LiDAR-Inertial based Localization and Perception for Indoor Pursuit-Evasion Differential Games

Haowen Lai, Wenhao Liang, Rui Yan, Zongying Shi, Yisheng Zhong
Department of Automation, Tsinghua University, Beijing 100084, P. R. China
E-mail: {lhw19,liangwh17,yr15}@mails.tsinghua.edu.cn and {szy,zys-dau}@mail.tsinghua.edu.cn

Abstract: Pursuit-evasion (PE) differential games are widely studied due to their enormous application prospects in robotics. Most of current works, however, focus on the theory and simulation, while even for some experiments the global information is directly acquired by external devices or just by sharing, which is far from the real scenarios. Motivated by this limitation, we propose a LiDAR-inertial based localization and perception method to estimate robot states such as position, pose and velocity required by high-level strategies. The LiDAR-inertial odometry (LIO) that utilizes data from 3D LiDAR and IMU provides self localization in an unknown environment. To locate other robots, a curve fitting based algorithm for 3D LiDAR point clouds is proposed, after which a sliding window average method is adopted to filter the noise in the perception results. Except for 3D LiDAR and IMU, no other sensors or communication is needed. Experiments are presented to illustrate the results.

Key Words: localization, robot perception, LiDAR-inertial odometry, pursuit-evasion differential games

1 Introduction

Differential game theory provides a proper framework to analyze pursuit-evasion (PE) problems [1–4]. PE games are representative of a huge number of important but challenging problems in robotics, aircraft control, reachability analysis and security [5–7]. An attractive class among them is reach-avoid games which have received a lot of attentions recently [8–11]. In these games, *attackers* or *evaders*, attempt to reach a goal region, and *defenders* or *pursuers*, strive to intercept the former before they reach the goal. Such games can be used to analyze many adversarial problems, such as oil pipelines protection, border or coastline protection and collision avoidance.

Most of the current studies [12–14] focused on theory and simulation under the full state information, without the support of real robot experiments. The key points for such experiments are the estimation of robot states, such as position, pose and velocity, of the decision-making robot itself, and the acquisition of those states of others. Biswas et al. [15] used external cameras attached to the ceiling to locate robots with markers on the top, giving states of all the robots. Then each individual could have access to the state information it needed. In a real application scene, it is not reasonable to expect the existence of such external observing devices, or just directly acquire them from the opponent team by sharing. Hence, the capability of localization for itself and perception for others is of great significance for robots in PE differential games.

3D LiDAR has been a popular sensor in the field of self localization due to its ability to provide wide-range geometric information [16], accurate measurement [17] and robust results in light-varying environment [18]. There are mainly two types of localization methods, which respectively base on map matching and online odometry. Map matching based localization methods estimate robot states by comparing current LiDAR data with a pre-built map model. Normal Distribution Transform (NDT) [19] considered the map as a probabilistic generative model. It first divided the map into small cubes, and then computed the mean vector and covariance matrix of each cube assuming that it satisfied some normal distribution. The localization result was derived from the optimization that finding a pose to maximize the probability density function. Wolcott et al. [20] represented the map with cells in a 2D grid over xy direction. The distribution of the z-height in each cell was modeled with a 1D Gaussian Mixture Model (GMM). Then a multi-resolution branch-and-bound searching method was utilized to find the best transformation.

Although maps can be regarded as some sort of priori, it is common that PE differential games start at unknown environment, where pre-built maps are unavailable. Odometry based methods, which are widely used in Simultaneous Localization and Mapping (SLAM), compare adjacent LiDAR frames to track the position and pose of the robot. ICP [21] adopts an iterative framework, but it is vulnerable to the initial transformation. Recently researchers have introduced inertial measurement unit (IMU) to LiDAR odometry so as to lower drift and improve accuracy. The resulting LiDAR-inertial odometry (LIO) can be categorized into two types [22], namely loosely-coupled and tightly-coupled fusion. In loosely-coupled LIO, such as LOAM [23] and LeGo-LOAM [24], robot state estimation from IMU through integration is utilized to provide LiDAR odometry with initial transformation as priori. However, IMU works independently and its bias is not estimated in the optimization process. To overcome this issue, tightly-coupled LIO [22, 25] is improved by jointly optimizing IMU bias and LiDAR odometry, resulting in more accurate localization.

