2014 11th IEEE International
Conference on Control & Automation (ICCA)
June 18-20, 2014. Taichung, Taiwan

Vision-based Formation for UAVs

Feng Lin, Kemao Peng, Xiangxu Dong
Temasek Laboratories
National University of Singapore
117411 Singapore
Email: {tsllinf, kmpeng, tsldngx}@nus.edu.sg

Shiyu Zhao and Ben M. Chen
Department of Electrical and Computer Engineering
National University of Singapore
117576 Singapore
Email: {shiyuzhao, bmchen}@nus.edu.sg

***Abstract*—In this paper, we present a vision-based relative sensing system for UAVs to realize leader-follower formation flight without inter-vehicle communication. A monocular camera is mounted on the follower to detected the leader and measure the relative distance by using the geometry information of the leader without artificial markers. The measured relative distance is utilized to estimate the velocity and acceleration of the leader under the quasi-steady states assumption. Experimental results show that the proposed sensing system is capable of achieving the vision-based leader-follower formation flight.**

I. Introduction

There is a boom in vision sensing in academic research and industrial applications, since it is able to provide human-like perception, such as geometry of the scene, photometry of objects, and dynamics of the environment. By integrating vision sensors with other avionic sensors, unmanned systems can autonomously perform a variety of applications. The vision-based leader-follower formation is focused in this paper (see Figure 1), which has arouse great interests recently (see [10], [11], [13], [14]). It can also be considered as an air-to-air target problem that an aircraft tracks a target in the sky in a pre-defined relative distance and orientation.

Typically, the displacement between the tracker and the follower can be measured by using radars [4]. Such radar measurement is effective in long distance, but it becomes ineffective in short distance. In contrast to radars, the visual measurement is effective in short distance measurement [15], [9]. Stereo vision is a straightforward solution that can easily provide direct depth measurement. Its computation, however, is too intensive to be implemented easily in airborne computers. In [12], the vision-based localization for the formation flight was presented by using a monocular camera, and known artificial markers on the target are assumed. However, the vision-based target detection and the orientation tracking is ignored.

This motivated us to use a monocular camera to realize detection and 3D relative measurement based on the shapes and sizes of the leader. In this paper, a vision algorithm will be developed to detect a leader UAV without artificial markers. The relative distance and bearing are determined based on the image measurement with the geometry information of the leader. Although the relative distance cannot be measured accurately yet, it is well enough for the leader-follower formation in steady level flight. It is noted that the formation performance becomes worse to track a maneuvering leader if the relative distance is applied as the only reference to the follower UAV. Thus, it needs to construct the high-order derivatives of the reference such as the target velocity and acceleration, and to design a tracking and following control law applicable to track the reference with the high-order derivatives. The velocity and acceleration of the target will be estimated by using a novel algorithm in which the smooth flight trajectory of the leader is considered. The algorithm is developed based on the extended Kalman filtering and under assumption of the quasi-steady states. The resulting closed-loop system is verified in experiments to follow a leader UAV.

Fig. 1. Illustration of the vision-based formation.

II. 3D Measurement with a Monocular Camera

The machine learning approach is taken to identify the leader UAV. The relative distance is measured with a monocular camera with known geometry information of the leader. Pan/tilt camera orientation control is utilized to keep the leader in the center of the image in spite of the maneuvering of both the leader and follower.

A. Target Detection

There were a lot of techniques published for the target detection such as Haar wavelet based AdaBoost cascade, Histogram of oriented gradient features and others [2], [1]. Among them, the first method is more applicable to our scenario and real-time processing. The principle idea for the target detect is to combine a set of the Haar wavelet features

to separate a specific target from background. To realize real-time operation, the set of the features have been formed in a cascade manner to reduce the computation cost. The Haar wavelet features are employed here due to their rich image representation and very fast computation by using the integral image [19], which is suitable for the onboard processing.

Normally, more than one hundred of thousands of such features can be extracted from a small image sub-window (e.g. 24 × 24 pixels) by varying the scale and location of the feature. Each Haar wavelet feature can be used to construct a weak classifier to identify the target.

$$h_j(x) \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} , \tag{1}$$

where x is a sampled sub-window in an image. θ_j is a threshold to achieve minimal misclassification. $p_j \in \{-1,\ 1\}$ is used to define the direction of inequality.

