Proceedings of the 33rd Chinese Control Conference
July 28-30, 2014, Nanjing, China

Guidance, Navigation and Control of an Unmanned Helicopter for Automatic Cargo Transportation

Fei Wang[12], Peidong Liu[1], Shiyu Zhao[1], Ben M. Chen[1], Swee King Phang[1], Shupeng Lai[1], Tong H. Lee[1], Chenxiao Cai[3]

1. Unmanned Systems Research Group, National University of Singapore

2. Temasek Laboratories, National University of Singapore

3. School of Automation, Nanjing University of Science and Technology

Abstract: This paper presents an intelligent and robust guidance, navigation and control solution for a rotorcraft UAV to perform automatic cargo transportation between two moving platforms. Different from the conventional GPS/INS based navigation, the proposed UAV system also integrates a scanning laser range finder to robustly correct its altitude measurement and an advanced vision sub-system to detect the cargo loading and unloading positions. The whole system has been verified in the 2nd AVIC Cup – International UAV Innovation Grand Prix. The joint team of National University of Singapore and Nanjing University of Science and Technology have won the first place in the final round of the rotary-wing category competition.

Key Words: UAV cargo transportation, UAV vertical replenishment, visual guidance

1 Introduction

Rotorcrafts are frequently used for cargo transportation or vertical replenishment on seaborne vessels (see Fig. 1). The conventional way of doing it involves a manned helicopter, which requires a very experience human pilot. Nevertheless, high risk still exists even when the most skillful pilot is hired. The recent advancement of unmanned aerial vehicles (UAVs) however has opened the possibility of using unmanned rotorcrafts for this kind of cargo transportation or good delivery tasks. One interesting case study is AirMule from UrbanAero, which has been used to transport supplies in Israel for military purposes [1]. A much fresher example is Amazon's PrimeAir project which aims to deliver package to every customer's backyard in 30 minutes using small quadrotor UAVs. To realize such applications, many theoretical and technical problems need to be solved. Cargo transportation using UAVs usually involves a slung load mechanism. In [2, 3], innovative control methods have been proposed to solve the general UAV slung load problem. For lighter cargos, the loading mechanism can also be designed in the form of rigid claws, such as those appeared in [4, 5].

However, most of the existing works assume that both the UAV and the cargo positions are accurately known. This assumption is valid only in occasions where the environment is fully controlled. However, direct and accurate measurement is usually unavailable in real-life scenarios. In order to expand the horizon of applications a small-scale UAV can do, an onboard real-time navigation and guidance system, which can provide accurate measurement and guidance information, needs to be developed. One elegant solution is to incorporate a computer vision sub-system for target searching and tracking. In fact, vision-based target detection and localization have been investigated extensively. Some of them rely on visual targets with special shapes and features such as points, lines and curves [6]. Others target on more general objects like a helipad [7], a mobile ground vehicle [8] or another UAV platform [9].

Although abundant theoretical methods in solving the above individual problems can be found in literature, there are limited results on full system integration and implementation. In this paper, we propose a comprehensive UAV cargo transportation system which incorporates a small-scale single-rotor helicopter with onboard sensors and processors, an innovative cargo loading and unloading mechanism, a set of UAV guidance, navigation and control (GNC) algorithms, and a vision-based target searching and localization subsystem.

Fig. 1: Rotorcraft vertical replenishment (U.S. Navy Photo: Use of released U.S. Navy imagery does not constitute product or organizational endorsement of any kind by the U.S. Navy.)

The developed UAV system, named NUS^{2}T-Lion, has taken part in the 2nd AVIC Cup – International UAV Innovation Grand Prix (UAVGP), which was held in Beijing in September 2013. In this competition, rotary-wing UAVs are required to automatically transport cargos between two parallel moving ships. The cargos are in the form of buckets with handles and they are initially placed within colored circles drawn on the surface of the first ship. Circles with a different color are drawn on the other ship, indicating the unloading positions. The ships are simulated by ground platforms moving on railways.

This paper is structured as follows: Section 2 introduces the hardware configuration and system overview of NUS^{2}T-Lion. Section 3 to Section 5 expand the UAV control, navi-

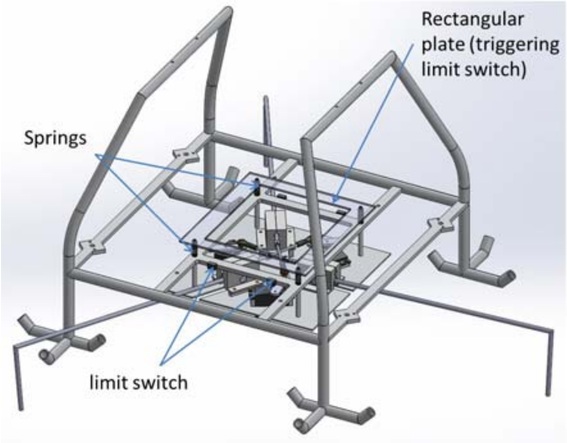

Fig. 2: Cargo grabbing mechanism with load sensing

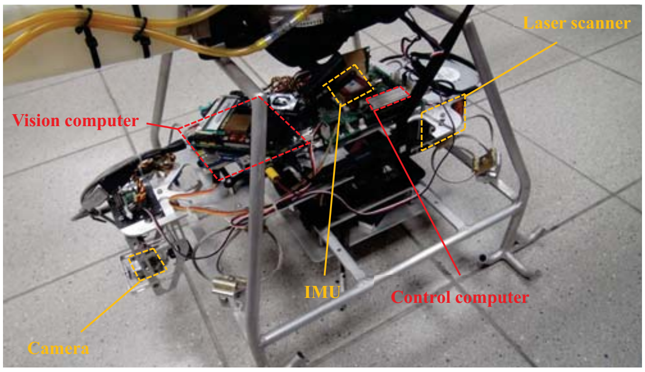

Fig. 3: Onboard avionic system

gation and guidance algorithms respectively. Section 6 provides flight test data and competition results to justify the performance and reliability of the overall system. Concluding remarks are made in Section 7.