The capability of perception for a robot is fundamental and crucial, since it is more practical and closer to the real application scene. 2D LiDAR was used in [26, 27] to perform obstacle detection and avoidance. In [26], convex hull algorithm was utilized to represent the boundaries of obstacles, while in [27], some simple geometric shapes were selected to fit obstacles. Yan et al. [16] proposed a multiple sensors perception pipeline mainly for the detection and tracking of human. Several kinds of feature were extracted from 3D LiDAR point clouds and trained with SVM.

This work was supported by the Guoqiang Research Institute of Tsinghua University.

This paper presents a LiDAR-inertial based localization and perception method for robots in indoor PE differential games. The contributions of this paper are threefold:

- First, we present a circle fitting based target perception algorithm using only 3D LiDAR point clouds. The position and velocity of the target are estimated at the same rate as the frame rate of the LiDAR.
- Second, a LiDAR-inertial based localization and perception framework for robots is proposed. The 3D LiDAR and IMU are fused and then utilized for self localization. Only LiDAR data are used for target perception. No other sensors are required.
- Third, we conduct experiments on PE differential games in which the robot obtains the values of states by self localization and target perception.

The rest of this paper is organized as follows. The problem is stated in Section 2. Section 3 describes the localization and perception method. Then, experiments are presented in Section 4. Finally, Section 5 concludes the paper.

2 Problem Statement

Consider a PE differential game in the plane $\mathbb{R}^2$ between a pursuer P and an evader E, where $\mathbb{R}$ is the set of reals. The dynamics of the two players are described by the following equations:

$$\dot{\boldsymbol{x}}_P = v_P \boldsymbol{u}_P, \qquad \boldsymbol{x}_P(0) = \boldsymbol{x}_{P0},$$
$$\dot{\boldsymbol{x}}_E = v_E \boldsymbol{u}_E, \qquad \boldsymbol{x}_E(0) = \boldsymbol{x}_{E0},$$

where $\boldsymbol{x}_P \in \mathbb{R}^2$, $\boldsymbol{x}_{P0} \in \mathbb{R}^2$ and $v_P > 0$ are the current position, initial position and speed of P respectively, and $\boldsymbol{x}_E \in \mathbb{R}^2$, $\boldsymbol{x}_{E0} \in \mathbb{R}^2$ and $v_E > 0$ are the current position, initial position and speed of E respectively. The control inputs of P and E are the headings $\boldsymbol{u}_P \in \mathbb{S}^1$ and $\boldsymbol{u}_E \in \mathbb{S}^1$ respectively, where $\mathbb{S}^1 = \{\boldsymbol{u} \in \mathbb{R}^2 \,|\, \|\boldsymbol{u}\|_2 \leq 1\}$.

The plane is divided into a goal region Ω_{goal} and a play region Ω_{play} by a straight line $\mathcal{T}$. The evader wins the game if it can reach the goal region Ω_{goal} from the play region Ω_{play} before being captured, while the pursuer wins the game by capturing the evader in the play region Ω_{play}. The evader is captured by the pursuer if $\|\boldsymbol{x}_P - \boldsymbol{x}_E\|_2 \leq r$, where $r \geq 0$ is the capture radius. Denote the speed ratio by $\alpha = v_P / v_E$. We here consider the case $\alpha > 1$ and take

$$\Omega_{\text{goal}} = \{\boldsymbol{x} \in \mathbb{R}^2 \,|\, y \leq 0\}, \;\; \Omega_{\text{play}} = \{\boldsymbol{x} \in \mathbb{R}^2 \,|\, y > 0\},$$
$$\mathcal{T} = \{\boldsymbol{x} \in \mathbb{R}^2 \,|\, y = 0\},$$

where $\boldsymbol{x} = [x, y]^\top$.

This paper studies a LiDAR-inertial based localization and perception method for indoor PE differential games. Suppose that two robots play in a bounded and obstacle-free indoor environment. We focus on the pursuer, and the method can also be extended for the evader easily. Thus, we refer to the pursuer as *source* robot (denoted by S) and the evader as *target* robot (denoted by T) hereinafter.

The works [1, 9, 10] show that some states of the source and the target are required for pursuit strategies. We here assume that the target adopts a given strategy unknown to the source, e.g., stop strategy or random strategy, and the source adopts the feedback strategy in [9]:

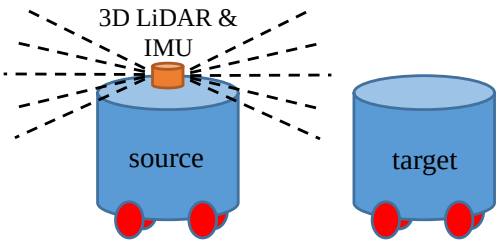


Fig. 1: The localization and perception scheme for pursuit-evasion (PE) games.

