS}} X_{\text{CS}}^t + V^t, \end{cases} \quad (3)$$

in which

$$\bar{A}^{t+1} = e^{-\alpha \Delta t} \begin{bmatrix} \hat{a}_{jx}^t \\ \hat{a}_{jy}^t \end{bmatrix} + (1 - e^{-\alpha \Delta t}) \bar{A}^t,$$

and the state vectors  $X_{\mathcal{M}}^t$ , system matrices  $\Phi_{\mathcal{M}}$ , noise driven matrices  $G_{\mathcal{M}}$  and measurement matrices  $H_{\mathcal{M}}$  are defined as

$$\begin{aligned} X_{\text{CV}}^t &= (x_j^t, v_{jx}^t, y_j^t, v_{jy}^t)', \\ X_{\text{CA}}^t &= X_{\text{Singer}}^t = X_{\text{CS}}^t = (x_j^t, v_{jx}^t, a_{jx}^t, y_j^t, v_{jy}^t, a_{jy}^t)', \\ \Phi_{\text{CV}} &= \begin{bmatrix} I_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & I_{2 \times 2} \end{bmatrix} \otimes \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \\ \Phi_{\text{CA}} &= \begin{bmatrix} I_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \otimes \begin{bmatrix} 1 & \Delta t & (\Delta t)^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \\ \Phi_{\text{Singer}} &= \Phi_{\text{CS}} \\ &= \begin{bmatrix} I_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \\ &\otimes \begin{bmatrix} 1 & \Delta t & (\alpha \Delta t - 1 + e^{-\alpha \Delta t})/\alpha^2 \\ 0 & 1 & (1 - e^{-\alpha \Delta t})/\alpha \\ 0 & 0 & e^{-\alpha \Delta t} \end{bmatrix}, \\ G_{\text{CV}} &= \begin{bmatrix} I_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & I_{2 \times 2} \end{bmatrix} \otimes \begin{bmatrix} (\Delta t)^2/2 \\ \Delta t \end{bmatrix}, \\ G_{\text{CA}} &= \begin{bmatrix} I_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \otimes \begin{bmatrix} (\Delta t)^2/2 \\ \Delta t \\ 1 \end{bmatrix}, \\ G_{\text{Singer}} &= G_{\text{CS}} \\ &= \begin{bmatrix} I_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \\ &\otimes \begin{bmatrix} (\alpha \Delta t - 1 + e^{-\alpha \Delta t})/\alpha^2 \\ (1 - e^{-\alpha \Delta t})/\alpha \\ e^{-\alpha \Delta t} \end{bmatrix}, \\ H_{\text{CV}} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\ H_{\text{CA}} &= H_{\text{Singer}} = H_{\text{CS}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \end{aligned}$$

in which  $v_{jx}^t, v_{jy}^t, a_{jx}^t$  and  $a_{jy}^t$  are true velocities and accelerations of the vehicle  $j$  at time  $t$  in  $x_e$  and  $y_e$  axes;  $\hat{v}_{jx}^t, \hat{v}_{jy}^t, \hat{a}_{jx}^t$  and  $\hat{a}_{jy}^t$  are estimated values of them, respectively;  $\mathbf{I}_{2 \times 2}$  and  $\mathbf{0}_{2 \times 2}$  are  $2 \times 2$  identity and zero matrix;  $\Delta t$  is the time slot between  $t + 1$  and  $t$ ;  $\alpha$  is the reciprocal of the maneuver time constant [15], [23];  $\otimes$  stands for the Kronecker product.

Let  $\mathbf{Y}^t(k)$ ,  $k = 1, 2$ , denote the  $k^{\text{th}}$  entry of vector  $\mathbf{Y}^t$ . Note that  $\mathbf{Y}^t(1)$  and  $\mathbf{Y}^t(2)$  are noisy measurements of  $x_j^t$  and  $y_j^t$ , respectively, which are directly from Eq. (1). As for initial model probability, it is safe to set as  $(\text{CV}, \text{CA}, \text{Singer}, \text{CS})' = (0.7, 0.1, 0.1, 0.1)'$  and initial model transition probability matrix as

$$\mathcal{P}_0 = \begin{bmatrix} 0.91 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.91 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.91 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.91 \end{bmatrix}. \quad (4)$$

This is by experience. After applying the interactive multiple model with canonical Kalman filter (IMM-KF) [21], [22], the estimated position  $\hat{\mathbf{O}}_j^t = (\hat{x}_j^t, \hat{y}_j^t)'$  and velocity  $\hat{\mathbf{v}}_j^t = (\hat{v}_{jx}^t, \hat{v}_{jy}^t)'$  are computed. Compared with  $\mathbf{O}_j^t$  that are directly from the sensor ( $j = 0$ ) or indirectly from Lemma 1 ( $j > 0$ ), the estimated  $\hat{\mathbf{O}}_j^t = (\hat{x}_j^t, \hat{y}_j^t)'$  are more precise. Since IMM-KF is not very sensitive to the initial model probability, arbitrarily assigning a reasonable initial value does not cause a disaster.