2 Hardware Configuration and System Overview

NUS^{2}T-Lion is built upon the Thunder Tiger Raptor 90 SE Nitro radio-controlled (RC) helicopter. Its original nitro engine is replaced by a more efficient gasoline engine for endurance extension. An omnidirectional claw mechanism is installed (see Fig. 2) to make the grasping of bucket handles independent to the UAV heading angle. The design also integrates a load sensing mechanism which can differentiate a successful cargo grasping from a failure. To achieve it, four limit switches are installed under a rectangular plate supported by springs. Once a cargo with sufficient weight is loaded, at least one of these limit switches will be triggered.

For the onboard avionics (see Fig. 3), there are three main sensors, namely a SBG Systems IG-500N GPS aided inertial navigation system (GPS/INS) unit, a Matrix Vision mvBlueFOX camera, and a Hokuyo URG-30LX scanning laser range finder. Two onboard computers are used; one for the implementation of GNC algorithms, and the other more powerful one dedicated for vision processing. This dual-computer structure improves system modularity and reliability. The control computer is a Gumstix Overo Fire embedded computer, which has an ARM processor running at 720 MHz. The vision computer is an Ascending Technologies Mastermind, which has a powerful Intel Core i7 processor.

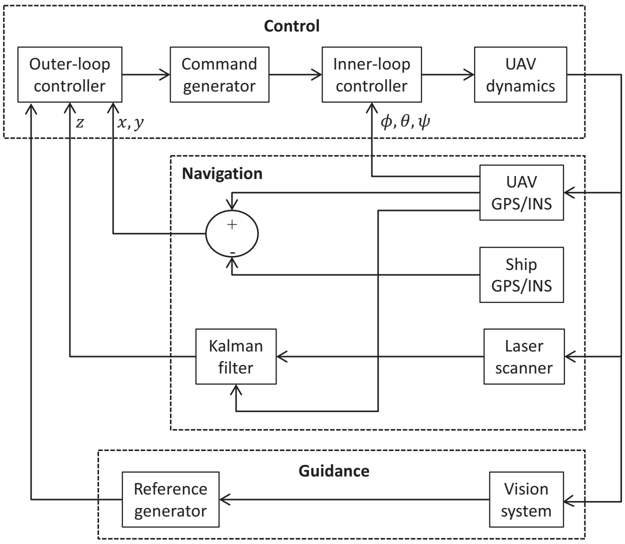

Fig. 4: System structure for guidance, navigation and control

With the hardware system ready, software algorithms need to be developed. Fig. 4 illustrates the overview of the GNC algorithms implemented on NUS^{2}T-Lion. The *control* block makes sure the UAV is stabilized in attitude and able to track 3D trajectories. The *navigation* block refines all necessary measurements by fusing raw data from various sensors. Last but not least, by implementing a vision-based target detection and tracking algorithm, the *guidance* block dynamically generates trajectories pointing towards correct destinations. The next three sections will expand these blocks in detail.

3 Modeling and Control

For all UAV applications, stability of the controlled platform is the foundation for the whole system. In this section, we briefly introduces the nonlinear model of the single rotor helicopter platform, and then proceed to control structure formulation, inner-loop control law design, and outer-loop control law design.

3.1 Nonlinear Flight Dynamics Modeling

The dynamic model of NUS^{2}T-Lion follows the same model structure introduced in [10]. There are four inputs to the system, namely the collective pitch input δ_{col}, controlling the heave dynamics, the lateral input δ_{lat}, controlling the rolling dynamics, the longitudinal input δ_{lon}, controlling the pitching dynamics, and the pedal input δ_{ped}, controlling the yawing dynamics. Small cross-couplings exist among the four channels. Outputs include the UAV global frame position $\mathbf{P}_{\mathrm{n}}$, body frame velocity $\mathbf{V}_{\mathrm{b}}$, attitude angles ϕ (roll), θ (pitch), ψ (yaw) and their respective angular rates p, q, r. Besides, a_s and b_s are the longitudinal and lateral flapping angles of the main rotor, and $\delta_{\mathrm{ped,int}}$ is an intermediate state variable in representing the yaw dynamics. Note that the last three state variables are not measurable.

Some of the model parameters can be directly measured, such as the dimensions, mass, moment of inertia of the UAV, while the others need to be identified by carrying out test-bench experiments and flight tests. The detailed nonlinear model formulation and parameter identification methods can be found in [10], and they will not be repeated here.

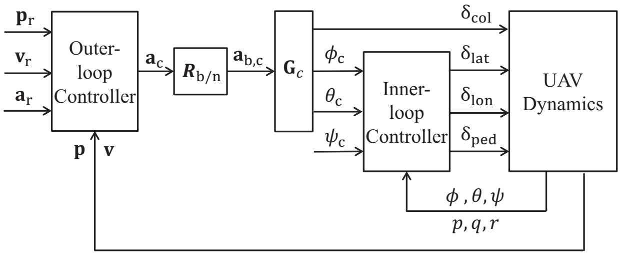

Fig. 5: Dual-loop control structure

3.2 Control Structure

In flight control engineering, a natural decomposition of the full-order dynamic model of a helicopter is based on motion types, namely rotational motion and translational motion. In general, the dynamics of rotational motion is much faster than that of the translational motion, which makes them severable in the frequency domain. Moreover, the linearized model of the single rotor helicopter system is found to be of non-minimum phase if the two motion dynamics are mixed together. This non-minimum phase characteristics will highly complicate the control problem and it is best to be avoided. Hence, it is preferable to form the control system in a dual-loop structure, and design the inner-loop and the outer-loop controllers separately.