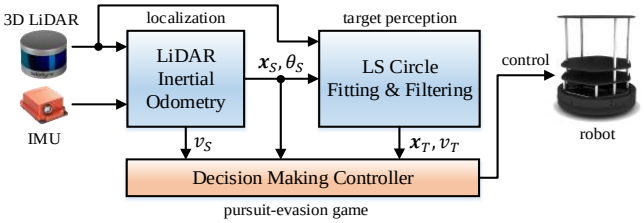


Fig. 2: The block diagram of the whole system.

$$\boldsymbol{u}_S = \frac{\boldsymbol{x}_I - \boldsymbol{x}_S}{\|\boldsymbol{x}_I - \boldsymbol{x}_S\|_2} \tag{1}$$

where $\boldsymbol{x}_I \in \mathbb{R}^2$ is computed by α, $\boldsymbol{x}_S$ and $\boldsymbol{x}_T$.

Note that v_S, v_T, $\boldsymbol{x}_S$ and $\boldsymbol{x}_T$ are necessary to compute the source control input $\boldsymbol{u}_S$. In [9], all these values are assumed to be known to S at any time. However, in practice, the above values are supposed to be estimated by the source itself, without any other assistance like external positioning devices or information sharing. In particular, the target has no incentive to share its own information to the source due to the adversarial settings.

3 Localization and Perception

In this section, the localization and perception for robots are described. First, we introduce the framework of our system and illustrate the methodology. Then, a LiDAR-inertial odometry is used to locate the source itself, and a circle fitting algorithm is proposed to perceive the target. Finally, a smoothing method is used to filter out the noise in the perception results.

3.1 System Overview

As mentioned in the previous section, the source robot needs to independently estimate the states of both itself and the target. To handle this issue, a localization and perception scheme is proposed. As shown in Fig. 1, a 3D LiDAR is mounted on top of the source robot, and an IMU is provided for inertial measurement. The data from the two sensors are fused for self localization, while the dense point clouds from the 3D LiDAR are also used for the target perception. No other sensors are introduced to the system.

The block diagram of the system is shown in Fig. 2. The LIO fuses data from 3D LiDAR and IMU to estimate the position $\boldsymbol{x}_S$, heading angle θ_S and speed v_S of the source. The least square (LS) method is used to improve the accuracy when we compute the relative position between the source and the target by LiDAR point clouds. To obtain the global

position $\boldsymbol{x}_T$ and speed v_T of the target, states $\boldsymbol{x}_S$ and θ_S of the source are also needed. Furthermore, a smoothing method is applied so as to filter out the noise in the estimated $\boldsymbol{x}_T$ and v_T. When these values are ready, the strategy (1) is adopted by the source robot.

3.2 LiDAR-Inertial Odometry for Self Localization

The capability of accurate self localization in an unknown environment is of vital importance to the source robot. As Fig. 2 shows, the target perception module also relies on the self localization results, meaning that errors generated in this part will be passed to the downstream perception.

In the indoor environment, global navigation satellite systems (GNSS) like GPS are usually unavailable. Moreover, map matching based localization methods become invalid in an unknown area due to the lack of prior maps. These issues can be overcome by odometry based methods. Tightly-coupled LIO has been recognized with several advantages such as accurate localization, low drift and real-time processing in the complex environments. Hence, this odometry is suitable for the source robot to locate itself in an indoor PE differential game.

Tightly-coupled LIO is composed of two sensors: IMU and 3D LiDAR. IMU can output linear acceleration and angular velocity at a relatively high rate, which can be modeled as [25]:

$$\begin{aligned} \tilde{\mathbf{a}}_t &= \mathbf{q}_t^{\mathrm{BW}}(\mathbf{a}_t - \mathbf{g}) + \mathbf{b}_t^{\mathrm{a}} + \mathbf{n}_t^{\mathrm{a}}, \\ \tilde{\boldsymbol{\omega}}_t &= \boldsymbol{\omega}_t + \mathbf{b}_t^{\omega} + \mathbf{n}_t^{\omega}, \end{aligned} \tag{2}$$

where $\tilde{\mathbf{a}}_t$ and $\tilde{\boldsymbol{\omega}}_t$ are respectively the measurements of linear acceleration and angular velocity in the IMU body frame (denoted by B) at time t, while $\mathbf{a}_t$ and $\boldsymbol{\omega}_t$ are the actual values in the world frame (denoted by W). $\mathbf{b}_t$ is the bias that slowly varies over time, and $\mathbf{n}_t$ is the noise with normal distribution. $\mathbf{q}_t^{\mathrm{BW}}$ is the rotation quaternion from W to B, and $\mathbf{g}$ represents the gravity vector in the world frame.