A single weak classifier cannot perform very well. Fortunately, as proved in [18], it is theoretically possible to combine multiple weak classifiers to be a better classifier, namely strong classifier. In [3], the AdaBoost algorithm is proposed to form a strong classifier by selecting a small set of weak classifiers and calculating their corresponding weight based on training samples. The detailed procedure and statistical behavior of the final strong classifier can be found in [3], [20]. The final strong classifier $H(x)$ is formed as a linear combination of selected n weak classifiers.

$$H(x) = \sum_{j=1}^{n} \alpha_j h_k(x) , \tag{2}$$

where α_j is a weight to minimize the exponential loss of the classifier. If $H(x) \geq \eta_t,\ \eta_t = \frac{1}{2}\sum_{j=1}^{n} \alpha_j$, the target is identified. The strong classifier will have better performance than each weak classifier.

Additionally, we can sort the weak classifiers according to the weight α_j and compute the strong classifier iteratively.

$$H_m(x) = H_{m-1}(x) + \alpha_m h_m(x),\ \alpha_m < \alpha_{m-1} , \tag{3}$$

where $H_{m-1}(x) = \sum_{j=1}^{m-1} \alpha_j h_j(x)$, and $H_0 = 0$. In each iteration, if $H_m(x) + \sum_{j=m+1}^{n} \alpha_j < \eta_t$, we can conclude that the target is not in the sampled sub-window, and the rest weak classifiers ($H_j, m < j \leq n$) do not need to be checked in the sampled sub-window. Thus, the computational cost can be reduced significantly and this is the principle idea of the cascade of the classifiers. This structure is suited to the scenario that there is an overwhelming majority of negative sub-windows in an image and the negative sub-windows have less pattern than the target that can be identified by using less features.

In order to train the classifier proposed, we need to collect training and testing images. Although we can synthesize such images using OpenCV utilities, we still prefer to collect natural images of the leader UAV in ground and flight tests. The positive images are taken under different situations, such as lighting, view angle, depth, background, and so on. The images

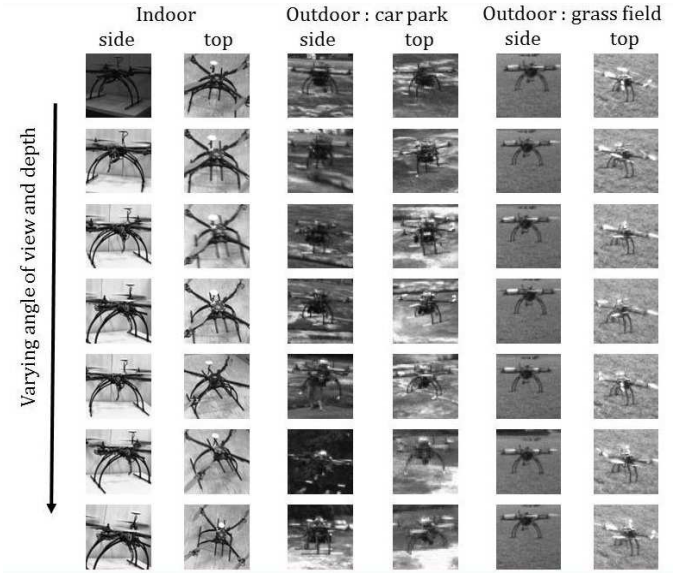

Fig. 2. Positive images.

of the target have been extracted from the collected images, and resize to 24 × 24 pixels shown in Fig. 2.

We also collect negative images that do not contain the objects of interest. We use both the on-line image database and images collected during the flight tests as the negative images. The number of negative images is at least twice large than that of positive images. We train the classifier using the haartraining utility provided by OpenCV libraries, and obtain the cascade classifier. According to the created the receiver operating characteristic (ROC) curve, the threshold (minimum number of windows in the neighbour) of the final layer classifier is selected. For the cascade classifier train, it should be noted that:

1) To reduce false alarm rate in the detection, we need to increase the stages of the classifier, but computation cost will increase in the target detection;
2) To increase the hit rate, we must increase the training samples. It will not affect the target detection time, but it will increase the training time.

When the training process is over, the cascade strong classifier is determined. The target detection will repeat the feature recognition with the known parameters to identify the target. The location of the target in the image will be applied to determine the relative displacement.

