Development of an Unconventional Unmanned Coaxial Rotorcraft: GremLion*

Feng Lin, Kevin Z.Y. Ang, Fei Wang, Ben M. Chen, Tong Heng Lee, Beiqing Yang, Miaobo Dong, Xiangxu Dong, Jinqiang Cui, Swee King Phang, Biao Wang, Delin Luo, Shiyu Zhao, Mingfeng Yin, Kun Li, Kemao Peng, and Guowei Cai

UAV Research Group, National University of Singapore, Singapore 117576

Abstract. In this paper, we present an unmanned system design methodology for a fully functional unmanned rotorcraft system: GremLion, developed with all necessary avionics and a ground control station. It has been employed to participate in the 2012 UAVForge competition. The proposed design methodology consists of hardware construction, software development, dynamic modeling and flight control, as well as mission algorithms. The test results have been presented in this paper to verify the proposed design methodology.

1 Introduction

In the last two decades, unmanned systems aroused great interests world wide [1, 2], especially advanced micro unmanned aerial vehicle (UAV) systems capable of vertical take-off and landing, beyond line of sight observations, autonomous obstacle avoidance in cluttered environments and much more [3, 4, 5, 6]. These capabilities could provide researchers, rescuers, and other users a new and valuable tool.

To boost the progress in urban UAV development, in year 2012, the Defense Advanced Research Projects Agency (DARPA) and Space and Naval Warfare Systems Center Atlantic (SSC Atlantic) collaboratively launched an initiative called the UAVForge competition to design, build and manufacture advanced micro unmanned air vehicle systems.

To participate in the UAVForge competition, the NUS UAV research group started to design and develop an unmanned coaxial rotorcraft: GremLion, together with necessary avionics and a ground control station. Here, we propose the design methodology which enabled us to efficiently develop the GremLion system comprising of the rotorcraft, avionics, software system, ground control station, flight control system and mission system.

2 UAVForge Competition

In the UAVForge competition, the mission of each team is to outfit a fictional Task Force with an unmanned remotely operated micro air vehicle system. The entire air vehicle

* This research is funded by DSO National Laboratories, Singapore.

A. Marcus (Ed.): DUXU/HCII 2013, Part III, LNCS 8014, pp. 120–129, 2013.

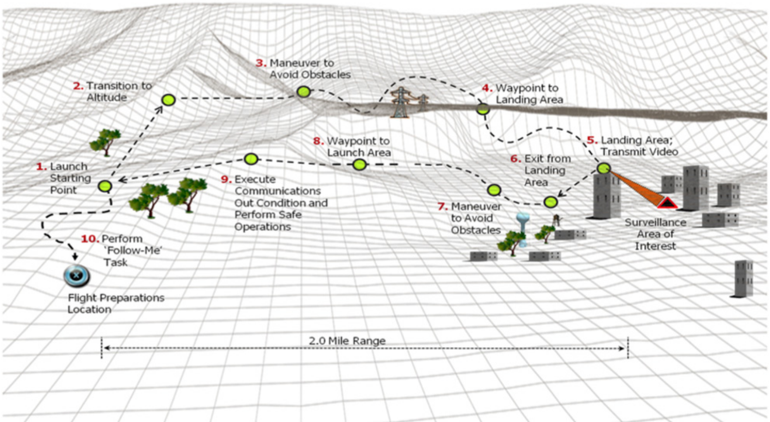

Fig. 1. UAVForge Competition Course

system must fit within a rucksack and a single person traveling by foot must be able to carry and operate the vehicle without assistance.

The job of the Task Force is to conduct observations of suspicious activities occurring within the vicinity of two nondescript buildings in an urban area. Due to the security in the region, all operations must be conducted beyond line of sight so as not to compromise your presence. If the UAV system is detected, the mission will be jeopardized. The total observation time required may be up to three hours of pictures and/or video to document the surveyed area. Once key observations have been made, the team must quickly retreat to their designated rendezvous location. Fig. 1 outlines the overall course of the competition.