By integrating $\tilde{\mathbf{a}}_t$ and $\tilde{\boldsymbol{\omega}}_t$ from (2), IMU can be used to predict the robot motion roughly, which serves as an initial transformation for LiDAR scan matching. Besides, the prediction is used for the de-skew of point clouds that distorted by motion, leading to a more accurate LiDAR scan matching result. To ease the computation burden when $\mathbf{b}_t$ is updated, the IMU preintegration technique is also applied.

LiDAR scan matching is based on point cloud features. Two types of features, namely edge feature and planar feature, are extracted based on the roughness of each point using the method in [23]. The matching procedure is similar to [25], but we process every frame instead of key frames, aiming to improve the output rate of the LIO.

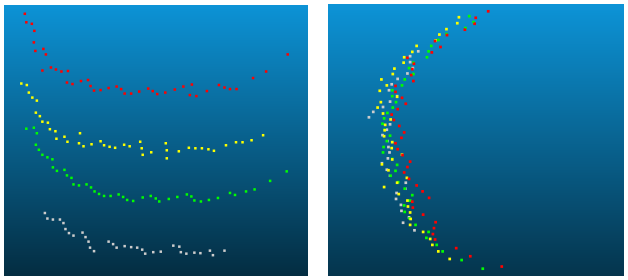
After jointly optimizing the LiDAR scan matching and the bias of IMU, the LIO outputs a 6-DoF transformation

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3), \tag{3}$$

where $\mathbf{R} \in SO(3)$ is the rotation matrix, and $\mathbf{t} \in \mathbb{R}^3$ is the translation vector. Since our robots play in a plane, the roll and pitch of $\mathbf{R}$ and z-height of $\mathbf{t}$ is omitted, which yields

$$\boldsymbol{x}_S = \begin{bmatrix} \mathbf{I}_2 & \\ & 0 \end{bmatrix} \mathbf{t}, \quad \theta_S = \mathrm{yaw}(\mathbf{R}),$$

where $\mathrm{yaw}(\cdot)$ computes the yaw angle from rotation matrix.



(a) Circles with different heights (b) Vertical view

Fig. 3: LiDAR point cloud of the target's surface. Points with different heights are illustrated with different colors.

3.3 Target Perception

The perception module aims to estimate the position $\boldsymbol{x}_T$ and speed v_T of the target from LiDAR data. Since the shape of the target affects its appearance in the point cloud, a specific model in the perception should be chosen accordingly. As shown in Fig. 3, the point cloud of the target's surface can be approximately viewed as a part of a circle with some noise. So, in order to estimate $\boldsymbol{x}_T$ and v_T, a general circle model is chosen to represent the target:

$$f(x, y) = x^2 + y^2 + ax + by + c = 0,$$

where $a \in \mathbb{R}$, $b \in \mathbb{R}$, and $c \in \mathbb{R}$ are the coefficients to be determined. To fit that, the ordinary least square method is utilized. For a LiDAR frame $\mathbb{F}$, the loss function is defined as follow:

$$\mathcal{L}(a, b, c) = \sum_{j=1}^{M} f^2(x_j, y_j),$$

where M is the total number of points on the target's surface, and (x_j, y_j) indicates the j-th point. Then, the fitting can be described as a convex optimization problem:

$$\min_{a,b,c} \ \mathcal{L}(a, b, c). \tag{4}$$

Denote that $\boldsymbol{a}_j = [x_j, y_j, 1]^\top$, $\mathbf{A} = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_M]^\top$, and $\mathbf{b} = \left[x_1^2 + y_1^2, \cdots, x_M^2 + y_M^2\right]^\top$. Then, the solution to (4) is

$$[a, b, c]^\top = -\left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b}. \tag{5}$$

Since $M \gg 3$ and $\mathbf{A}$ usually has full column rank in real cases, the inverse of $\mathbf{A}^\top \mathbf{A}$ exists. By (5), the relative position of the target with respect to the source $\boldsymbol{x}_{ST}$ and the target's radius r can be derived as follows:

$$\boldsymbol{x}_{ST} = \left[-\frac{a}{2}, -\frac{b}{2}\right]^\top, \quad r = \sqrt{\|\boldsymbol{x}_{ST}\|_2^2 - c}.$$