B. Relative Displacement

The relative distance and orientation of the leader with respect to the follower can be computed by using the known camera model and geometry information of the leader when it is identified and its location is determined in the image. The camera model is given below

$$\begin{cases} x_{\mathrm{i}} &= f\dfrac{x_{\mathrm{c}}}{z_{\mathrm{c}}} \\ y_{\mathrm{i}} &= f\dfrac{y_{\mathrm{c}}}{z_{\mathrm{c}}} \end{cases} , \tag{4}$$

where f is the focus length of the camera. $(x_{\rm c}, y_{\rm c})$ and $(x_{\rm i}, y_{\rm i})$ are the coordinates of a point in the camera and image frame respectively.

Since the geometry information of the leader is known, the maximal width, $y_{\rm max,i}$, of the leader at Y axis of the image frame is used to determine the relative distance to the camera. However, the corresponding real width of the leader is not easy to be known. A nominal width, $y_{\rm norm,c}$, is approximately regarded as the corresponding real width. In such approximation, the distance of the leader to the camera can be calculated as follows,

$$\lambda = f\frac{y_{\rm norm,c}}{y_{\rm max,i}} = f\frac{y_{\rm norm,c}}{|y_{\rm L,i} - y_{\rm R,i}|} \ , \tag{5}$$

where λ denotes the distance of the leader to the camera. $y_{\rm L,i}$ and $y_{\rm R,i}$ are the Y coordinates of the left and right extreme points of the leader detected from on its contour in the image.

The required relative distance is the distance of the leader's mass center to the follower's mass center. Unfortunately, it is impossible to measure and/or calculate the coordinates of the leader's mass center in the camera frame with the visual measurement. Thus, a coordinate average of the points on the leader contour is adopted as the center of the leader in the camera frame. The relative distance in the camera frame, $p_{\rm t,c}$, can be determined as follows,

$$p_{\rm t,c} = \frac{1}{N}\sum_{k=1}^{N} \lambda C_{\rm a} p_{k,\rm i} \ , \tag{6}$$

where $C_{\rm a} = {\rm Diag}[f^{-1},\ f^{-1},\ 1]$ is the camera parameter matrix. $p_{k,\rm i}$ denotes the coordinates of the k-th point in the image frame. N denotes the total number of the chosen points. The relative distance of the leader with respect to the follower in the North-East-Down (NED) frame, $\Delta p_{\rm gt}$, can be given as follows,

$$\Delta p_{\rm gt} = B'_{\rm gb}(B'_{\rm bc} p_{\rm t,c} + c_{\rm b}) \ , \tag{7}$$

where $B_{\rm gb}$ denotes the transformation matrix from the NED frame to the body frame. $B_{\rm bc}$ denotes the transformation matrix from the body frame to the camera frame. $c_{\rm b}$ denotes the coordinates of the camera in the body frame.

C. Next Camera Direction

In order to hold the leader appearing in the images, the camera is controlled to point to it. The next camera direction needs to be estimated to drive the pan/tilt camera. Such estimation is available as the future relative motion between the leader and follower can be predicted under the quasi-steady states seen in Section III. If the next displacement of the leader relative to the tracker is predicted as $\Delta\hat{p}_{\rm gt}$ in the NED frame, we can compute the pitch angle $\theta_{\rm b}$ and azimuth angle $\psi_{\rm b}$ of Z axis in camera frame with respect to the body frame.

$$\theta_{\rm b} = \arctan\left(\frac{-d_{\rm z}}{\sqrt{d_{\rm x}^2 + d_{\rm y}^2}}\right), \quad \psi_{\rm b} = \arctan\left(\frac{d_{\rm y}}{d_{\rm x}}\right),$$
$$\begin{pmatrix} d_{\rm x} \\ d_{\rm y} \\ d_{\rm z} \end{pmatrix} = \frac{\Delta p}{\|\Delta p\|}, \quad \Delta p = B_{\rm gb}\Delta\hat{p}_{\rm gt} - c_{\rm b}, \tag{8}$$

Note that the Z axis of the camera framework is align with X axis of the body frame. The camera can be driven to the predicted direction to detect the leader. Eventually, the determined relative displacement will be applied to estimate the velocity and acceleration of the leader.