#### D. Heading Calibration

Heading information is generated by the GMA module for leading or trailing vehicles. It is likely that drift problem happens. Therefore, the periodic calibration (reset) is necessary, it is convenient to handle under the IMM-KF frame. Specifically, when the CV or CS model is dominant over other models for a relative long time, it is safe to reset heading of leading vehicle  $\theta_0^t$  at time  $t$  as  $\theta_0^t := \arctan(\hat{v}_{0x}/\hat{v}_{0y})$ . We note that this calibration formula is only applicable for the leading vehicle ( $j = 0$ ). It does not work for trailing vehicles. However, if the train kept in the CV/CS mode for a long time, the train would become straight. In this case, for trailing vehicle  $j$  ( $j > 0$ ) at time point  $t$ , just let  $\theta_j^t := \theta_0^t$ . Then headings of trailing vehicles are also calibrated.

#### VII. DATA PREPARATION FOR COLLISION DETECTION

The specific techniques for 3D collision detection are mature and available for application [24]. This paper only aims at providing the required data to support such task.

Lemma 1 provides positions of all vehicles including both the leading and trailing vehicles. But it only gives the locations where the APM or PIM are installed. In order to effectively detect collision, corner points of the vehicles are more interested to monitor. For instance, the corner point of the last trailing vehicle of a train is more likely to strike other assets at an airport (see Fig. 1 (a)). Given trailing vehicle  $j$  at time  $t$ , we aim to calculate the position of red-filled tetragon corner point which is shown in Fig. 12. We assume its location in LGCS  $x_j^t - \mathbf{O}_j^t - y_j^t$  is  $\mathbf{P}_j^t$ . Its real-time coordinates  $\mathbf{P}_j^t$  in GGEC  $x_e - \mathbf{O}_e - y_e$  can be computed by

$$\mathbf{P}_j^t = \hat{\mathbf{O}}_j^t + \mathbf{Q}'(\theta_j^t) \cdot \mathbf{P}_j^t, \quad (5)$$

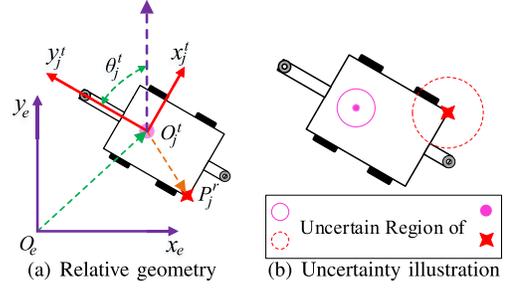


Fig. 12. Relative geometry and uncertainty illustration of a corner point on the trailing vehicle  $j$ . In (b), the radius of a uncertain region is determined by the “ $3\sigma$ ” rule, where  $\sigma$  denotes the standard deviation of its corresponding stochastic point (i.e., the red-filled tetragon corner point).

where  $\hat{\mathbf{O}}_j^t$  is position of vehicle  $j$  calculated by the IMM-KF algorithm with raw measurement  $\mathbf{O}_j^t$ , which is a precision-improved value. If  $j = n$ , it helps detect the latent collision in Fig. 1 (a).

According to Eq. (1) and (5), we see that the corner point  $\mathbf{P}_j^t$  and the positions of the dollies  $\mathbf{O}_j^t$  are related to the position of the leading vehicle  $\mathbf{O}_0^t$  and the headings  $\theta_j^t$ ,  $j = 0, 1, 2, \dots, n$ . However,  $\mathbf{O}_0^t$  and  $\theta_j^t$  are all Gaussian random variables with zero-mean and variances given in Table I (see the Precision columns). For reliable collision detection, we should also investigate the variances of the corner point  $\mathbf{P}_j^t$ , see Fig. 12 (b). Since the status transfer functions (1) and (5) are highly nonlinear, we use the Unscented Transformation (UT) to handle this problem. The UT uses the sigma points to approximate the mean and variance (or covariance matrix) of the random variable after a nonlinear transform of other random variable(s) [22]. Intuitively, if we have a nonlinear transformation  $u = f(\mathbf{w})$  from a random vector  $\mathbf{w}$  to a random variable  $u$ , suppose we know the mean and covariance of  $\mathbf{w}$ , we could use UT to estimate the mean and variance of  $u$ . Note that the mean propagation equation for  $\mathbf{P}_j^t$  is already given by Eq. (1) and (5).