Note that $\boldsymbol{x}_{ST}$ is a position originated at the source robot. To get the target's global position $\boldsymbol{x}_T$, a transformation from the origin of the global frame to the source robot should be applied, i.e., the transformation $\mathbf{T}$ in (3):

$$\boldsymbol{x}_T = \mathbf{T}\boldsymbol{x}_{ST}.$$

Then, the target's speed can be derived using discrete approximation:

$$v_T^{(i)} = \left\| \frac{1}{\Delta t_i} \left(\boldsymbol{x}_T^{(i)} - \boldsymbol{x}_T^{(i-1)} \right) \right\|_2,$$

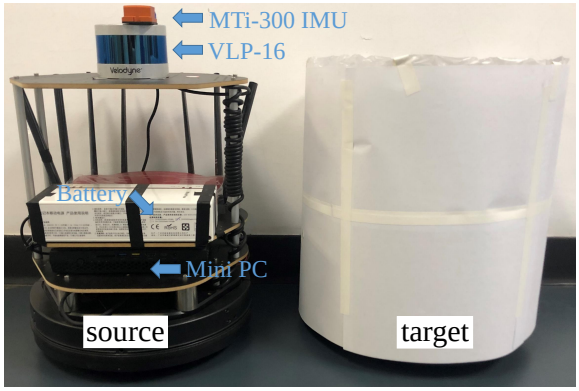


Fig. 4: The experimental platform. The target robot is surrounded with a piece of paper for a better reflection of laser.

where the superscript i belongs to positive integers that indicate the state estimated in the i-th LiDAR frame $\mathbb{F}_i$, and Δt_i is the time interval between $\mathbb{F}_{i-1}$ and $\mathbb{F}_i$.

3.4 Smoothing and Filtering

Due to the noise in LiDAR point clouds and the fact that the surface of the target robot is not a perfect circle, the target's position $\boldsymbol{x}_T$ may be different from the true value. Slight though the error is, it may has a critical impact on the estimation of the speed v_T, since the distance of the target moving between two consecutive LiDAR frames is small.

In this part, a sliding window average (SWA) method is proposed to smooth the position $\boldsymbol{x}_T$ and filter out noise:

$$\tilde{\boldsymbol{x}}_T^{(i)} = \begin{cases} \frac{1}{N}\sum_{j=i-N+1}^{i} \boldsymbol{x}_T^{(j)}, & i \geq N \\ \boldsymbol{x}_T^{(i)}, & 1 \leq i < N \end{cases}$$

where N is the number of states used in the SWA. The smoothed position results in a more stable and accurate estimation of speed, which can be computed by

$$\tilde{v}_T^{(i)} = \left\| \frac{1}{\Delta t_i} \left(\tilde{\boldsymbol{x}}_T^{(i)} - \tilde{\boldsymbol{x}}_T^{(i-1)} \right) \right\|_2 .$$

4 Experimental Results

In this section, experimental results are presented. First, the experimental platform is introduced. Then, the performance of our localization and perception pipeline is given. Finally, we conduct experiments on PE differential games in which the source robot acquires the states by self localization and target perception.

4.1 Platform and Setup

All experiments are conducted on two TurtleBot2 robots, each of which is a differential drive robot equipped with a Kobuki base shown in Fig. 4. We replace the original Kinect sensor with a Velodyne Lidar's Puck (VLP-16) and an Xsens interial sensor (MTi-300 IMU).

All algorithms, including self localization, target perception, decision making and robot control, are implemented in C++ and run on an Intel NUC8i7HVK Mini PC installed on the source robot. Robot Operating System (ROS) is chosen as the basic software framework, and Point Cloud Library (PCL) is used for point cloud processing. Additionally, part of the localization module is adapted from [25], an SLAM framework using tightly-coupled LiDAR-inertial odometry.