Fig. 5 gives an overview of the proposed dual-loop control structure. The outer-loop controller generates accelerations commands in the global frame $\mathbf{a}_{\mathrm{c}}$, while the inner-loop controller is looking for attitude references (ϕ_{c}, θ_{c}, ψ_{c}). To connect the two layers, a global-to-body rotation $\mathbf{R}_{\mathrm{b/n}}$ and a command conversion $\mathbf{G}_{\mathrm{c}}$ is needed. Note that the body-axis acceleration command $\mathbf{a}_{\mathrm{b,c}}$ does not mean anything to the yaw reference ψ_{c} and the acceleration reference in the UAV body z-axis actually directly relates to the collective control input δ_{col}. If $\mathbf{G}_{\mathrm{a}}$ is the steady-state gain matrix from (δ_{col}, ϕ_{c}, θ_{c}) to the UAV body-frame accelerations, then we can get the conversion matrix $\mathbf{G}_{\mathrm{c}}$ as the inverse of $\mathbf{G}_{\mathrm{a}}$, i.e.,

$$\begin{pmatrix}\delta_{\mathrm{col}} & \phi_{\mathrm{c}} & \theta_{\mathrm{c}}\end{pmatrix}^{\mathrm{T}} = \mathbf{G}_{\mathrm{c}}\mathbf{a}_{\mathrm{b,c}} = \mathbf{G}_{\mathrm{a}}^{-1}\mathbf{a}_{\mathrm{b,c}}. \tag{1}$$

For NUS^{2}T-Lion,

$$\mathbf{G}_{\mathrm{c}} = \begin{bmatrix} 0 & 0 & 0.0523 \\ 0 & 0.1022 & 0 \\ -0.1022 & 0 & 0 \end{bmatrix}.$$

3.3 Inner-loop Control Law Design

Despite the high complication of the nonlinear model, NUS^{2}T-Lion's inner-layer linearized model is verified to be invariant under non-acrobatic flight conditions. Hence, it is reasonable to design an inner-loop control law based on the linearized model, while using the nonlinear model for simulation and verification purposes. Besides, NUS^{2}T-Lion falls into the category of small-scale outdoor helicopter which is vulnerable to wind gust disturbances. Hence, the H_∞ control method is evaluated to be the best choice. The linearized inner-dynamics model of NUS^{2}T-Lion can be represented in a 9th order state space form:

$$\begin{cases} \dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} + E\boldsymbol{w}, \\ \boldsymbol{y} = C_1\boldsymbol{x} + D_1\boldsymbol{w}, \\ \boldsymbol{h} = C_2\boldsymbol{x} + D_2\boldsymbol{u}, \end{cases} \tag{2}$$

where $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{h}$, $\boldsymbol{u}$, $\boldsymbol{w}$ are the system state, measured output, controlled output, input and wind disturbance respectively. More specifically,

$$\boldsymbol{x} = \begin{bmatrix}\phi & \theta & \psi & p & q & r & a_s & b_s & \delta_{\mathrm{ped,int}}\end{bmatrix}^{\mathrm{T}},$$
$$\boldsymbol{u} = \begin{bmatrix}\delta_{\mathrm{lat}} & \delta_{\mathrm{lon}} & \delta_{\mathrm{ped}}\end{bmatrix}^{\mathrm{T}},$$
$$\boldsymbol{w} = \begin{bmatrix}u_{\mathrm{wind}} & v_{\mathrm{wind}} & w_{\mathrm{wind}}\end{bmatrix}^{\mathrm{T}},$$
$$A = \left[\ \mathbf{0}_{9\times 3} \ \middle|\ \bar{A}\ \right],$$

where

$$\bar{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 620.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 327.6 & 0 \\ 0 & 0 & -13.5 & 0 & 0 & 165.6 \\ -1 & 0 & 0 & -5.41 & 6.45 & 0 \\ 0 & -1 & 0 & -3.72 & -5.41 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \left[\begin{array}{ccc} \multicolumn{3}{c}{\mathbf{0}_{5\times 3}} \\ \hline 0 & 0 & -54.69 \\ 2.975 & -0.3 & 0 \\ 0.780 & 3.23 & 0 \\ 0 & 0 & -4.46 \end{array}\right],$$

$$E = \left[\begin{array}{ccc} \multicolumn{3}{c}{\mathbf{0}_{3\times 3}} \\ \hline -0.0001 & 0.1756 & -0.0395 \\ 0 & 0.0003 & 0.0338 \\ -0.0002 & -0.3396 & 0.6424 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right].$$

As the onboard IMU can provide measurements of the first six state variables, C_1 can be formed accordingly and and D_1 can be left as a zero matrix. C_2 and D_2 constitute weighting parameters specifying the control objective. In our implementation, they are set in the following forms:

$$C_2 = \left[\begin{array}{ccccccccc} c_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_6 & 0 & 0 & 0 \\ \hline \multicolumn{9}{c}{\mathbf{0}_{3\times 9}} \end{array}\right], \tag{3}$$

$$D_2 = \left[\begin{array}{ccc} \multicolumn{3}{c}{\mathbf{0}_{6\times 3}} \\ \hline d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{array}\right]. \tag{4}$$

The H_∞ control problem is to find an internally stabilizing proper measurement feedback control law, such that the H_∞-norm of the overall closed-loop transfer matrix function from $\boldsymbol{w}$ to $\boldsymbol{h}$ is minimized. According to [11], the minimum H_∞-norm, γ^*, can be numerically computed. However, it is almost impossible to find a control law with finite gain to achieve this particular optimal performance. Usually, an H_∞ suboptimal controller is designed, resulting in a suboptimal H_∞-norm $\gamma > \gamma^*$. It is also proven that when the subsystem (A, E, C_1, D_1) is left invertible and of minimum phase, which is exactly the case for NUS^{2}T-Lion, the achievable H_∞ control performance under the state feedback and the measurement feedback are identical. In other words, it is appropriate to design the state feedback control law and the observer separately for this kind of systems. Moreover, only a reduced-order observer is needed to estimate the three unmeasurable state variables in this case.

