Probabilistic Trajectory Estimation based Leader Following for Multi-Robot Systems

Mao Shan, Ying Zou, Mingyang Guan, Changyun Wen, Kwang-Yong Lim, Cheng-Leong Ng, Paul Tan
ST Engineering-NTU Corporate Laboratory
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore 639798
Email: shanmao@ntu.edu.sg, zouy0011@e.ntu.edu.sg
{myguan, ecywen}@ntu.edu.sg, {limkwangyong, ngclg, paultan}@stengg.com

***Abstract*—The paper is concerned with the multi-robot leader-following problem in the presence of frequent dropouts in vision detection. In many scenarios, for instance a structured environment, it is inevitable to experience outage of vision detection due to reasons such as the target moving out of view, vision occlusion, motion blurring, etc. The paper proposes a Bayesian trajectory estimation based leader-following approach that can offer accurate path following given intermittent vision observations. The follower robot estimates the trajectory of the leader robot based on the noise-corrupted odometry information of both robots, and inter-robot relative observations based on detection of fiducial markers using an RGBD camera. A linear trajectory-following control method is employed to track a historical pose of the leader robot on the estimated trajectory. Results are obtained based on evaluating the proposed leader-following approach in tests with a zig-zag shaped trajectory and with a trajectory that contains sharp turns.**

I. Introduction

Multi-robot systems have been an active research topic in areas of navigation and control. In applications such as exploration, search and rescue, transportation, etc., a team of mobile robots are usually required to travel and accomplish specific tasks together in a kind of formation.

In a typical leader-follower robots configuration, the first leading robot in the platoon defines the trajectory of the whole group, while each of the rest robots is following its preceding robot. The first robot could be teleoperated, self-navigating, following a human operator, or moving along a predetermined path. Consider the leader-following in convoy applications, autonomous navigation task is left to the first robot only, while each follower robot only performs relative localisation with respect to the preceding robot using comparatively simpler sensing and control techniques, which significantly reduce development and monetary costs.

As the robot platoon can be decomposed into chained pairs of leader and follower, as shown in Figure 1, for simplicity, the leader-following problem discussed in the remainder of the paper is defined within the scope of a such pair of robots, while the proposed approach could be equally implemented for each pair of consecutive robots.

The work was partially supported by ST Engineering-NTU Corporate Lab through NRF Corporate Laboratory@University scheme.

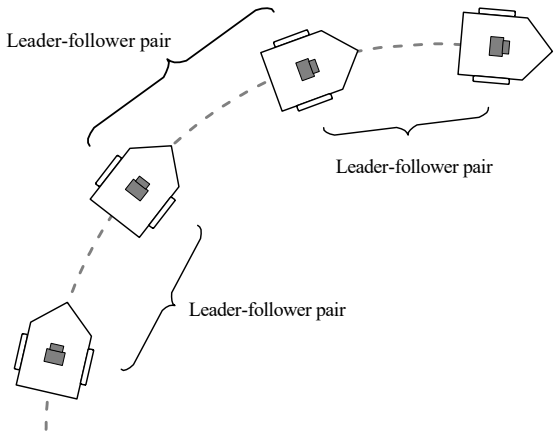


Fig. 1. A robot platoon decomposed into chained leader-follower pairs.

Most of existing leader-following approaches employ vision sensors, e.g., camera [1]–[3], laser [4], or a hybrid of them [5], to detect the leader as well as surrounding obstacles, as they provide accurate relative measurements to visual features. Vision sensors are however prone to occlusion by obstacles or walls. Despite significantly lower cost than laser sensors, cameras have more concerns raised from change of lighting condition, motion blurring, limited field-of-view and/or detection range, etc. A vision-based only approach proposed in [5] augments the camera's field-of-view by employing a mechanical servo system to keep the camera pointing to the leader, while the follower in [3] rotates and starts searching once the leader goes out of scene. However, vision occlusion is inevitable and only intermittent vision observations are available in many scenarios, e.g., a structured environment. Reference [4] tackles the observation dropout issue through fusion of odometry and vision information. The use of odometry data combined with robot kinematic model enables motion prediction during temporary outage periods of vision measurements.