D. Motion Compensation

The motion compensation has been realized by virtually simulate a pan/tilt servo base frame and remove the pitch angle of the UAV platform.

$$\bar{\theta}_{\rm b} = \theta_{\rm b} - \theta \ . \tag{9}$$

The tilt servo control of the pan/tilt camera is illustrated In Fig. 3. In Fig. 3, $\theta_{\rm target}$ is the azimuth angle of the target with respect to the UAV.

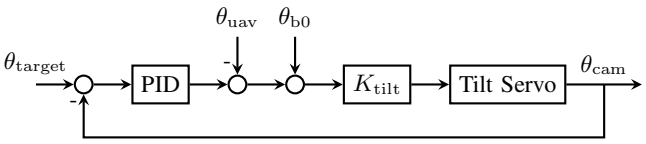

Fig. 3. The vision-based control of the tilt servo.

E. Image Tracking

The target detection methods mentioned above are normally used to initialize image tracking in many vision applications. An effective combination of detection and motion estimator is able to speed up the processing, as well as deal with moving targets and uncertainties of outdoor environments, such as variations in lighting, background, etc. Image tracking typically involves mathematical tools such as the Kalman filter, Bayesian network [6], [5], [21]. This image tracking method can be referred to as *Filtering and Data Association approach*, considered as a top-down process.

To predict the target location in the image, a motion model is required. It is well known that the motion of a point mass in the two-dimensional plane can be defined by its two-dimensional position and velocity vector. Let $\boldsymbol{x} = [x, \dot{x}, y, \dot{y}]^{\rm T}$ be the state vector of the centroid of the tracked target in the Cartesian coordinate system. Non-maneuvering motion of the target is defined by it having zero acceleration: $[\ddot{x}, \ddot{y}]^{\rm T} = [0, 0]^{\rm T}$. Strictly speaking, the motion of the intended targets may be maneuvering with unknown inputs. Nevertheless, we assume the standard 4-th order non-maneuvering motion model by setting the acceleration as $[\ddot{x}, \ddot{y}]^{\rm T} = \boldsymbol{w}(t)$, where $\boldsymbol{w}(t)$ is a white noise process [8], which should be sufficient for the target tracking in the image.

In addition, it is necessary to consider the motion of the pan/titl camera, which can be compensated using its steady state value, since it normally has higher bandwidth than the dynamics of the UAV. The discrete-time model of the target motion can be expressed as

$$\begin{cases} \boldsymbol{x}(k|k-1) &= \Phi\boldsymbol{x}(k-1) + F_{\rm cam}\Delta\boldsymbol{u}_{\rm cam} + \Lambda\boldsymbol{w}(k-1), \\ \boldsymbol{z}(k) &= Hx(k) + \boldsymbol{v}(k), \end{cases}$$

where

$$\Phi = \begin{bmatrix} 1 & \mathrm{T_s} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \mathrm{T_s} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \frac{\mathrm{T_s^2}}{2} & 0 \\ \mathrm{T_s} & 0 \\ 0 & \frac{\mathrm{T_s^2}}{2} \\ 0 & \mathrm{T_s} \end{bmatrix},$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\boldsymbol{u}_{\mathrm{cam}} = \begin{pmatrix} \Delta\phi_{\mathrm{b}} \\ \Delta\theta_{\mathrm{b}} \end{pmatrix}, \quad F_{\mathrm{cam}} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$\mathrm{T_s}$ is the sampling period of the vision software. A Kalman filter can then be designed based on the above motion model to estimate the states of the target in the image plane. Based on the motion mode, we can predict the possible location of the target in the next frame, and also define the region of interest $N_{\mathrm{s}}(k)$ as a neighbourhood of the predicted location of the target in the image $\mathbf{I}(x, y)$, which is given below.