Based on the procedures in [11] and selecting the following parameters for C_2 and D_2,

$$c_1 = 13,\;\; c_2 = 12,\;\; c_3 = 1,\;\; c_4 = 1,\;\; c_5 = 1,$$

$$c_6 = 6,\;\; d_1 = 13,\;\; d_2 = 12,\;\; d_3 = 30,$$

γ^* can be calculated as 0.2057. We choose an appropriate $\gamma = 0.21$, and the corresponding H_∞ suboptimal control law can be formed as follows:

$$\boldsymbol{u} = F\boldsymbol{x} + G\boldsymbol{r} \tag{5}$$

where

$$F = \begin{bmatrix} -0.9952 & -0.1177 & 0.0017 & -0.0271 & 0.0098 & 0.0143 & -1.8795 & -0.5324 & 0.0457 \\ 0.1386 & -0.9927 & -0.0005 & -0.0056 & -0.0467 & -0.0043 & 0.0253 & -1.8175 & -0.0503 \\ -0.0186 & 0.0096 & 0.0526 & -0.0006 & 0.0026 & 0.2379 & -0.0925 & -0.0216 & 1.3287 \end{bmatrix},$$

$$G = \begin{bmatrix} 0.9952 & 0.1177 & -0.0017 \\ -0.1386 & 0.9927 & 0.0005 \\ 0.0186 & -0.0096 & -0.0526 \end{bmatrix}.$$

The last three unmeasurable state variables, denoted by $\hat{\boldsymbol{x}}$, can be estimated by an observer as follows:

$$\dot{\hat{\boldsymbol{x}}} = \bar{F}\hat{\boldsymbol{x}} + \bar{G}\boldsymbol{y} + \bar{H}\boldsymbol{u} \tag{6}$$

where

$$\bar{F} = \begin{bmatrix} -0.9952 & -0.1177 & 0 \\ 0.1386 & -0.9927 & 0 \\ -0.0186 & 0 & -28 \end{bmatrix},$$

$$\bar{G} = \begin{bmatrix} 0 & 0 & 0 & -9.309 & 0.2404 & 0 \\ 0 & 0 & 0 & -1.225 & -5.106 & 0 \\ 0 & 0 & 0 & 0 & 0 & -3.452 \end{bmatrix},$$

$$\bar{H} = \begin{bmatrix} 2.975 & -0.3 & 0 \\ 0.780 & 3.23 & 0 \\ 0 & 0 & 4.78 \end{bmatrix}.$$

3.4 Outer-loop Control Law Design

A Robust and Perfect Tracking (RPT) control method is used for the outer-loop position tracking control. The controller structure and design techniques are adopted from [12]. By perfect tracking, it means the ability of the controlled system to track a given reference with arbitrarily fast settling time subjected to disturbances and initial conditions.

Since the outer-loop position outputs, x, y, z, are independent and differentially flat, stand-alone RPT controllers based on double integrator models can be designed to track the corresponding position references. For each axis, the nominal system can be written as

$$\begin{cases} \dot{\boldsymbol{x}}_{\mathrm{n}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}_{\mathrm{n}} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \boldsymbol{u}_{\mathrm{n}} \\ \boldsymbol{y}_{\mathrm{n}} = \boldsymbol{x}_{\mathrm{n}} \end{cases} \tag{7}$$

To achieve better tracking performance, it is common to include an integrator to ensure zero steady state error subjected to step inputs. This requires to form an augmented system as follows:

$$\begin{cases} \dot{\boldsymbol{x}}_{\mathrm{o}} = \begin{bmatrix} 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_{\mathrm{o}} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \boldsymbol{u}_{\mathrm{o}} \\ \boldsymbol{y}_{\mathrm{o}} = \boldsymbol{x}_{\mathrm{o}} \\ \boldsymbol{h}_{\mathrm{o}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_{\mathrm{o}} \end{cases} \tag{8}$$

where $x_{\mathrm{o}} = \begin{bmatrix} \int (p_{\mathrm{e}}) & p_{\mathrm{r}} & v_{\mathrm{r}} & a_{\mathrm{r}} & p & v \end{bmatrix}^{\mathrm{T}}$ with $p_{\mathrm{r}}, v_{\mathrm{r}}, a_{\mathrm{r}}$ being the position, velocity and acceleration references, p, v being the actual position and velocity, and $p_{\mathrm{e}} = p_{\mathrm{r}} - p$ being the position error. According to [12], a linear control law of the form below can be acquired,

$$\boldsymbol{u}_{\mathrm{o}} = F_{\mathrm{o}}\boldsymbol{x}_{\mathrm{o}}, \tag{9}$$

where

$$F_{\mathrm{o}} = \begin{bmatrix} \dfrac{k_{\mathrm{i}}\omega_n^2}{\varepsilon^3} & \dfrac{\omega_n^2 + 2\zeta\omega_n k_{\mathrm{i}}}{\varepsilon^2} & \dfrac{2\zeta\omega_n + k_{\mathrm{i}}}{\varepsilon} & 1 & -\dfrac{\omega_n^2 + 2\zeta\omega_n k_{\mathrm{i}}}{\varepsilon^2} & -\dfrac{2\zeta\omega_n + k_{\mathrm{i}}}{\varepsilon} \end{bmatrix}. \tag{10}$$

ε is a design parameter to adjust the settling time of the closed-loop system. $\omega_n, \zeta, k_{\mathrm{i}}$ are the parameters that determine the desired pole locations of the infinite zero structure of (8) through

$$p_{\mathrm{i}}(s) = (s + k_{\mathrm{i}})(s^2 + 2\zeta\omega_n s + \omega_n^2). \tag{11}$$