The approaches in [1]–[5] only track the current pose of the leader, and are effective when the robots are moving close. In scenarios where the inter-robot spacing is larger, cutting corners on turns could happen if the tracking is performed without the knowledge of the leader robot's past poses. In

this case, the follower risks hitting an obstacle or wall, and it is required to replan an obstacle-avoiding path to recover tracking. There are alternative approaches that follow the leaders's trajectory, and thus without the cutting corners issue. Essentially, they are identical to tracking an estimated historical pose of the leader, either filtered or smoothed. The path-following strategy presented in [6] picks and tracks a simply "memorised" past position of the leader vehicle. Reference [7] proposes to follow the leader robot with a constant time delay. One of the shortcomings of the constant-time-delay concept is the unnecessary cloning of the leader's behaviour. For instance, if the leader was stopped by an obstacle a few seconds ago and now resumes moving, the follower will repeat the same action even if its way is not obstructed.

The paper proposes a probabilistic trajectory estimation based leader-following approach that can work with intermittent vision measurements. The robot pair can communicate with each other so that the odometry information of the leader robot, although corrupted with noise, is shared with the follower in real time. An RGBD camera mounted on the follower robot is able to measure the bearing and range (from image depth information) relative to the leader robot. A particle filter running in the follower robot estimates the trajectory of the leader robot within a sliding time window, based on the odometry data of the robot pair and intermittent vision observations. It is followed by a linear trajectory-following control algorithm to track a historical pose of the leader robot picked from the estimated trajectory. The use of trajectory estimation is advantageous as the estimate of the historical pose is smoothed with observation up to a later time.

The remainder of the paper is organised as follows. Section II presents Bayesian estimation formulation of the trajectory estimation and the particle filtering algorithm. The control method is elaborated in Section III. The evaluation results of the approach are presented and discussed in Section IV. Lastly, Section V concludes the paper.

II. Probabilistic Trajectory Estimation

A. Robot Kinematic Model

The pose of a robot at time step t could be denoted by a state vector describing its coordinates and heading information

$$\mathbf{x}_t = \begin{bmatrix} x_t & y_t & \theta_t \end{bmatrix}^T.$$

The kinematic model for a differential-wheel robot is written as

$$\begin{bmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} \cos(\theta_t) & 0 \\ \sin(\theta_t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_t \\ \omega_t \end{bmatrix}$$

where v_t and ω_t are the instantaneous linear driving velocity and steering angular velocity of the centre point of the robot at time step t, respectively. Given d as the distance along the axle between two wheel centres, they can be easily obtained from the linear velocities v_t^l and v_t^r of two driving wheels by

$$\begin{bmatrix} v_t \\ \omega_t \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{d} & \frac{1}{d} \end{bmatrix} \left(\begin{bmatrix} v_t^l \\ v_t^r \end{bmatrix} + \begin{bmatrix} w_t^l \\ w_t^r \end{bmatrix} \right)$$

where v_t^l and v_t^r are grouped into a wheel control vector $\mathbf{u}_t$, because they are more intuitive for controlling a differential-wheel robot; w_t^l and w_t^r are mutually independent random variables representing additive process noise to left and right wheels, respectively.

The process noise is introduced to incorporate dynamic nonidealities because in most of circumstances, the assumption that the control commands are ideally executed by the robot electromechanical system does not hold. These nonidealities include nonplanar surface, wheel slippage, gear backlash, encoder quantisation error, motor deadzone and saturation, to name a few. For this reason, the process noise is not necessarily Gaussian. The kinematic model can also be rewritten in the probabilistic form of $P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$.

B. Inter-Robot Relative Observations

Thanks to recent advances in computer vision, fiducial marker tracking libraries, e.g. ARToolkit [8], ArUco [9], ARTag [10], which have been extensively used in augmented reality, enable fast and accurate measurement of robot bearing in the use of an ordinary RGB camera. Many fiducial markers also provide unique identity information, which overcomes the data association problem. An RGBD camera is preferred over an RGB camera in the approach so as to additionally produce accurate range observations converted from image depth information.

These inter-robot bearing and range observations are conditional on current states of both robots, and are corrupted with their corresponding random observation noises. A probabilistic inter-robot observation model is written as

$$P\left(\mathbf{z}_t^{l\to f}|\mathbf{x}_t^l, \mathbf{x}_t^f\right)$$

where $\mathbf{z}_t^{l\to f}$ represents the vector of bearing and range measurements of the leader robot relative to the follower robot at time t; $\mathbf{x}_t^l$ and $\mathbf{x}_t^f$ denote the states of the leader and follower robots at t, respectively.