$$N_{\mathrm{s}}(k) = \{\mathbf{I}(x, y) \mid x \in [x_1, x_2],\ y \in [y_1, y_2]\}\ , \tag{10}$$

where

$$\begin{aligned} x_1 &= \hat{x}(k|k-1) - 0.5 \times (w_{\mathrm{t}}(k-1) + \varepsilon\sigma_{\mathrm{x}}(k|k-1))\ , \\ x_2 &= \hat{x}(k|k-1) + 0.5 \times (w_{\mathrm{t}}(k-1) + \varepsilon\sigma_{\mathrm{x}}(k|k-1))\ , \\ y_1 &= \hat{y}(k|k-1) - 0.5 \times (h_{\mathrm{t}}(k-1) + \varepsilon\sigma_{\mathrm{y}}(k|k-1))\ , \\ y_2 &= \hat{y}(k|k-1) + 0.5 \times (h_{\mathrm{t}}(k-1) + \varepsilon\sigma_{\mathrm{y}}(k|k-1))\ , \end{aligned}$$

w_{t} and h_{t} are the width and height of the target detected in the previous frame. σ_{x} and σ_{y} are the standard deviation of the estimation of x and y, given by the Kalman filter. ε is a coefficient. We can also reduce the probability of the false alarm by using the motion estimator.

F. Data Fusion

In the vision aided detection and tracking, we will obtain the different outputs from the detector, motion estimator and tracker. In order to achieve a robust results, we make the decision in terms of the following rules.

1) If an object has been continuously identified by the detector more than a certain frames, we will define it as the target.
2) Once the target is identified, the motion estimator will predict the region where the target will appear in the new image.
3) We will adopt the target's location given by the detector in first priority, as long as this location is in the predicted region by the motion estimator. That is because the detector has already considered both of the geometry and intensity information. Otherwise, if the detector cannot find the target, we will select the output of the image tracker as the target.
4) In rare case, if both detector and tracker cannot give the target location, we will use the predicted location by the Kalman filter as the location of the target. That is mainly caused by the distorted or corrupted image.
5) If the target has been lost in a certain frames, the algorithm will switch to the detection mode to search the target.

After the target has been detected, we will find the contour of the target first and calculate the geometry center of the target. We will also update the target template, including the histogram and the threshold values accordingly.

III. Estimation of Target Velocity and Acceleration

It is useful to estimate the leader's velocity and acceleration with the measured relative displacement to realized perfect formation. The algorithm is developed with the extended Kalman filtering in a class of target motion such as the quasi-steady states (QSS). Such QSS was proposed in the wind frame for the flight control design [16] and was defined in the NED frame to describe a class of the practical flight motions for estimation of target motion [17]. The definition of QSS is introduced below.

Definition 3.1: The quasi-steady states are the flight states in which the derivatives of the aircraft ground speed, flight path angle and azimuth angle are constant.

Remark 3.1: The definition of QSS implies the classification of the steady, maneuvering and super maneuvering states. The steady states are the flight states in which the derivatives of the aircraft ground speed, flight path angle and azimuth angle are zero. The maneuvering states are QSS. The super-maneuvering states are the flight states in which the derivatives of the aircraft ground speed, flight path angle and azimuth angle are not constant.

The estimation algorithm developed based on the assumption of QSS is applicable to the steady maneuvering motions of the target, which include the main motion of the target. Based on the definition of QSS, we proceed to formulate the extended Kalman filtering.

A. Kinematical Models of Target

In QSS, the kinematical models of the target can be presented as follows,

$$\dot{p}_{\mathrm{gt}} = V_{\mathrm{gt}} \begin{pmatrix} C_{\theta_{\mathrm{st}}} C_{\psi_{\mathrm{st}}} \\ C_{\theta_{\mathrm{st}}} S_{\psi_{\mathrm{st}}} \\ -S_{\theta_{\mathrm{st}}} \end{pmatrix}, \quad \begin{pmatrix} \ddot{V}_{\mathrm{gt}} \\ \ddot{\theta}_{\mathrm{st}} \\ \ddot{\psi}_{\mathrm{st}} \end{pmatrix} = 0, \tag{11}$$

where $C_{\theta_{\mathrm{st}}} = \cos\theta_{\mathrm{st}}$, $C_{\psi_{\mathrm{st}}} = \cos\psi_{\mathrm{st}}$, $S_{\theta_{\mathrm{st}}} = \sin\theta_{\mathrm{st}}$ and $S_{\psi_{\mathrm{st}}} = \sin\psi_{\mathrm{st}}$. p_{gt} denotes the displacement of the target in the NED frame. V_{gt}, θ_{st} and ψ_{st} denote the ground speed, flight path angle and azimuth angle of the target respectively. The extended Kalman filtering (EKF) based estimation algorithm will be developed by using those kinematical models.