Theoretically, the RPT controller gives arbitrarily fast response when ε approaches zero. However, due to physical constraints of the UAV dynamics and its limited inner-loop bandwidth, it is safer to set the bandwidth of the outer loop to be one fifth to one third of the inner-loop bandwidth. For the case of NUS^{2}T-Lion, the following design parameters are used:

$$x, y : \begin{cases} \varepsilon & = & 1 \\ \omega_n & = & 0.707 \\ \zeta & = & 0.707 \\ k_{\mathrm{i}} & = & 0.25 \end{cases} \qquad z : \begin{cases} \varepsilon & = & 1 \\ \omega_n & = & 0.99 \\ \zeta & = & 0.707 \\ k_{\mathrm{i}} & = & 0.29 \end{cases}$$

4 Navigation

In designing the flight control law for NUS ^{2}T-Lion, the previous section assumes that all measurements are available and reliable. In practice, there is still missing information and unacceptable amount of measurement errors. First, the cargo platforms are moving, which requires the UAV to synchronize its motion with. Second, the UAV height measurement from GPS/INS is not accurate. If it is used blindly, the UAV may crash onto the surface of the cargo platform, resulting in catastrophic consequences. Third, the x, y planar position of the UAV also needs to be controlled extremely accurate for the cargo grasping and releasing actions to take place. While the first two problems can be solved in a navigation sense, which will be discussed in this section, the third problem is more appropriate to be solved in a guidance sense, which will be covered in Section 5.

4.1 Navigation in Ship Frame

As the cargo platforms are simulating ships, they will be called 'ships' in the following context to avoid ambiguity from the term 'platform' which is used for both the UAV platform and the cargo platforms. A reliable solution for the UAV to synchronize itself with the motion of the ship is to install another GPS/INS sensor on the ship and send its information to the UAV onboard system. By doing so, the UAV can be controlled in a ship-referenced frame instead of the global frame. In this ship-referenced frame, a zero steady-state tracking error in position and velocity means the UAV is controlled right above the ship with the same velocity, but subject to GPS/INS measurement noises.

As measurements provided by GPS/INS sensors on the UAV and on the ship are defined in the same global frame and the motion of the ship involves no rotation, it is adequate to convert all position, velocity and acceleration measurements into the ship frame by simple subtraction. So,

$$\begin{cases} \mathbf{p} = \mathbf{p}_{\text{uav}} - \mathbf{p}_{\text{ship}}, \\ \mathbf{v} = \mathbf{v}_{\text{uav}} - \mathbf{v}_{\text{ship}}, \\ \mathbf{a} = \mathbf{a}_{\text{uav}} - \mathbf{a}_{\text{ship}}. \end{cases} \tag{12}$$

If we refer back to Fig. 5, the outer-loop measurements and references are now represented in the ship frame instead of the global frame. Following this convention, it is also straight forward to convert the UAV heading angle to the ship frame as well, where

$$\psi = \psi_{\text{uav}} - \psi_{\text{ship}}. \tag{13}$$

4.2 Height Calculation via Laser Scanner

An accurate height measurement is not only needed for the cargo loading and unloading tasks, but also helpful for autonomous taking-off and landing. Motivated by this, a scanning laser range finder is installed onboard of the UAV and the corresponding algorithm to calculate the UAV height based on its range measurements is developed.

For each laser scan, an array of range measurements with different radial directions are acquired. A simple trigonometric transformation can be applied to to convert the data from polar coordinates to Cartesian coordinates. Then, the *split-and-merge* algorithm [13] is applied to divide these 2D points into clusters, with each cluster of points belonging to an individual line segment. The main steps of the *split-and-merge* algorithm is summarized below with Fig. 6 giving a graphical illustration.

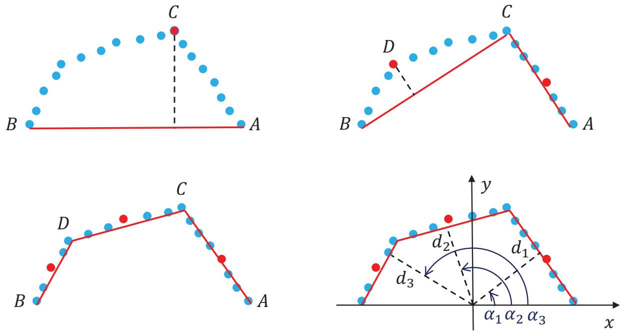

Fig. 6: The *split-and-merge* and line extraction algorithm

1) Connect the first point A and the last point B.
2) Find point C among all data points that has the longest perpendicular distance to line AB.
3) If this longest distance is within a threshold, then a cluster is created containing points between A and B.
4) Else, the input points will be split into two sub-groups, A-C and C-B. For each sub-group, the *split-and-merge* algorithm will be called recursively.

Each cluster of points will then be fitted by a line equation with the line's normal direction α_k and its perpendicular distance to the center of laser scanner d_k. After filtering out those line segments with dissimilar gradients to the ground plane, the remainings are sorted by their perpendicular distances to the laser scanner center. The furthest line segments are kept, and among them the longest one is believed to be the true ground. Finally, the UAV height can be calculated as the perpendicular distance of this line to the laser scanner center, compensated by the offset between the laser scanner and the UAV center of gravity (CG) as well as the UAV attitude angles. Using this method, an accurate height measurement can be obtained as long as the laser scanner projects a portion of its laser beams onto the true ground. It even works for the case when the UAV flies over places with scattered protruding objects.