In the observation model, we define an effective detection area of fiducial markers in front of the camera by taking into account the field-of-view (FOV) of the camera as well as minimum and maximum detection ranges. When the markers are within the effective detection area, we also define a successful detection probability considering environment interference and motion blurring.

C. Bayesian Estimation

Define a sliding most-recent time window T_w, within which the trajectory of the leader robot is of interest for estimation by the follower robot. To achieve this, we keep the temporal-spatial joint states of the leader robot from a historical time step $k - T_w + 1$ up to the current time step k as well as the follower robot's state at k. Estimates are updated using relative observations between the leader and follower robots generated at time k. Recursively, we have the joint posterior inferred through two stages.

TABLE I
PARTICLE FILTERING ALGORITHM

$\{\mathbf{x}^{l,i}_{k-T_w+1:k}, \mathbf{x}^{f,i}_k, w^i_k\}^L_{i=1} \leftarrow$ *Particle_filter* $\left(\{\mathbf{x}^{l,i}_{k-T_w:k-1}, \mathbf{x}^{f,i}_{k-1}, w^i_{k-1}\}^L_{i=1}, \mathbf{u}^l_k, \mathbf{u}^f_k, \mathbf{z}^{l\to f}_k\right)$

draw $\{\mathbf{x}^{l,i}_k\}^L_{i=1} \sim \left\{P\left(\mathbf{x}^l_k|\mathbf{x}^{l,i}_{k-1}, \mathbf{u}^l_k\right)\right\}^L_{i=1}$ and $\{\mathbf{x}^{f,i}_k\}^L_{i=1} \sim \left\{P\left(\mathbf{x}^f_k|\mathbf{x}^{f,i}_{k-1}, \mathbf{u}^f_k\right)\right\}^L_{i=1}$
marginalisation: $\{\mathbf{x}^{l,i}_{k-T_w+1:k-1}, w^i_{k-1}\}^L_{i=1} \leftarrow \{\mathbf{x}^{l,i}_{k-T_w:k-1}, \mathbf{x}^{f,i}_{k-1}, w^i_{k-1}\}^L_{i=1} \setminus \{\mathbf{x}^{l,i}_{k-T_w}, \mathbf{x}^{f,i}_{k-1}\}^L_{i=1}$
augmentation: $\{\mathbf{x}^{l,i}_{k-T_w+1:k}, \mathbf{x}^{f,i}_k, w^i_{k-1}\}^L_{i=1} \leftarrow \{\mathbf{x}^{l,i}_{k-T_w+1:k-1}, w^i_{k-1}\}^L_{i=1} \bigcup \{\mathbf{x}^{l,i}_k, \mathbf{x}^{f,i}_k\}^L_{i=1}$
if $\mathbf{z}^{l\to f}_k$ exists **do**
  update weight with inter-robot observations $\{w^i_k\}^L_{i=1} \leftarrow \left\{P\left(\mathbf{z}^{l\to f}_k|\mathbf{x}^{l,i}_k, \mathbf{x}^{f,i}_k\right) w^i_{k-1}\right\}^L_{i=1}$
else do
  $\{w^i_k\}^L_{i=1} \leftarrow \{w^i_{k-1}\}^L_{i=1}$
end if
normalise weights $\{w^i_t\}^L_{i=1}$
if $\hat{N}_{eff} < N_{thr}$ **do**
  resample with replacement L particles from $\{\mathbf{x}^{l,i}_{k-T_w+1:k}, \mathbf{x}^{f,i}_k, w^i_k\}^L_{i=1}$ according to $\{w^i_k\}^L_{i=1}$
end if

The prediction stage for the leader-follower robot pair is written as

$$\begin{aligned} &P\left(\mathbf{x}^l_{k-T_w+1:k}, \mathbf{x}^f_k|\mathbf{Z}_{1:k-1}\right) \\ &= \iint P\left(\mathbf{x}^l_k|\mathbf{x}^l_{k-1}, \mathbf{u}^l_k\right) P\left(\mathbf{x}^f_k|\mathbf{x}^f_{k-1}, \mathbf{u}^f_k\right) \\ &\times P\left(\mathbf{x}^l_{k-T_w:k-1}, \mathbf{x}^f_{k-1}|\mathbf{Z}_{1:k-1}\right) d\mathbf{x}^l_{k-T_w} d\mathbf{x}^f_{k-1} \end{aligned} \tag{1}$$

where $P\left(\mathbf{x}^l_{k-T_w:k-1}, \mathbf{x}^f_{k-1}|\mathbf{Z}_{1:k-1}\right)$ is the prior distribution, $P\left(\mathbf{x}^l_k|\mathbf{x}^l_{k-1}, \mathbf{u}^l_k\right)$ represents the kinematic model of the leader robot given control vector $\mathbf{u}^l_k$ at time step k, while the model for the follower robot is denoted by $P\left(\mathbf{x}^f_k|\mathbf{x}^f_{k-1}, \mathbf{u}^f_k\right)$.