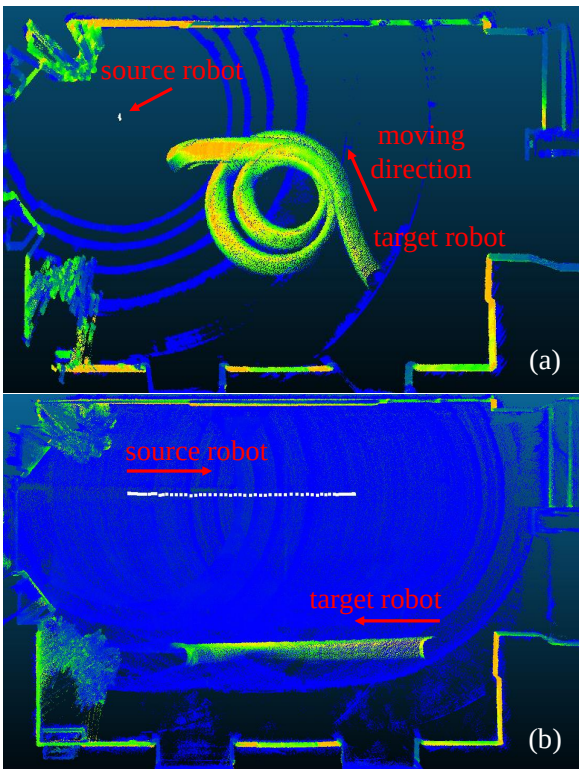


Fig. 5: Self localization with LiDAR-inertial odometry. The position of the source robot is in white dots. (a) The source robot is stationary. (b) The source robot moves along a straight line.

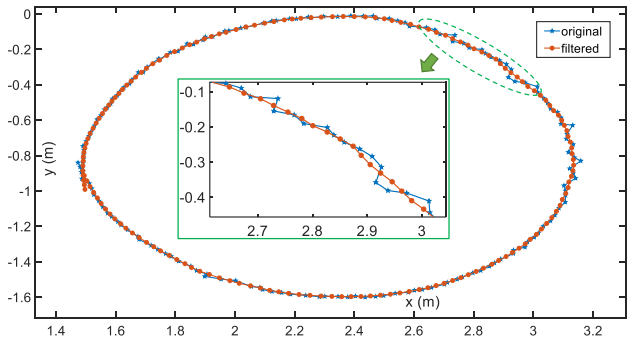
4.2 Localization and Perception

The proposed LiDAR-inertial based localization and perception method is tested in an indoor unknown environment to verify its effectiveness and accuracy. No other prior information is used except the boundary of play region and the height and shape of the target robot.

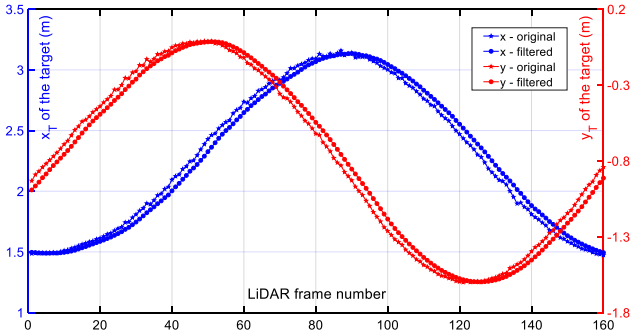
The results of self localization are shown in Fig. 5, which is an overlap of multiple LiDAR frames. The thin walls and clear edges reveal that the self localization is highly accurate, because each frame is first transformed with respect to $\mathbf{T}$ of the source robot and then added together directly. There is a total number of 501 frames in Fig. 5 (a), while Fig. 5 (b) contains 318 frames. Note that whether the source robot is stationary or not has little effect on the accuracy of self localization, making it more robust to different strategies.

To test the perception module as well as the smoothing and filtering method, experiments are conducted when both two robots are moving. Fig. 6 shows the estimation of target's position, while Fig. 7 shows that of target's velocity. In this case, the target robot moves at a constant linear speed of 0.32 m/s and an angular velocity of 0.40 rad/s, whose true trajectory is a circle. The filtering parameter N is set to 6.

As shown in Fig. 6(a), the perception results without filtration fit the target's trajectory well, except for some noise in the zoomed upper right part. This is because the surface of the target robot is not a perfect circle, and the LiDAR point cloud corresponds to different part of this surface while both robots are moving. The proposed smoothing method filters out the noise of the perception results effectively, which can

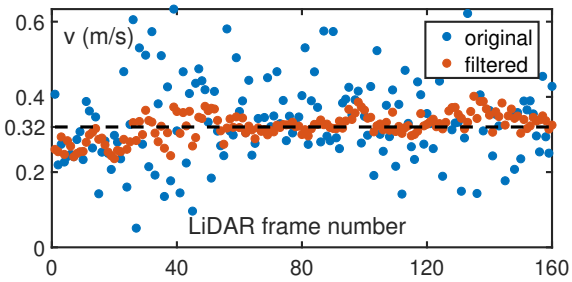


(a) Target's trajectory in the xy-plane.