4.3 Height Measurement Fusion

As there are two sources of height measurements, one from GPS/INS and the other from laser scanner, it is best to combine them so that the UAV state variables in the z-axis, i.e. $\boldsymbol{x}_\text{h} = \begin{bmatrix} z & w_g & a_{z,g} & \delta_z \end{bmatrix}^\text{T}$, can be optimally estimated. Here, z is the UAV vertical height with respect to the ground surface, w_g and $a_{z,g}$ are the corresponding velocity and acceleration and δ_z is the position offset between the GPS/INS measurement and the laser counterpart. This offset has to be considered because the two sensory systems have different zero references and it also accounts for the time-varying position bias of the GPS/INS sensor. In addition, we also formulate the estimator by considering the physical dynamics of a single-axis mass point system:

$$\begin{cases} \dot{\boldsymbol{x}}_\text{h} = A_\text{h}\boldsymbol{x}_\text{h} + E_\text{h}\boldsymbol{w}_\text{h} \\ \boldsymbol{y}_\text{h} = C_\text{h}\boldsymbol{x}_\text{h} + \boldsymbol{v}_\text{h} \end{cases} \tag{14}$$

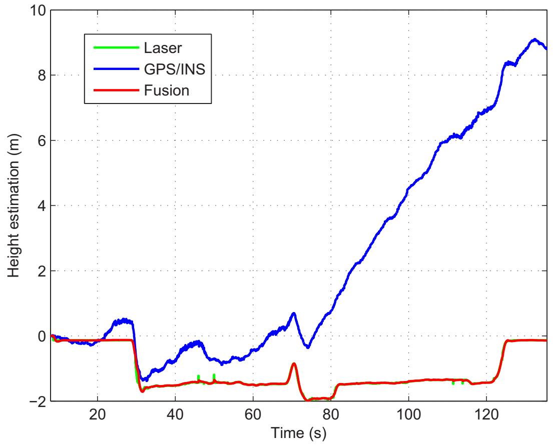

Fig. 7: Result of height estimation by data fusion

where

$$A_{\rm h}=\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&0\\0&0&0&0\end{bmatrix},\ E_{\rm h}=\begin{bmatrix}0&0\\0&0\\1&0\\0&1\end{bmatrix},\ C_{\rm h}=\begin{bmatrix}1&0&0&0\\1&0&0&1\\0&1&0&0\\0&0&1&0\end{bmatrix}$$

and $\boldsymbol{w}_{\rm h}$, $\boldsymbol{v}_{\rm h}$ are Gaussian noises with covariance matrices $Q_{\rm h}$ and $R_{\rm h}$ respectively. $Q_{\rm h}$ and $R_{\rm h}$ can be chosen by analyzing signal noise levels logged in UAV hovering flight test.

By applying the Kalman filter, a reliable estimation of UAV height can be obtained. Figs. 7 shows the height estimation result via data collected in one flight test. It can be seen that the fused result has higher quality than the original height information from GPS/INS or laser scanner alone. The slow drifting of GPS/INS and a few small outliers from laser height measurement are not present any more. At the same time, the estimated UAV vertical velocity is also less noisy than the raw GPS/INS measurement.

5 Guidance

The guidance block leads the UAV to do meaningful movements by generating mission oriented reference trajectories. It involves two main objectives in this cargo transportation task. One is to locate the cargo loading and unloading positions via vision, and the other is to generate a smooth trajectory linking the UAV current position to the destination position by considering its kinematic constraints.

5.1 Vision-based Target Localization

Vision algorithms for target detection and localization are usually mission dependent. In the context of UAVGP, the fundamental task is to identify the correct target ellipse in the captured 2D image and then estimate its pose in the physical 3D space. There are three key algorithms in our vision system, namely *ellipse detection*, *ellipse tracking* and *single-circle-based pose estimation*.

1) *Ellipse detection* has been investigated extensively in literature. Ellipse fitting, introduced in [14] is chosen as the core ellipse detection algorithm in this work, because it is very efficient compared to other hough transform based methods. Unfortunately, ellipse fitting only fits the best ellipse for a given contour without questioning whether the contour is suitable to be seen as an ellipse in the first place. To complement its shortage, a three-step procedure, consisting of pre-processing, ellipse fitting and post-processing, is proposed. The pre-processing is based on affine moment invariants (AMIs) [15], while the post-processing is based on the algebraic error between the contour and the fitted ellipse.

2) *Ellipse tracking* is to continuously track a single ellipse after its detection has been initialized. In the UAVGP setup, multiple ellipses may be detected in an image but only one of them is the true target. There are two challenging problems. First, the areas enclosed by the ellipses are exactly the same in both shape and color. Thus, template matching based on color, shape or feature points may not be suitable. Second, when implementing vision-based tracking algorithms on a flying platform, the fast dynamic motion of the UAV may cause large displacement of the target ellipse between two consecutive images. In order to track the target ellipse robustly, the frame rate of the image sequence must be high, which requires a very efficient tracking algorithm. As a result, the CAMShift method [16] is chosen here. It runs very fast and can track the target ellipse even when its scale, shape and color keep changing.

3) *Single-circle-based pose estimation* is to calculate the 3D position of the target circle after its projected ellipse on the 2D image has been identified. Circle-based camera calibration and pose estimation have been studied in [17, 18]. However, these studies mainly focused on the cases of concentric circles, but our aim is to do pose estimation via only one circle. Theoretically, it is impossible to estimate the pose of a single circle purely from its perspective projection. However, from a practical point of view, it can be solved by adopting a reasonable assumption that the image plane of the camera is parallel to the plane that contains the circle. This assumption is fulfilled in our implementation because the onboard camera is installed on a pan-tilt mechanism which can be actively controlled to align with the ground plane.

Due to its own research significance in the computer vision society, more detailed discussion about this vision-based target detection and localization algorithm are documented in [19]. Fig. 8 shows a number of consecutive images taken by the onboard camera. In each image, the green ellipse is the target ellipse tracked by the vision algorithm, while the yellow one is the area of interest returned by the CAMShift algorithm. It can be seen that the target ellipse has been correctly detected and tracked even when its scale or shape varies.