Since the trajectory estimation of the leader robot is carried out in the follower robot, the control vector $\mathbf{u}^l_k$ is assumed to be known to the follower robot in real time through wireless communication.

The update stage is written as

$$\begin{aligned} P\left(\mathbf{x}^l_{k-T_w+1:k}, \mathbf{x}^f_k|\mathbf{Z}_{1:k}\right) &\propto P\left(\mathbf{z}^{l\to f}_k|\mathbf{x}^l_k, \mathbf{x}^f_k\right) \\ &\times P\left(\mathbf{x}^l_{k-T_w+1:k}, \mathbf{x}^f_k|\mathbf{Z}_{1:k-1}\right). \end{aligned} \tag{2}$$

The advantage of keeping track of the joint state space is that estimates of historical states can be smoothed with "future" observations.

Since only relative observations are employed in the trajectory estimation process, the poses of the leader robot have to be translated to the local reference frame of the follower robot at time step k.

For $t = k - T_w + 1$ to k,

$$\mathbf{x}^{l,R}_t = R\left(\theta^f_k\right)\left(\mathbf{x}^l_t - \mathbf{x}^f_k\right)$$

where the orthogonal rotation matrix $R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

D. Particle Filtering Algorithm

Particle filtering is better suited to the tracking problem discussed in this paper over its parametric counterparts. Parametric filters, such as Kalman filter and its variants, are considered an ideal choice only in systems where Gaussian assumptions hold. In otherwise applications, such as the one investigated in the paper, the particle filter is able to deal with nonlinear and non-Gaussian components intrinsically, whilst the use of parametric filters inevitably brings linearisation and/or approximation errors in model parameterisation processes, which degrade their tracking performance significantly.

The particle filter running within the follower robot employs a collection of L particles to approximate $P\left(\mathbf{x}^l_{k-T_w+1:k}, \mathbf{x}^f_k|\mathbf{Z}_{1:k}\right)$. From the particles the trajectory of the leader robot is computed at each time step for the purpose of path-following control of the follower robot.

$$\{\mathbf{x}^{l,i}_{k-T_w+1:k}, \mathbf{x}^{f,i}_k, w^i_k\}^L_{i=1} \sim P\left(\mathbf{x}^l_{k-T_w+1:k}, \mathbf{x}^f_k|\mathbf{Z}_{1:k}\right)$$

The particle filtering algorithm is initialised to represent the initial state by drawing:

$$\{\mathbf{x}^{l,i}_0, \mathbf{x}^{f,i}_0, w^i_0\}^L_{i=1} \sim P\left(\mathbf{x}^l_0, \mathbf{x}^f_0\right).$$

The algorithm is presented in Table I in the form of a pseudocode. Resampling is used in the algorithm when the effective particles quantity $\hat{N}_{eff}$ is below a threshold N_{thr} to prevent weight degeneracy.

We then obtain the estimated trajectory $\hat{\mathbf{x}}^{l,R}_{k-Tw+1:k}$ of the leader robot within the local reference frame of the follower robot at time step k.

For $t = k - T_w + 1$ to k,

$$\{\mathbf{x}^{l,R,i}_t\}^L_{i=1} = \left\{R\left(\theta^{f,i}_k\right)\left(\mathbf{x}^{l,i}_t - \mathbf{x}^{f,i}_k\right)\right\}^L_{i=1}$$

$$\hat{\mathbf{x}}^{l,R}_t = \sum_{i=0}^{L} \mathbf{x}^{l,R,i}_t w^i_t.$$

III. Trajectory-Following Control

Given $\hat{\mathbf{x}}^{l,R}_{k-Tw+1:k}$ describing the estimated trajectory of the leader robot from Section II-D, an appropriate function f_d is required to pick a point from the trajectory