#### VIII. EXPERIMENT RESULTS

All the related source data, source codes and detailed usage instructions are available online at GitHub: <https://github.com/Spratm-Asleaf/GSE-Tracking>. In this section, we only demonstrate the main results.

First, as a simple demonstration to positioning performance of APM, in Fig. 13, we draw a small piece of trajectory that APM generates in a field test. As we can see, the trajectory is smooth, indicating the precision of positioning is pretty high. This is also supported by the fact that the trajectory variance (fluctuations) is  $0.645\text{cm}^2$  (i.e., the error range is  $\pm 3 \times \sqrt{0.645\text{cm}^2} = \pm 2.41\text{cm}$ ).

Second, in Fig. 14, we show the validity of our geometry based positioning method (i.e., Eq. (1)) as well as the illustration of uncertainty region of a corner point on the last trailing vehicle. In Fig. 14 (a), the positions of two trailing vehicles are calculated by Eq. (1). We conduct 500 times of simulation for this train with 500 *i.i.d.* respective uncertainties. However, we only plot 20 times for figure clearness. In Fig. 14 (b), we show the uncertainty region of the corner

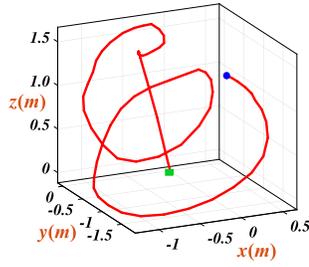


Fig. 13. Small piece of trajectory that APM generates. Unit: meter; Blue circle: starting point; Green rectangle: ending point; z-axis: the height.

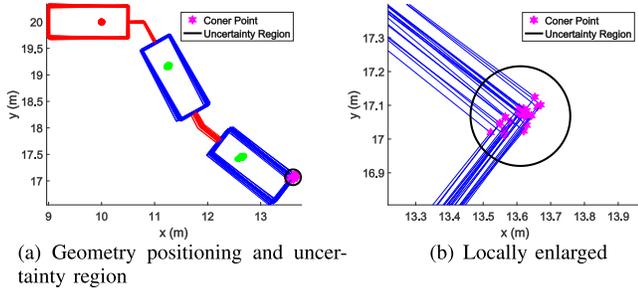


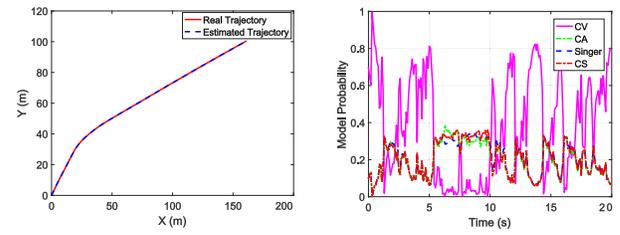
Fig. 14. Uncertainty illustration of a corner point on the last trailing vehicle.

TABLE II  
ERROR RANGE OF ESTIMATION AND OBSERVATION

|           | Estimation            | Observation           |
|-----------|-----------------------|-----------------------|
| In x axis | $\pm 2.0987\text{cm}$ | $\pm 2.5219\text{cm}$ |
| In y axis | $\pm 1.8243\text{cm}$ | $\pm 2.7576\text{cm}$ |

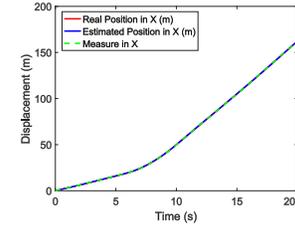
point. Results show that the radius of the region is 0.1477m, which is consistent with Fig. 14 (b). That is, (virtually) all of the corner points are located inside the uncertainty region.

Last, in Fig. 15, we show the reliability of our multiple model based tracking algorithm. We are working on a part of trajectory (20 seconds in total with sampling period 0.1 seconds) of APM in a field test. We aim to obtain the precision-improved location of APM and estimate the velocity of it as well. The precision-improved location is termed as estimated position in our tracking frame, which corresponds to the direct observed position. As we can see from Fig. 15 (d) and (h), the estimated positions have lower error than the directly measured positions. This is also supported by the fact given in Table II. Note that the observation error is consistent with that in Table I, in which the error range is  $\pm 2.5\text{cm}$ . Besides, Fig. 15 (b) displays the model probability of CV, CA, Singer and CS over time. It indicates that the CV model is dominating compared to others in time windows [0, 5] and [10, 20] (i.e., the true moving pattern of the vehicle then is CV), while CA and Singer are dominating compared to others in the time window [5, 10] (i.e., the vehicle then is maneuvering so that acceleration is nonzero). This phenomenon could also be observed from Fig. 15 (a), where the trajectory changes its direction within the time window [5, 10].