B. Formulation of EKF

A formulation of the extended Kalman filtering can be written to estimate the velocity and acceleration of the target as follows,

$$\begin{cases} x(k+1) = f[x(k)] + w(k), \\ y(k) = Cx(k) + v(k), \end{cases} \tag{12}$$

where

$$x := \begin{pmatrix} \Delta p_{\mathrm{gt}} \\ V_{\mathrm{tpt}} \\ \dot{V}_{\mathrm{tpt}} \end{pmatrix}, V_{\mathrm{tpt}} = \begin{pmatrix} V_{\mathrm{gt}} \\ \theta_{\mathrm{st}} \\ \psi_{\mathrm{st}} \end{pmatrix}, w := \begin{pmatrix} w_{\mathrm{pgt}} \\ w_{\mathrm{Vtpt}} \\ w_{\dot{V}\mathrm{tpt}} \end{pmatrix}, y := \Delta p_{\mathrm{gt}},$$

Δp_{gt} denotes the displacement of the target with respect to the tracker in the NED frame. w and v denote the system and measurement noises respectively. They are zero-mean Gaussian noises. We need to present the related items for computation of EKF such as

$$f, \quad \frac{\partial f}{\partial x}, \quad C.$$

The function f can be presented as follows,

$$f = A\,x(k) + B\Delta V(k), \tag{13}$$

where

$$\Delta V(k) = V_{\mathrm{gt}}(k) \begin{pmatrix} C_{\theta_{\mathrm{st}}} C_{\psi_{\mathrm{st}}} \\ C_{\theta_{\mathrm{st}}} S_{\psi_{\mathrm{st}}} \\ -S_{\theta_{\mathrm{st}}} \end{pmatrix} - V_{\mathrm{g}}(k) \begin{pmatrix} C_{\theta_{\mathrm{s}}} C_{\psi_{\mathrm{s}}} \\ C_{\theta_{\mathrm{s}}} S_{\psi_{\mathrm{s}}} \\ -S_{\theta_{\mathrm{s}}} \end{pmatrix},$$

$$A = \begin{bmatrix} I & 0 & 0 \\ 0 & I & TI \\ 0 & 0 & I \end{bmatrix}, \quad B = \begin{bmatrix} TI \\ 0 \\ 0 \end{bmatrix},$$

T denotes the sampling period. I denotes the identity matrix with appropriate dimensions. $V_{\mathrm{g}}, \theta_{\mathrm{s}}$ and ψ_{s} denote the ground speed, flight path angle and azimuth angle of the tracker. The partial derivative of f to x is given as follows,

$$\left(\frac{\partial f}{\partial x}\right)' = A + B\,[\,0 \quad B_{\mathrm{Vtpt}} \quad 0\,], \tag{14}$$

where

$$B_{\mathrm{Vtpt}} = \begin{pmatrix} C_{\theta_{\mathrm{st}}} C_{\psi_{\mathrm{st}}} & -V_{\mathrm{gt}}(k) S_{\theta_{\mathrm{st}}} C_{\psi_{\mathrm{st}}} & -V_{\mathrm{gt}}(k) C_{\theta_{\mathrm{st}}} S_{\psi_{\mathrm{st}}} \\ C_{\theta_{\mathrm{st}}} S_{\psi_{\mathrm{st}}} & -V_{\mathrm{gt}}(k) S_{\theta_{\mathrm{st}}} S_{\psi_{\mathrm{st}}} & V_{\mathrm{gt}}(k) C_{\theta_{\mathrm{st}}} C_{\psi_{\mathrm{st}}} \\ -S_{\theta_{\mathrm{st}}} & -V_{\mathrm{gt}}(k) C_{\theta_{\mathrm{st}}} & 0 \end{pmatrix}.$$

The matrix C is constant and is presented as follows,

$$C = [\,I \quad 0 \quad 0\,]. \tag{15}$$

Remark 3.2: In the EKF formulation, the derivatives of the target ground speed, flight path angel and azimuth angle are assumed subject to the Gaussian-Markov process even though they are defined to be constant in QSS. The physical constraint can be imposed to the state equation, such as the target ground speed being more than zero, the maximal/minimal derivatives of the target ground speed, flight path angle and azimuth angle.