$$\mathbf{x}_d = f_d\left(\hat{\mathbf{x}}^{l,R}_{k-Tw+1:k}\right)$$

where $\mathbf{x}_d = \begin{bmatrix} x_d & y_d & \theta_d \end{bmatrix}^T$ serves as the estimated desired/reference pose for the path-following controller at the current time step. It should meet user-defined selection criteria, for instance, a differentiable point on the trajectory that has a certain travel distance and/or straight-line distance to the latest position of the leader robot. In addition, reference tangential and angular velocities are required and calculated by

$$\begin{bmatrix} v_d \\ \omega_d \end{bmatrix} = \begin{bmatrix} \pm\sqrt{\dot{x}_d^2 + \dot{y}_d^2} \\ \dot{\theta}_d \end{bmatrix}$$

where v_d is a positive value when the leader robot is moving forward, and a negative number if moving backward.

A linear trajectory-following control algorithm is adopted to produce the control vector $\mathbf{u}^f_{k+1}$ for the follower robot at the next time step $k+1$.

$$\begin{bmatrix} v^f_{k+1} \\ \omega^f_{k+1} \end{bmatrix} = \begin{bmatrix} v_d \cos\theta_d \\ \omega_d \end{bmatrix} + \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 sign\left(v_d\right) & k_3 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ \theta_d \end{bmatrix}$$

$$\mathbf{u}^f_{k+1} = \begin{bmatrix} v^{f,l}_{k+1} \\ v^{f,r}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & -\frac{d}{2} \\ 1 & \frac{d}{2} \end{bmatrix} \begin{bmatrix} v^f_{k+1} \\ \omega^f_{k+1} \end{bmatrix}$$

where coefficients $k_1 = k_3 = 2\zeta\sqrt{\omega_d^2 + bv_d^2}$, $k_2 = b|v_d|$; $0 < \zeta < 1$ is a damping coefficient, and $b > 0$ is introduced as an additional degree of freedom. Please note that ζ and b are the only two parameters for the trajectory-following control algorithm. Interested readers can refer to [11] and [12] for details of the controller.

IV. Evaluation

A. Setup

A full list of evaluation parameters is presented in Table II. The vision related parameters are set based on detection of an ArUco board attached to the back of the leader robot using Kinect for Windows v2.

Student's *t*-distribution is used to generate independent and identically distributed kinematic process noise for each wheel. It is heavier-tailed than Gaussian, which means that it is more likely to produce samples that fall far from its mean. The Student's *t*-distribution used is formulated by $St\left(0, \lambda, \nu\right)$, where $\nu = 3$ is degrees of freedom, and the precision $\lambda = 1200$. It approaches normal distribution $\mathcal{N}\left(0, 0.05^2\right)$ when ν increases to ∞. Although the leader robot is moving along a predetermined path, its odometry data are intentionally corrupted with the process noise before transmitted to the follower.

Triangular distribution is used in inter-robot bearing measurement noise, which could be represented in the form of

$$Tri\left(\mu, \sigma^2\right) = \begin{cases} \frac{\sqrt{6\sigma^2} - |x-\mu|}{6\sigma^2} & \text{for } |x-\mu| \le \sqrt{6\sigma^2} \\ 0 & \text{elsewhere} \end{cases}$$

where the mean $\mu = 0$ and the standard deviation $\sigma = 2$ degrees.

B. Performance Metrics

Root-mean-square error (RMSE) is used to quantify position accuracy in trajectory estimation and path following. Three RMSE values are calculated to measure estimation performance.

1) e_{traj}: the RMSE of the leader's trajectory estimate at the time step t w.r.t the corresponding segment of ground truth trajectory,
2) e_{lpos}: the RMSE of the leader's current location estimate w.r.t the corresponding ground truth point,
3) e_{cpos}: the RMSE of the position of the reference pose estimate $\mathbf{x}_d$ w.r.t the corresponding point on the ground truth trajectory.

We also compute e_{ctrl}, which stands for the RMSE of the follower's current location w.r.t the control point, to evaluate the performance of path-following control.

Lastly, e_{fpos} is calculated to represent the RMSE of the follower's location w.r.t the corresponding ground truth point.