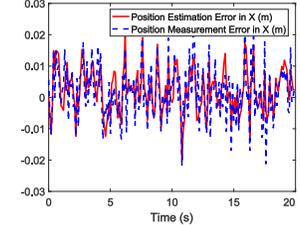


(a) True and estimated trajectory

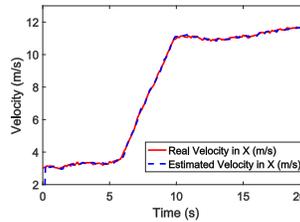
(b) Model probability over time



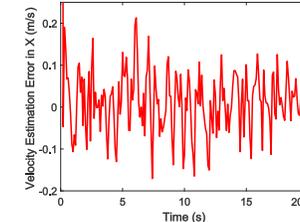
(c) True and estimated position in x axis



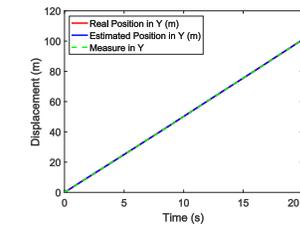
(d) Estimation and measurement error of position in x axis



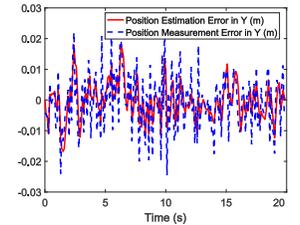
(e) True and estimated velocity in x axis



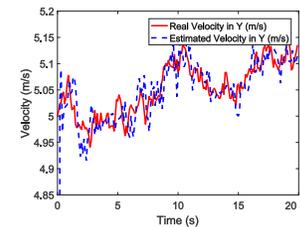
(f) Estimation error of velocity in x axis



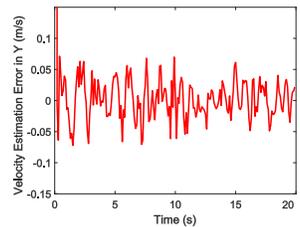
(g) True and estimated position in y axis



(h) Estimation and measurement error of position in y axis



(i) True and estimated velocity in y axis



(j) Estimation error of velocity in y axis

Fig. 15. Tracking results of APM.

### IX. CONCLUSION

In this paper we study the problem of determining the high accuracy, real-time positions and headings of a multi-carriage logistic train at airport. It has challenging requirements like (a) centimeter-level accuracy and (b) alterable number and linking sequence of trailing vehicles. To the best of the authors' knowledge, the ROPHS is the first comprehensive and systematic approach for real-time tracking of the airport

multi-carriage logistic train with centimeter-level accuracy, which makes possible for collision detection. In a bigger view, this paper displays the data-gathering-level support for developing our Aviation Assets and Vehicle Optimization system. For more information, see [1] and our project website: <https://alim.algorithmexchange.com/caas/>.

## X. ACKNOWLEDGMENT

This article was finished when C. Li was with the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore.