IV. Integrated Simulation

Although each proposed algorithm, such as vision processing and formation control, has been tested separately, it is still useful to test them in an integrated simulator which simulates dynamics of the UAVs and generates the synthetic images for the vision processing. The simulation is to check the data flow of the algorithms and verify the performance of the whole system before flight tests. As illustrated in 4, the integrated simulation includes the following key components

1) Plant model: simulate 6 degree of freedom (DoF) dynamic model of the UAVs;
2) Flight control: simulate the inner-loop and outer-loop control of the UAVs;
3) Image simulation: generate the synthetic images based on the camera model and relative pose between the leader and the follower;
4) Vision processing: execute the proposed the vision-based relative sensing algorithms;
5) Motion estimation: execute the prosed EKF based motion estimation algorithm;
6) Pan/tilt control: execute the proposed pan/tilt camera control algorithm.

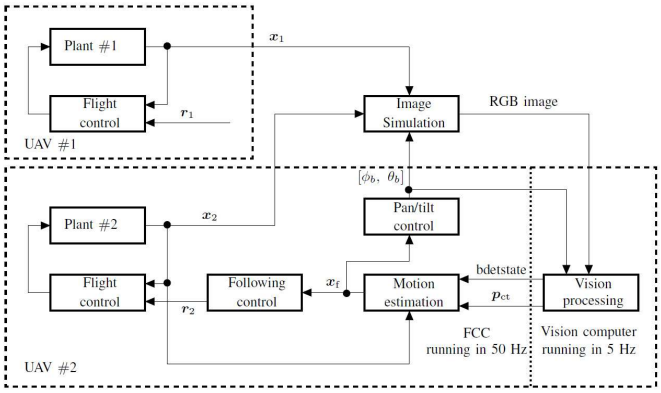

Fig. 4. Block diagram of the integrated simulation.

In Fig. 4, $\boldsymbol{x} = [x, y, z, u, v, w, \phi, \theta, \psi]$ is the state of the UAV. x_f is the estimated target states, ϕ_{b} and θ_{b} is the servo control input for the pan/tilt camera.

V. Experimental Results

In this section we present flight experimental results to evaluate the proposed vision-based formation algorithms. The flight experimental platform is an autonomous quadrotor UAV constructed by NUS UAV Team. The flight and formation control law is designed with the dynamic inversion and RPT techniques [17], and the online path planning is realized using the reflexxes motion libraries presented in [7]. The proposed vision-based formation algorithms have been implemented in the onboard system of the UAV.

The leader-follower formation has been conducted with the leader flying at the speed of 2 m/sec. The follower was controlled to keep a pre-defined relative distance to the leader,

i.e. 6 m in x-axis and 1 m in z-axis in the tracking frame. The flight trajectories and the tracking errors have been shown in Fig. 5 and 6. During the flight, the target detection is running in 5 Hz. In Fig. 6, it is observed that the tracking errors are converged to zero. The flight test has shown the robustness and effectiveness of the proposed vision-based formation algorithms.

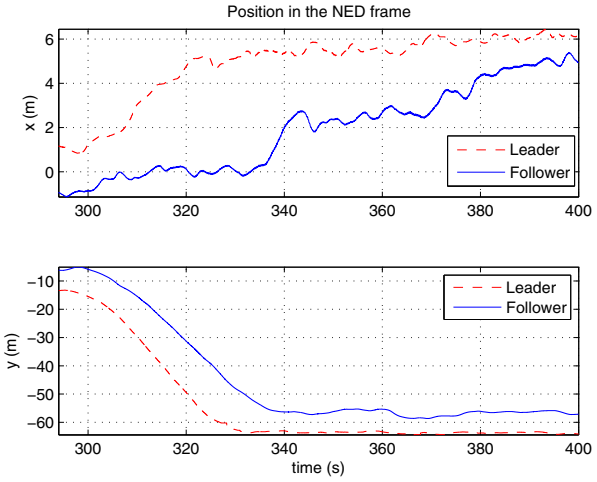

Fig. 5. The position of the leader and follower in the NED frame.