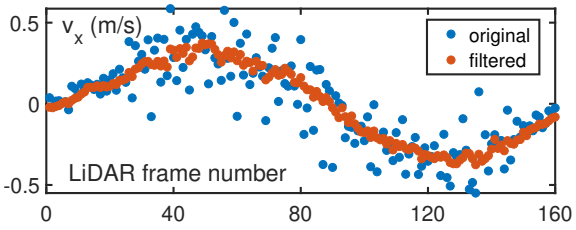


(b) Target's position in the x and y coordinates.

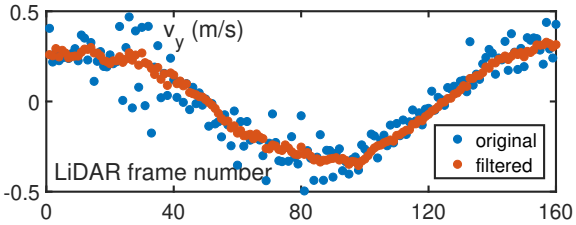
Fig. 6: The estimation of the target's position, where the original position $\boldsymbol{x}_T$ and the filtered position $\tilde{\boldsymbol{x}}_T$ are both plotted.



(a) The target's speed (i.e. the magnitude of its velocity).



(b) The target's velocity in the x coordinate.



(c) The target's velocity in the y coordinate.

Fig. 7: The estimated velocity of the target using the estimated position. Both the original velocity $\boldsymbol{v}_T$ (estimated using $\boldsymbol{x}_T$) and the filtered velocity $\tilde{\boldsymbol{v}}_T$ (estimated using $\tilde{\boldsymbol{x}}_T$) are shown in blue and red dots, respectively.

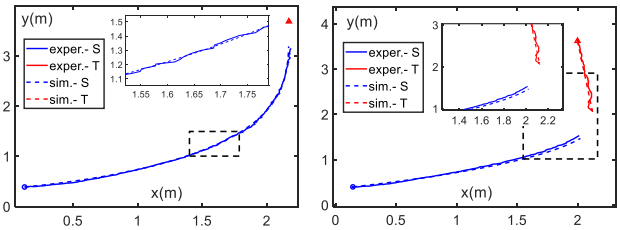
also be observed in Fig. 6(b). Compared to the original position, the filtered one is smoother and more suitable for the decision-making module and the robot controller.

The estimated velocity of the target is shown in Fig. 7, in which the effect of smoothing and filtering can be observed. Small noise in the position will be amplified after difference operation, which may cause a deadly impact on the decision-making module and the robot controller. The filtered velocity is smoother and closer to the constant speed $v_T = 0.32$ m/s we set in the experiment, indicated by the black dash line in Fig. 7(a).

4.3 PE Differential Games

To apply the proposed localization and perception method to PE differential games, simulation and experiments on real robots are conducted. Trajectories of the source and the target are shown in Fig. 8. In this game, the source acquires all required states of itself and the target by the proposed method, and adopts the feedback strategy (1).

Two different strategies are considered for the target, i.e. the target is stationary as in Fig. 8(a) and moves randomly as in Fig. 8(b). In both cases, the speed of the source robot v_S is set to 0.25 m/s. The results show that the experimental trajectories accurately follow the simulation ones. The estimation error of the target's position becomes larger when two robots are very close to each other, because in this case there is little laser on the target's surface.



(a) The target is stationary. (b) The target moves randomly.

Fig. 8: Trajectories of the source (shown in blue line) and the target (shown in red line) in a PE differential game. The initial positions of them are denoted by a blue circle and a red triangle, respectively. Simulation trajectories are shown in dash lines and experimental ones are shown in solid lines.

5 Conclusion

This paper proposes a LiDAR-inertial based localization and perception method for indoor PE differential games. The real-time states of the source and the target robots, including position, heading angle and speed, are estimated by data from a 3D LiDAR and an IMU. Our method has been tested and verified in a real indoor PE differential game.

For future work, outdoor experiments will be performed. Moreover, the drift in odometry based localization methods increases after a long run. We intend to reduce the drift by combining the odometry with some priori information about the environment, such as maps and visual markers.

References

[1] R. Yan, Z. Shi, and Y. Zhong, "Reach-avoid games with two defenders and one attacker: An analytical approach," *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1035–1046, 2019.

[2] E. Garcia, D. W. Casbeer, and M. Pachter, "Optimal strategies for a class of multi-player reach-avoid differential games in 3D space," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4257–4264, 2020.