3 The Coaxial Helicopter

GremLion, shown in Fig. 2, features a coaxial design driven by two contra-rotating rotors that can compensate the torque due to aerodynamic drag. Coaxial rotor designs

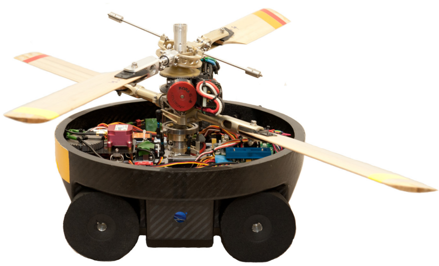

Fig. 2. GremLion

Table 1. Main specifications of GremLion

Specifications	GremLion
Upper rotor span	798 mm
Lower rotor span	895 mm
Upper rotor speed	1900 rpm
Lower rotor speed	1700 rpm
No-load weight	2.4 kg
Maximum takeoff weight	5.1 kg
Power source	LiPo battery
Flight endurance	15 mins

allow for a more stable, more maneuverable, quieter and safer helicopter due to inclusion of a coaxial main rotor and exclusion of a tail rotor. Coaxial rotor design also means a smaller footprint. Coaxial rotor helicopters also provide a better power to weight ratio than traditional helicopters, produce greater lift and are also much more efficient. Therefore, this platform is suited for the size requirement of the competition, which can be kept in a rucksack. Its key specifications are listed in Table 1.

To reduce the mechanical complexity of conventional dual-swash plate designs, a novel actuation system has been employed in GremLion with a single swash plate linked to the lower rotor system, which is shown in Fig. 3. The operating principle of such a actuation system is presented as follows:

a. **Heave Channel**: Unlike the conventional single rotor helicopters which utilize the collective pitch of their rotor blades to adjust the lift force, GremLion's collective pitches are fixed and thrust variation is accomplished by changing the rotor spinning speed simultaneously.
b. **Yaw Channel**: The yaw motion (head turning) is produced by the difference of spinning speed between the top and bottom rotors. In order to stabilize the heading of the rotorcraft, a hardware rate gyro is installed to finely adjust the spinning speed of the two rotors so that yaw dynamics becomes much more damped.
c. **Lateral and Longitudinal Channels**: To have lateral and longitudinal motions, the bottom rotor cyclic pitch is actively controlled by three servos. This is done through a swash plate mechanism which acts as a link between the servos and the bottom rotor cyclic pitch. The aileron and elevator inputs cooperate with the roll and pitch rate feedback controller to stabilize the angular rate of roll and pitch motion.
d. **Mechanical Stability Augmentation**: The top rotor is not actively linked to any servos, but it is passively controlled via a mechanical stabilizer bar. This slows down the whole platform's response to the rapid changes in the cyclic pitch of the bottom rotor.

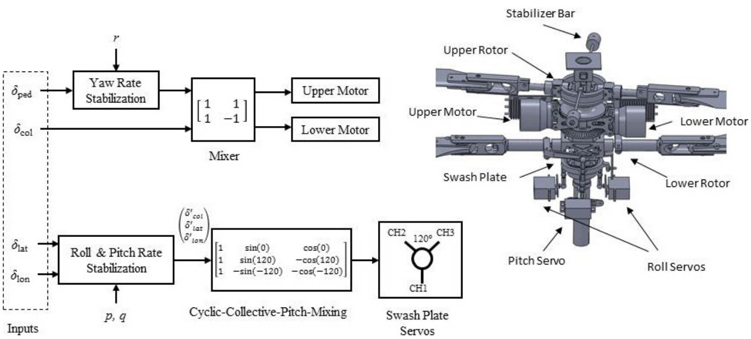

Fig. 3. Operating principle of the actuation system

4 Avionic System

The avionic system has been developed to realize fully autonomous flight. The following key components have been selected, which are the most suitable commercial off-the-shelf (COTS) products to date.