5.2 Trajectory Planning

After the target location in the camera frame is obtained, it needs to be transformed to the ship orientation for trajectory planning. For each axis, the trajectory planner needs to know the current UAV velocity v_0, the maximum allowed velocity $v_{\max}$, the maximum allowed acceleration $a_{\max}$ and the displacement from the current position to the final position S. Fig. 9 illustrates the basic idea of the proposed algorithm. If

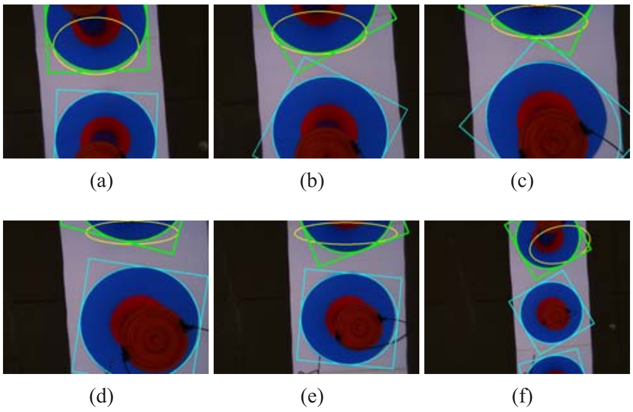

Fig. 8: Onboard images with ellipse detection and tracking

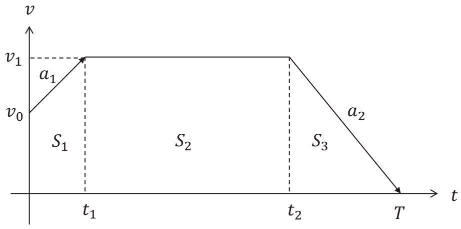

Fig. 9: Trajectory planning with continuous velocity

v_0, a_1, a_2 and T are known, t_1 has a conditional closed-form solution as follows:

$$\begin{cases} t_1 = -C/B & \text{if} \quad A = 0, \\ t_1 = \dfrac{-B - \sqrt{D}}{2A} & \text{if} \quad AC > 0 \ \& \ AB < 0 \ \& \ D > 0, \\ t_1 = \dfrac{-B + \sqrt{D}}{2A} & \text{if} \quad AC < 0 \ \& \ D > 0, \\ t_1 = -1 & \text{if} \quad \text{otherwise}, \end{cases}$$

where

$$\begin{cases} A &= a_1^2 - a_1 a_2, \\ B &= 2v_0 a_1 + 2a_1 a_2 T, \\ C &= v_0^2 + 2a_2 v_0 T - 2a_2 S, \\ D &= B^2 - 4AC. \end{cases}$$

Correspondingly,

$$t_2 = T + \frac{v_0 + a_1 t_1}{a_2}.$$

However, a_1, a_2 and T are not known exactly. To solve this problem, a recursive algorithm by listing all four cases of a_1-a_2 combination and repeatedly increasing T by a 1-second step is proposed as Fig. 10. The iteration stops until a feasible solution occurs.

6 Experimental and Competition Results

The aforementioned GNC algorithms have been implemented onboard of NUS^{2}T-Lion and modified towards the competition requirements. Figs. 11–13 show the position data logged in one of the flight tests. As the logged data is obtained from GPS/INS, it may not be as accurate as the ground truth. Nevertheless, it still verifies the general flight performance and indicates whether the UAV is doing the right movements. In Fig. 11, the x position becomes larger progressively because the UAV is moving from the nearest cargo to the the furthest with respect to its take-off position. It always comes back to a position around zero because the reference path is designed in a way that the onboard camera has the best view of both ships before every loading or unloading dive. In Fig. 12, the y position signal goes back and forth, indicating alternative movements between the two ships. In Fig. 13, it is clear to see all the diving motions of the UAV from its change in vertical displacement. The UAV will stay at a low altitude with a variable time duration depending on how many loading or unloading trials have been performed.

With such performance, NUS^{2}T-Lion has successfully accomplished all tasks in the UAVGP rotary-wing competition. A final score of 1127.56 with 472.44 from the preliminary contest and 655.13 from the final has made the team second position in the overall Grand Prix. In fact, 655.13 is the highest score in the final round. It should be highlighted that unlike the preliminary contest in which the two ships are stationary, the final round requires the UAV to carry out