C. Results

In the first test, the leader-follower robots are initially placed on a 2D plane with a separation of 1 m. The leader robot is set to move at a constant linear velocity of $v^l_t = 1.5\ m/s$, while its heading is given as a function of time

$$\theta^l_t = \frac{\pi}{2}\cos\left(\frac{2\pi t}{14}\right)$$

TABLE II
Testing parameters

Parameter	Value
total Monte Carlo cases	50
total time duration	60 s
iteration period	33.3 ms
kinematic process noise in each wheel	$St\left(0, 1200, 3\right)$ m/s
vision observation noise in bearing	$Tri\left(0, 2\right)$ deg
vision observation noise in range	$\mathcal{N}\left(0, 0.05^2\right)$ m
vision detection success rate	0.8
vision detection max. range	4 m
vision detection min. range	0.5 m
camera horizontal FOV	70 deg
camera FPS	30
initial position interval	1 m
following distance	3 m
particle quantity L	2000
ζ in path-following controller	0.7
b in path-following controller	1.05

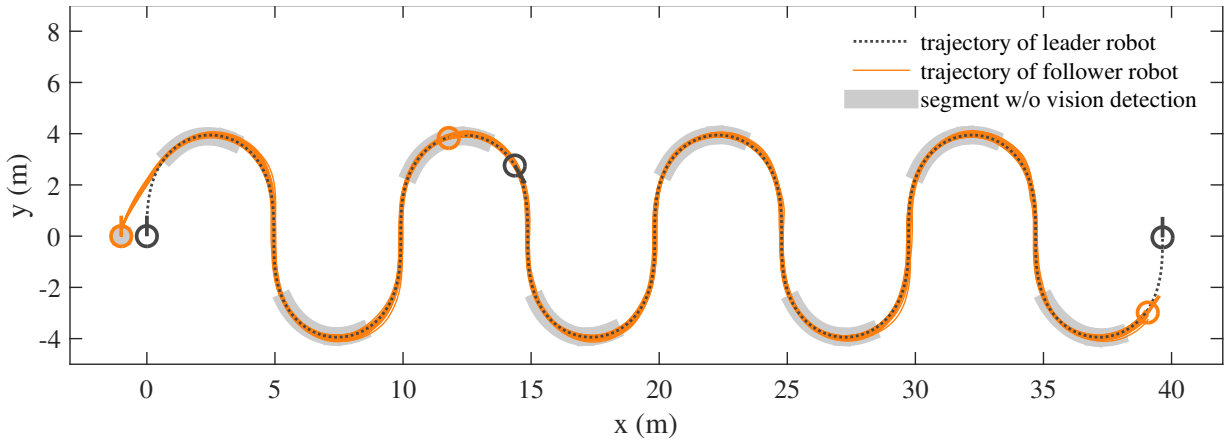


Fig. 2. Trajectories of the leader robot and the follower robot of 50 Monte Carlo runs of the proposed approach. The line segments marked grey in the figure represent positions where the follower loses vision detection due to the leader robot moving out of FOV. From left to right, the figure also shows the positions of the robot pair at $t = 0$, 19, and 56 s.

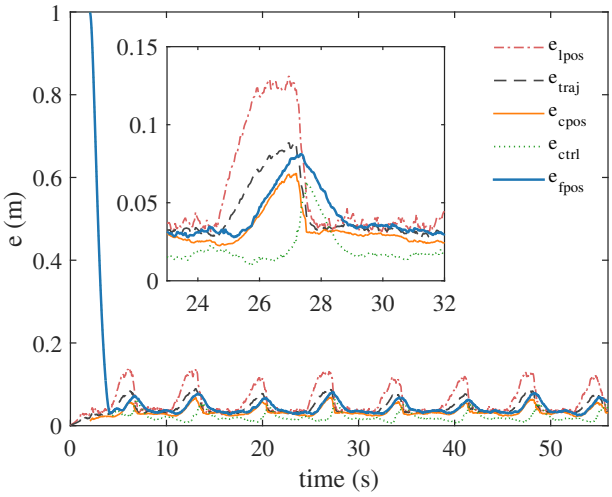


Fig. 3. Mean RMSEs of 50 Monte Carlo runs. It is shown that most of the time e_{cpos} is lower than e_{lpos} and with smaller fluctuation. As revealed in the figure, the estimation errors grow fast during vision detection outage, which happened 8 times in the test. It is however observed that in each outage period, e_{cpos} increases at a slower rate than e_{lpos}.

which forms a zig-zag shaped trajectory when time goes on, as illustrated in Figure 2. At every time step, the reference pose $\mathbf{x}_d$ for path-following control is set to be the point on the leader's estimated trajectory with a travel distance of 3 m to the current position of the leader robot.