1. **Navigation Sensors.** IG-500N is currently the world smallest GPS enhanced attitude and heading reference system (AHRS) embedded with extended Kalman filter (EKF). It includes a MEMS-based inertial measurement unit (IMU), a GPS receiver and a pressure sensor. It is able to provide precise and drift-free 3D orientation and position even in aggressive maneuvers, updated at 100 Hz.
2. **Onboard Computer.** The onboard processor is the brain of the whole avionic system. It collects measurement data from various sensors, does filtering and fusion, executes flight control laws, and output control signals to carry out the desired control actions. In addition, it is also responsible for communicating with the GCS for real-time inspection and command issuing, as well as logging in-flight data for after-flight analysis. We have chosen two Gumstix Overo Fire embedded computers for flight control, navigation and vision processing purposes respectively. It has Wi-Fi functionality despite its small size and weight. In order to improve its real-time performance, the original Linux operating system provided by the manufacturer is replaced by the QNX Neutrino Real Time Operating System (RTOS).
3. **Servo Controller.** The UAV100 is an 8 channel radio control (RC) PPM servo controller and 8 channel PPM servo receiver that is designed to allow ground personnel to take over the control of the UAV at the flick of a switch to prevent a catastrophic failure from a malfunction in the flight computer.
4. **Communication.** The communication unit includes a pair of Microhard wireless data transceivers. This pair of transceivers establish communication between the onboard system and the ground station. They are configured to operate in point-to-point mode and works in the 2.400 to 2.4835GHz range.

System integration is not a trivial task in small-scale UAV development. We proposed a simple and uniform design approach, which is independent of the hardware components used and can be easily adopted to construct any small-scale UAVs [1]. Based on this method, essential mechanical parts and all related avionic components have been assembled onto the vehicle. The integrated components resulted in the final integrated platform which is shown in Fig. 2. This platform have been extensively used in test flights for model identification and verification.

5 Onboard Software and Ground Control Station

Based on the developed hardware system, a framework of a UAV software system is proposed which consists of two main components: onboard system and ground control system.

The UAV onboard system has six main modules: simulation model, sensing (sensor data acquisition and processing), flight control (automatic navigation and control), wireless communications (vehicle-vehicle and vehicle-ground communications).

The sensor data acquisition, navigation and control and UAV dynamics construct the control loop. Specifically, a UAV model is built in the onboard to realize hardware-in-the-loop simulation. Besides, the vehicle-vehicle communication is applied for cooperative data exchange to feed to the cooperative control module to realize UAV team cooperative control, such as formation flight control. In addition, the flight status data of each UAV is transmitted back to ground station via the vehicle-ground station communication. While the user can send commands to each UAV with this communication link. Sensors come into play apart from the traditional inertial navigation system (INS) with GPS, ultrasonic sensor can be used for landing where precise height measurement is necessary. Currently, with the development of image processing, the onboard camera becomes a more important role in target detection, tracking and many other applications. For detailed information of software development for UAVs, please refer to [7].

The GCS is composed of background tasks and foreground tasks. The background layer has mainly two tasks, receiving flight status from and sending commands to multiple UAVs, both of which interact with the UAV onboard communication task. The receiving thread accepts all the data from the fleet of UAVs and identify each status data via the telegraph packet header. Consequently, the corresponding multiple display is executed, and the cooperative way points of the paths are demonstrated. Similarly, the upload link can broadcast the commands to all UAVs, or alternatively send commands to a specific UAV, both via the sending task. The global status data from UAVs are dynamically updated from the background layer. The foreground task composes of information monitoring and task management, where the information monitoring module consists of various user-friendly views. A document class implementation in MFC is deployed to realize the communication between the background tasks and foreground tasks. The document class performs the flight data storage (up to 2000 updates), data processing (rotation computation in 3D view), command interpreting and packaging, and etc.

Specifically, based on our previous development for single UAVs, we incorporate the Google Maps view to better demonstrate the cooperative behaviors of the fleet of multiple UAVs. We captured several maps from Google Earth where we will conduct outdoor flight test and recorded the GPS data on the corners of the map. In the flight test, the GPS signal from the onboard system will be updated on the global shared data, and the cooperative paths of multiple UAVs are displayed on the Google Map way point view. For indoor flight tests, since the GPS signal is not available, we can manually set the position information to simulate this functionality in the way point view.