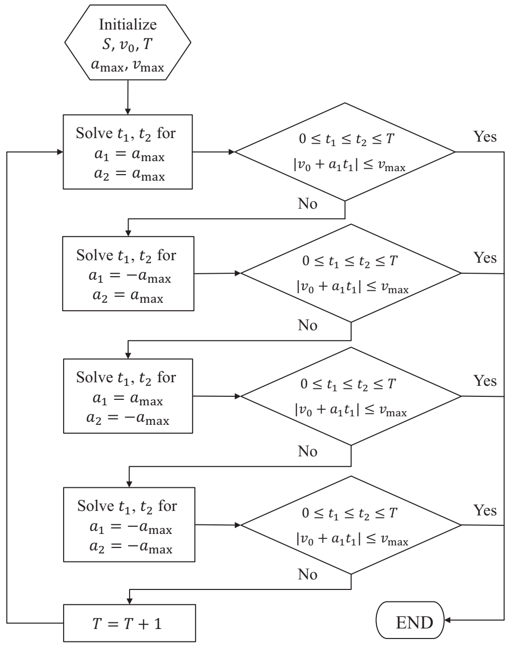

Fig. 10: Flowchart of the trajectory planning algorithm

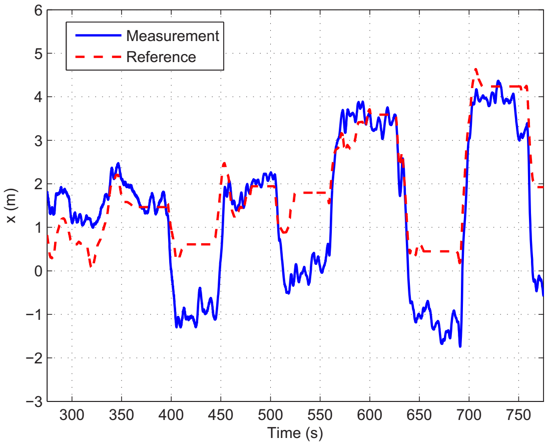

Fig. 11: Flight test result of x-axis position

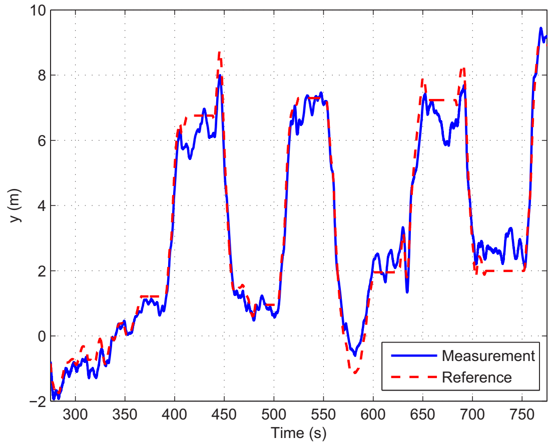

Fig. 12: Flight test result of y-axis position

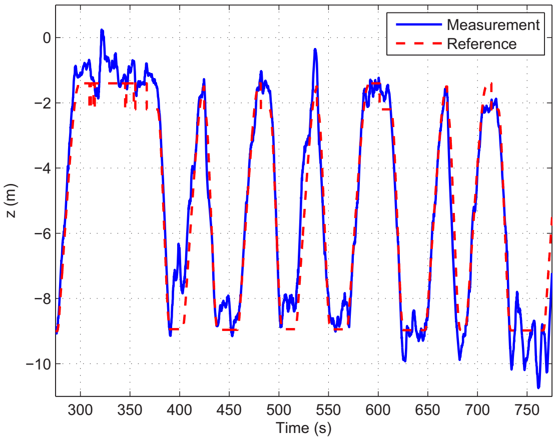

Fig. 13: Flight test result of z-axis position

cargo transportation with the two ships moving. This demands for higher intelligence and robustness from the UAV system, and it is indeed the strongest point of our solution. The Chinese version video for this final flight can be found at `http://v.youku.com/v_show/id_XNjI5MTM2MDI0.html`.

7 Conclusion

We have presented a comprehensive design and development methodology in solving the rotorcraft UAV cargo transportation problem. Besides the hardware innovations, the main contributions of our work are on UAV guidance, navigation and control algorithms. One highlight of this work is the integration of the UAV navigation system with a vision-based guidance system. Further studies are needed to make the system more intelligent and able to work in harsher conditions such as completely GPS-less environments and transporting cargos in between ships with wobbling decks caused by sea waves. This is one of the ongoing projects that our group is conducting at moment.

References

[1] A. Bujak, M. Smolarek, and A. Gebczyǹska, "Applying military telematic solutions for logistics purposes," in *11th International Conference on Transport Systems Telematics*, pp. 248–256, 2011.

[2] M. Bernard and K. Kondak, "Generic slung load transportation system using small size helicopters," in *IEEE International Conference on Robotics and Automation*, pp. 3258–3264, 2009.

[3] M. Bisgaard, A. la Cour-Harbo, and J. D. Bendtsen, "Adaptive control system for autonomous helicopter slung load operations," *Control Engineering Practice*, vol. 18, no. 7, pp. 800–811, 2010.

[4] M. Orsag, C. Korpela, and P. Oh, "Modeling and control of MM-UAV: Mobile manipulating unmanned aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1-4, pp. 227–240, 2013.

[5] J. Thomas, J. Polin, K. Sreenath, and V. Kumar, "Avian-inspired grasping for quadrotor micro uavs," in *IDETC/CIE*, ASME, 2013.

[6] B. K. G. Mrdjan Jankovic, "Visually guided ranging from observations of points, lines, and curves via identifier based nonlinear observer," *System & Control Letter*, vol. 25, no. 1, pp. 63–73, 1995.

[7] S. Saripalli, J. F. Montgomery, and G. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," in *IEEE International Conference on Robotics and Automation*, 2002.

[8] 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, pp. 1038–1049, 2012.

[9] P. Vela, A. Bester, J. Malcolm, and A. Tannenbaum, "Vision-based range regulation of a leader follwer formation," *IEEE Transaction on Control Systems Technology*, vol. 17, no. 2, pp. 442–448, 2009.

[10] G. Cai, B. Chen, and T. Lee, *Unmanned Rotorcraft Systems*. New York: Springer, 2011.

[11] B. M. Chen, *Robust and H_∞ Control*. Communications and Control Engineering Series, Springer, 2000.

[12] B. M. Chen, T. H. Lee, and V. Venkataramanan, *Hard Disk Drive Servo Systems*. Advances in Industrial Control Series, New York: Springer, 2002.

[13] G. A. Borges and M. J. Aldon, "A split-and-merge segmentation algorithm for line extraction in 2d range images," in *15th International Conference on Pattern Recognition*, pp. 441–444, 2000.

[14] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 476–480, May 1999.

[15] J. Flusser and T. Suk, "Pattern recognition by affine moment invariants," *Pattern Recognition*, vol. 26, pp. 167–174, January 1993.

[16] J. G. Allen, R. Y. D. Xu, and J. S. Jin, "Object tracking using CamShift algorithm and multiple quantized feature spaces," in *Proceedings of the Pan-Sydney area workshop on Visual information processing*, (Darlinghurst, Australia), pp. 3–7, 2004.

[17] J.-S. Kim, P. Gurdjos, and I.-S. Kweon, "Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 637–642, 2005.

[18] D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart, "Vision based position control for MAVs using one single circular landmark," *Journal of Intelligent and Robotic Systems*, vol. 61, pp. 495–512, 2011.

[19] S. Zhao, Z. Hu, M. Yin, K. Ang, P. Liu, F. Wang, X. Dong, F. Lin, B. M. Chen, and T. H. Lee, "A robust vision system for a uav transporting cargoes between moving platforms," in *Proceedings of the 2014 Chinese Control Conference*, 2014.