Figure 2 also summaries trajectories of the follower robot from 50 Monte Carlo runs of the proposed leader-following approach. The trajectories are overall close to the leader's, showing accurate trajectory estimation and path-following control using the proposed approach. The zig-zag shape of the leader's trajectory however guides the leader to move out of the FOV of the follower's camera periodically.

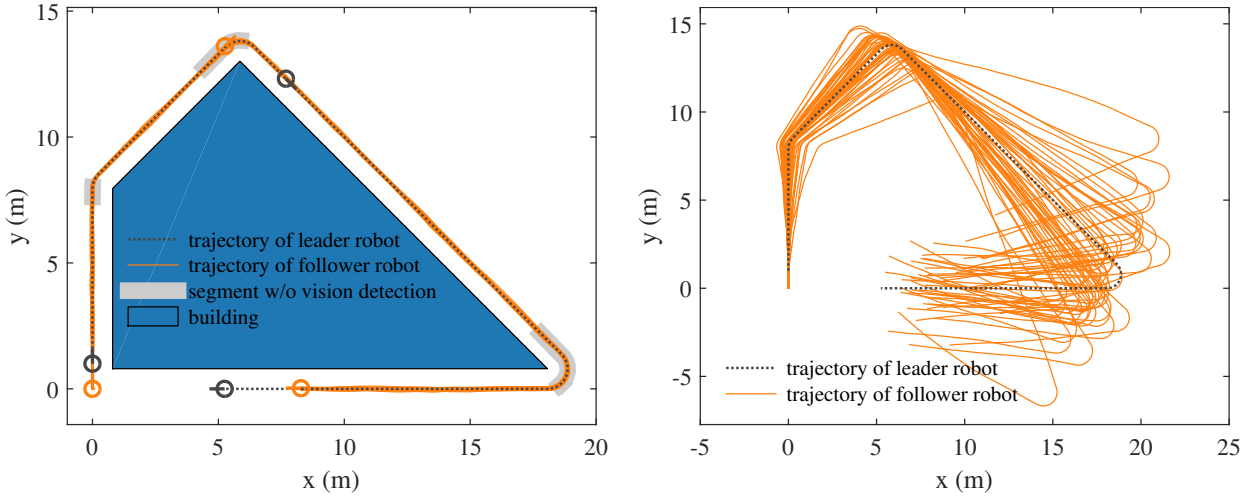
As observed in Figure 2, the follower's trajectory tends to deviate from the leader's when the follower fails the vision detection of the leader for seconds. The grey segments in Figure 2 highlight those observation dropout periods. It can also be seen in Figure 3 that during each such detection outage period, the errors in trajectory estimation, i.e., e_{lpos}, e_{traj}, and e_{cpos}, grow fast as in this situation the estimation algorithm has to rely on odometry data only, which contain kinematic process noise.

Furthermore, the follower's position error e_{fpos} increases along with the position error at the control point e_{cpos}, because the controller always sets the estimated historical pose $\mathbf{x}_d$ on the leader's trajectory as the desired pose of the follower. The averaged e_{fpos} is about 0.03 m before an observation outage happens, as shown in Figure 3. The error then accumulates over time, peaking at 0.08 m just before the outage finishes. As soon as the follower's camera recaptures the leader robot, the errors in trajectory estimation drop noticeably. Nevertheless, it takes a longer time for the controller to correct the accumulated position error e_{fpos} of the follower.

The test highlights the benefits of using a smoothed historical pose estimate as the reference point for path-following control. It can be summarised from the test results that 1) e_{cpos} is generally lower than e_{lpos} and 2) e_{cpos} grows at a slower rate than e_{lpos} during a vision detection outage period, both because of the reference pose estimate $\mathbf{x}_d$ being smoothed using "future" observations. Nevertheless, it is also observed that e_{fpos} is slightly larger than e_{cpos}, which can be explained by non-zero e_{ctrl} existing in path-following control.

Another test is performed to evaluate the algorithms when the leader robot makes sharper turns. As demonstrated in Figure 4, the leader robot moves around a polygon shaped building and sequentially makes three turns that are increasingly sharper, which are 45°, 90°, and 135°. At each turn, outage of vision detection happens due to a combination of out of FOV and occlusion by corners. The same set of parameters in Table II is used in the test, expect for a new total time duration of 32 s and a different leader's trajectory.