## REFERENCES

- [1] S. Wang, Y. Che, H. Zhao, and A. Lim, "Accurate tracking, collision detection, and optimal scheduling of airport ground support equipment," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 572–584, Jan. 2021.
- [2] J. Landry and S. Ingolia, *Ramp Safety Practices*, vol. 29. Washington, DC, USA: Transportation Research Board, 2011.
- [3] G. Pestana, T. R. da Silva, and P. Reis, "Handling airport ground operations using an A-SMGCS approach," in *Proc. Aerosp. Conf.*, Mar. 2011, pp. 1–15.
- [4] P. G. Ansola, A. G. Higuera, F. J. Otamendi, and J. de Las Morenas, "Agent-based distributed control for improving complex resource scheduling: Application to airport ground handling operations," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1145–1157, Dec. 2014.
- [5] N. Bubner, N. Bubner, R. Helbig, and M. Jeske, "Logistics trend radar, delivering insight today," *Creating Value Tomorrow*, 2014. [Online]. Available: <https://www.dhl.com/content/dam/dhl/global/core/documents/pdf/global-core-logistics-trend-radar-5thedition.pdf>
- [6] I. Kovynyov and R. Mikut, "Digital technologies in airport ground operations," *NETNOMICS: Econ. Res. Electron. Netw.*, vol. 20, no. 1, pp. 1–30, Apr. 2019.
- [7] D. A. Tabares and F. Mora-Camino, "Aircraft ground handling: Analysis for automation," in *Proc. 17th AIAA Aviation Technol., Integr., Oper. Conf.*, Jun. 2017, p. 3425.
- [8] CAAS. (2018). *Raising Airport Capabilities*. Accessed: Nov. 3, 2019. [Online]. Available: <https://www.caas.gov.sg/who-we-are/areas-of-responsibility/developing-the-industry/raising-airport-capabilities>
- [9] Wikimedia. (2015). *Dolly for ULD and Dollies for Loose Luggages*. Accessed: Oct. 10, 2019. [Online]. Available: [https://commons.wikimedia.org/wiki/File:Dolly\\_for\\_ULD\\_and\\_dollies\\_for\\_loose\\_luggages.JPG](https://commons.wikimedia.org/wiki/File:Dolly_for_ULD_and_dollies_for_loose_luggages.JPG)
- [10] J. Haverinen and A. Kempainen, "A geomagnetic field based positioning technique for underground mines," in *Proc. IEEE Int. Symp. Robot. Sensors Environ. (ROSE)*, Sep. 2011, pp. 7–12.
- [11] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, Winter 2010.
- [12] T. D. Pickett, N. W. Anderson, and B. E. Romig, "Geometry-based monitoring and control of coupled mobile machines," U.S. Patent 9904290, Feb. 27, 2018.
- [13] R. L. Nelson, Jr., "System and method for determining implement train position," U.S. Patent 9380738, Jul. 5, 2016.
- [14] P. Henkel and A. Sperl, "Real-time kinematic positioning for unmanned air vehicles," in *Proc. IEEE Aerosp. Conf.*, Mar. 2016, pp. 1–7.
- [15] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.
- [16] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking: III. Measurement models," *Proc. SPIE*, vol. 4473, pp. 423–446, Nov. 2001.
- [17] Z. Sahinoglu and S. Gezici, "Ranging in the IEEE 802.15.4a Standard," in *Proc. IEEE Annu. Wireless Microw. Technol. Conf.*, Dec. 2006, pp. 1–5.
- [18] Z. Chen, Q. Zhu, and Y. C. Soh, "Smartphone inertial sensor-based indoor localization and tracking with iBeacon corrections," *IEEE Trans. Ind. Informat.*, vol. 12, no. 4, pp. 1540–1549, Aug. 2016.
- [19] P. Spachos and K. N. Plataniotis, "BLE beacons for indoor positioning at an interactive IoT-based smart museum," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3483–3493, Sep. 2020.
- [20] T. E. Meyer, "Grid, ground, and globe: Distances in the GPS era," *Surveying Land Inf. Sci.*, vol. 62, no. 3, pp. 179–202, 2002.
- [21] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, Jan. 1998.
- [22] D. Simon, *Optimal State Estimation: Kalman, H<sub>∞</sub>, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.
- [23] R. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-6, no. 4, pp. 473–483, Jul. 1970.
- [24] P. Jiménez, F. Thomas, and C. Torras, "3D collision detection: A survey," *Comput. Graph.*, vol. 25, no. 2, pp. 269–285, Apr. 2001.



(especially optimal estimation theory) and control technology.

**Shixiong Wang** (Graduate Student Member, IEEE) received the B.Eng. degree in detection, guidance and control technology, and the M.Eng. degree in systems and control engineering from the School of Electronics and Information, Northwestern Polytechnical University, China, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. with the Department of Industrial Systems Engineering and Management, National University of Singapore. His research interest includes statistics and optimization theories with applications in signal processing



cover hierarchical learning, forecasting, data mining, and machine learning.

**Chongshou Li** received the B.Eng. degree from Northwestern Polytechnical University, China, in 2010, and the Ph.D. degree from the City University of Hong Kong, in 2010, respectively. He is currently an Associate Professor with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. He was a Research Assistant Professor with the National University of Singapore from 2017 to 2021, and a Research Scientist with Nanyang Technological University from 2015 to 2016. His research areas



as operations research, management science, and transportation science. He is one of the five Returning Singaporean Scientists and a recipient of China 1000 Talents Scheme.

**Andrew Lim** received the B.Sc. and Ph.D. degrees in computer and information sciences from the University of Minnesota, Minneapolis, MN, USA, in 1987 and 1992, respectively. He is currently a Professor with the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore. His research interest includes big data analytics, demand generation, and supply management problems in the domains of logistics, transportation, and healthcare. Many of his works have been published in leading journals, such