[3] D. Shishika and V. Kumar, "A review of multi agent perimeter defense games," in *Decision and Game Theory for Security*, (Cham), pp. 472–485, Springer International Publishing, 2020.

[4] R. Yan, X. Duan, Z. Shi, Y. Zhong, and F. Bullo, "Matching-based capture strategies for 3D heterogeneous multiplayer reach-avoid differential games," *arXiv preprint:1909.11881*, 2019.

[5] X. Dong, Y. Hua, Y. Zhou, Z. Ren, and Y. Zhong, "Theory and experiment on formation-containment control of multiple multirotor unmanned aerial vehicle systems," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 1, pp. 229–240, 2019.

[6] H. Liu, Y. Wang, F. L. Lewis, and K. P. Valavanis, "Robust formation tracking control for multiple quadrotors subject to switching topologies," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 3, pp. 1319–1329, 2020.

[7] R. Yan, Z. Shi, and Y. Zhong, "Defense game in a circular region," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 5590–5595, 2017.

[8] R. Yan, Z. Shi, and Y. Zhong, "Task assignment for multiplayer reach–avoid games in convex domains via analytical barriers," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 107–124, 2020.

[9] R. Yan, Z. Shi, and Y. Zhong, "Optimal strategies for the lifeline differential game with limited lifetime," *International Journal of Control*, vol. 0, no. 0, pp. 1–14, 2019.

[10] R. Yan, Z. Shi, and Y. Zhong, "Cooperative strategies for two-evader-one-pursuer reach-avoid differential games," *International Journal of Systems Science*, vol. 0, no. 0, pp. 1–19, 2021.

[11] R. Yan, Z. Shi, and Y. Zhong, "Guarding a subspace in high-dimensional space with two defenders and one attacker," *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.

[12] M. Pachter, E. García, and D. Casbeer, "Differential game of guarding a target," *Journal of Guidance, Control, and Dynamics*, vol. 40, pp. 1–8, 06 2017.

[13] Z. Zhou, J. Ding, H. Huang, R. Takei, and C. Tomlin, "Efficient path planning algorithms in reach-avoid problems," *Automatica*, vol. 89, pp. 28–36, 2018.

[14] J. Selvakumar and E. Bakolas, "Feedback strategies for a reach-avoid game with a single evader and multiple pursuers," *IEEE Transactions on Cybernetics*, vol. 51, no. 2, pp. 696–707, 2019.

[15] J. Biswas, J. P. Mendoza, D. Zhu, B. Choi, S. Klee, and M. Veloso, "Opponent-driven planning and execution for pass, attack, and defense in a multi-robot soccer team," in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '14, p. 493–500, 2014.

[16] Z. Yan, S. Schreiberhuber, G. Halmetschlager, T. Duckett, and N. Bellotto, "Robot perception of static and dynamic objects with an autonomous floor scrubber," *Intelligent Service Robotics*, vol. 13, no. 1, pp. 403–417, 2020.

[17] V. Rosas-Cervantes and S. G. Lee, "3d localization of a mobile robot by using monte carlo algorithm and 2d features of 3d point cloud," *International Journal of Control Automation and Systems*, vol. 18, no. 2, pp. 2955–2965, 2020.

[18] B. Zhou, Z. Tang, K. Qian, F. Fang, and X. Ma, "A lidar odometry for outdoor mobile robots using ndt based scan matching in gps-denied environments," in *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 1230–1235, 2017.

[19] M. Magnusson, *The Three-Dimensional Normal-Distributions Transform — an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. PhD thesis, 12 2009.

[20] R. W. Wolcott and R. M. Eustice, "Fast lidar localization using multiresolution gaussian mixture maps," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2814–2821, 2015.

[21] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[22] W. Dai, B. Tian, and H. Chen, "Tightly-coupled lidar-inertial odometry and mapping in real time," in *2020 39th Chinese Control Conference (CCC)*, pp. 3258–3263, 2020.

[23] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems Conference*, 07 2014.

[24] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758–4765, 2018.

[25] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5135–5142, 2020.

[26] D. Ghorpade, A. D. Thakare, and S. Doiphode, "Obstacle detection and avoidance algorithm for autonomous mobile robot using 2d lidar," in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 1–6, 2017.

[27] Y. Peng, D. Qu, Y. Zhong, S. Xie, J. Luo, and J. Gu, "The obstacle detection and obstacle avoidance algorithm based on 2-d lidar," in *2015 IEEE International Conference on Information and Automation*, pp. 1648–1653, 2015.