Figure 4 summarises follower's trajectories together with the leader's from 50 Monte Carlo runs. It demonstrates again



(a) (b)

Fig. 4. Trajectories of the robot pair moving around a building. Sequentially, the leader takes three turns that are increasingly sharper. a) shows the path-following results with vision detection from 50 Monte Carlo runs. The grey line segments in the figure represent positions where the vision detection outage happens due to the leader out of FOV of the follower and/or vision occlusion. The positions of the robot pair at $t = 0$, 11.8, and 32 s are also shown on the left, top, and bottom of the figure, respectively. b) illustrates the dead-reckoning results from 50 Monte Carlo runs, which are obtained by disabling vision detection and performing kinematic prediction based on odometry data.

the accurate trajectory estimation and path-following, and also the turning of the follower without cutting corners using the proposed approach. As a comparison, the evaluation is also performed without vision detection. The dead-reckoning results presented in Figure 4 show severe deviations from the leader's trajectory due to accumulated odometry error.

V. Conclusion

The paper proposes a Bayesian trajectory estimation based leader-following approach that can work with the presence of frequent dropouts in vision detection. The approach performs probabilistic trajectory prediction for the leader robot using noise-corrupted odometry measurements of both robots, and corrects the accumulated odometry error in the use of intermittent inter-robot relative observations that are obtained by detection of fiducial markers using an RGBD camera.

Evaluation results conclude that using the proposed approach the follower robot is able to 1) accurately follow the trajectory of a leader robot performing zig-zag movement, which causes periodic vision detection dropouts, and 2) turn without cutting corners when the leader is making sharp turns. The results are achieved mainly due to the use of smoothed historical pose from the leader's estimated trajectory as the reference point in path-following control.

More simulation tests and field experiments are expected in the future for further evaluating and improving the proposed approach. Also, the approach can be improved by incorporating other low-cost sensor information, for instance, ultra-wideband communication based ranging, inertial measurements, etc., to constrain the growth of location uncertainty of the leader robot during the outage periods of vision detection.

References

[1] S. Benhimane, E. Malis, P. Rives, and J. R. Azinheira, "Vision-based control for car platooning using homography decomposition," in *Proc. the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 2005, pp. 2161–2166.

[2] L. Xu, Z. Cao, P. Zhao, and Y. Yin, "The identifier-based relative position estimation for leader-follower robotic system," in *Proc. the IEEE International Conference on Mechatronics and Automation*, Tianjin, China, Aug. 2014, pp. 1691–1695.

[3] G. Orlando, E. Frontoni, A. Mancini, and P. Zingaretti, "Sliding mode control for vision based leader following," in *Proc. the European Conference on Mobile Robots (ECMR 2007)*, Freiburg, Germany, Sep. 2007, pp. 1–6.

[4] F. Espinosa, C. Santos, M. Marrón-Romera, D. Pizarro, F. Valdés, and J. Dongil, "Odometry and laser scanner fusion based on a discrete extended kalman filter for robotic platooning guidance," *Sensors*, vol. 11, no. 9, pp. 8339–8357, Aug. 2011.

[5] L. Zhang, T. Ahamed, Y. Zhang, P. Gao, and T. Takigawa, "Vision-based leader vehicle trajectory tracking for multiple agricultural vehicles," *Sensors*, vol. 16, no. 4, p. 578, Apr. 2016.

[6] S. K. Gehrig and F. J. Stein, "A trajectory-based approach for the lateral control of car following system," in *Proc. the IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, CA, Oct. 1998, pp. 3596–3601.

[7] H. K. Goi, J. L. Giesbrecht, T. D. Barfoot, and B. A. Francis, "Vision-based autonomous convoying with constant time delay," *Journal of Field Robotics*, vol. 27, no. 4, pp. 430–449, May 2010.

[8] ARToolKit, http://www.hitl.washington.edu/artoolkit/, [Online].

[9] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, Jun. 2014.

[10] M. Fiala, "ARTag, a fiducial marker system using digital techniques," in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, Jun. 2005, pp. 590–596.

[11] A. De Luca, G. Oriolo, and M. Vendittelli, "Control of wheeled mobile robots: An experimental overview," in *RAMSETE: Articulated and Mobile Robotics for Services and Technology*. Springer Science and Business Media, 2001.

[12] M. Brezak, I. Petrović, and N. Perić, "Experimental comparison of trajectory tracking algorithms for nonholonomic mobile robots," in *Proc. the IEEE International Conference on Industrial Electronics (IECON'09)*, Porto, Portugal, Nov. 2009, pp. 2229–2234.