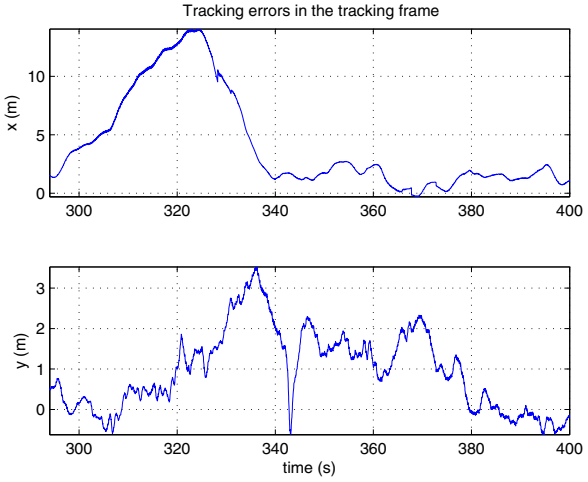

Fig. 6. The tracking error in the tracking frame.

VI. Concluding Remarks

The development of the vision-based leader-follower formation system has been presented in this paper. The system has been verified in simulation and flight experiments. A monocular camera is applied to implement the 3D measurement with the learning technique. The target velocity and acceleration can be estimated with the measured relative displacement under assumption of the quasi-steady states. The autonomous formation control law is successfully designed with the dynamic inversion and the robust perfect tracking technique together. Such techniques are also applicable to the other flight vehicles. How to handle the non-smooth flight trajectory of quadrotor UAVs will be investigated in future work.

References

[1] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: an evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 743–761, 2012.

[2] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: survey and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 2179–2195, 2009.

[3] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the 13th Conference on Machine Learning*, San Francisco, USA, 1996, pp. 148–156.

[4] P. E. Howl and D. Malvern, "Target tracking using television-based bistatic radar," *IEE Proceedings - Radar, Sonar and Navigation*, vol. 146, no. 3, pp. 166–174, 1999.

[5] W. M. Hu, T. N. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, man, and cybernetics*, vol. 34, no. 3, pp. 334–352, 2004.

[6] E. N. Johnson, A. J. Calise, Y. Watanabe, J. Ha, and J. C. Neidhoefer, "Real-time vision-based relative aircraft navigation," *Journal of Aerospace Computing, Information, and Communication*, vol. 4, pp. 707–738, 2007.

[7] T. Kroger and F. M. Wahl, "Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94–110, 2010.

[8] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking, Part I: Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[9] F. Lin, X. Dong, B. M. Chen, K. Y. Lum, and T. H. Lee, "A robust real-time embedded vision system on an unmanned rotorcraft for ground target following," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1038–1049, February 2012.

[10] L. Ma, C. Cao, N. Hovakimyan, C. Woolsey, V. Dobrokhodov, and I. Kaminer, "Development of a vision-based guidance law for tracking a moving target," AIAA Paper 2007–6744, 2007.

[11] V. Madyastha and A. Calise, "An adaptive filtering approach to target tracking," in *Proceedings of the 2005 American Control Conference*. Portland, OR, USA: IEEE, 2005, pp. 1269–1274.

[12] Z. Mahboubi, Z. Kolter, G. B. T. Wang, and A. Y. Ng, "Camera based localization for autonomous uav format flight," in *Proceedings of A2011 AIAA Infotech@Aerospace*, Saint Louis, U.S.A., 2011.

[13] S. Oh and E. N. Johnson, "Relative motion estimation for vision-based formation flight using unscented kalman filter," AIAA Paper 2007–6866, 2007.

[14] M. Pachter, N. Ceccarelli, and P. R. Chandler, "Vision-based target geolocation using micro air vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 3, pp. 597–615, 2008.

[15] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 14–35, 1993.

[16] K. Peng, K. Y. Lum, E. K. Poh, and D. Li, "Flight control design using hierarchical dynamic inversion and quasi-steady states," in *AIAA Guidance, Navigation and Control Conference*, 2008, pp. 2008–6491.

[17] K. Peng, S. Zhao, F. Lin, and B. M. Chen, "Vision based target tracking/following and estimation of target motion," in *AIAA Guidance, Navigation and Control Conference*, 2013, pp. 2013–5036.

[18] R. E. Schapire, *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.

[19] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. 511–518.

[20] T. Zhang, "Statistical behavior and consistency of classification methods based on convex risk minimization," *Machine Learning*, vol. 32, pp. 56–134, 2004.

[21] Q. M. Zhou and J. K. Aggarwalb, "Object tracking in an outdoor environment using fusion of features and cameras," *Image and Vision Computing*, vol. 24, no. 11, pp. 1244–1255, 2006.