6 Dynamics Modeling

In order to systematically design a flight control law for the GremLion platform with good performance, an accurate mathematical model reflecting the flight dynamics of the air vehicle needs to be derived. To obtain this model, two approaches can be considered. One is called the first-principle modeling method which focuses on direct mathematical formulation of the system based on the law of physics, while the other one is called the system identification method which numerically estimates the parameters of a system with sufficient in-flight data. Although both approaches have shown their individual successes in literature, using either of them alone is not good enough to generate a model with good fidelity for the full envelope of UAV flight. Therefore, for the modeling of GremLion, the above two methods will be used in a complementary way.

Quite a few assumptions based on the facts of near-hovering condition has been made to reduce the model complexity. First of all, linear acceleration, linear velocity, angular rates, and roll pitch angles are all near zero when hovering, thus terms involving the second order of these variables can be dropped off when deriving mathematical expressions. Second, we assume very fast response of servos and motors (i.e. the response time from control inputs to the change of servo positions or change of speed of motors is much faster than the UAV dynamics). Third, although the coaxial helicopter flies with the top and bottom rotors pitching cyclically at different angles, it is reasonable to look at the two rotors as a whole system and model them as a single imaginary rotor with the so-called resultant longitudinal and lateral angles expressed as a_s and b_s. With this assumption, the model can be simplified to a large extent, yet maintaining good fidelity when the helicopter does not do aggressive maneuvering.

A reasonable linear model of GremLion flying at the near-hover condition with the coupling terms can be represented in the following state-space form:

$$\dot{\mathbf{x}} = A_{\text{id}}\mathbf{x} + B_{\text{id}}\mathbf{u} \tag{1}$$

where $\mathbf{x} = \mathbf{x}_{\text{act}} - \mathbf{x}_{\text{trim}}$ is the difference between the actual state variables and their trimmed values, and similarly, $\mathbf{u} = \mathbf{u}_{\text{act}} - \mathbf{u}_{\text{trim}}$, which are respectively given as

$$\begin{aligned}\mathbf{x} &= [u,\ v,\ p,\ q,\ \phi,\ \theta,\ a_s,\ b_s,\ w,\ r]^{\mathrm{T}}\\ \mathbf{u} &= [\delta_{\text{lat}},\ \delta_{\text{lon}},\ \delta_{\text{col}},\ \delta_{\text{ped}}]^{\mathrm{T}}\end{aligned}$$

Where $[u,\ v,\ w]^{\mathrm{T}}$ are the body-axis velocity in x, y and z directions, $[p,\ q,\ r]^{\mathrm{T}}$ are the angular velocity in x, y and z axes, $[a_s,\ b_s]^{\mathrm{T}}$ are the equivalent longitudinal and lateral flapping angles of the top and bottom rotors together. $[\delta_{\text{lat}},\ \delta_{\text{lon}}, \delta_{\text{col}}, \delta_{\text{ped}}]^{\mathrm{T}}$ are the inputs to the system (aileron, elevator, throttle, rudder). And A_{id} and B_{id} matrices are derived in the near hovering condition by the identification of the unknown model parameters which could be obtained by doing manual flight tests and giving frequency sweeping signals (sinusoidal signals with various frequencies) to the four input channels.

7 Navigation and Control

In flight control engineering, a natural stratification of the full-order dynamic model of a helicopter is based on motion types, i.e. rotational motion and translational motion. In general, the dynamics of rotational motion is much faster than that of the translational motion. Thus, the controlled object can be divided into two parts and the overall control system can be formulated in a dual-loop structure. In this way, inner-loop and outer-loop controllers can be designed separately.

The main task of the inner-loop controller is to stabilize the attitude and heading of GremLion in all flight conditions. H_∞ technique is preferred for robust stability. For the outer loop, the controlled object covers only the translational motion. The main task is to steer the UAV to fly with reference to a series of given locations. A robust and perfect tracking (RPT) approach is implemented for the outer-loop since time factor is important. It should be noted that both control laws are designed using the asymptotic time-scale and eigenstructure assignment (ATEA) method, which is fully developed for MIMO LTI systems by Chen et al. [8]. It makes the design process very systematic and effective. To give an overall view, the dual-loop control structure is shown in Fig. 4.

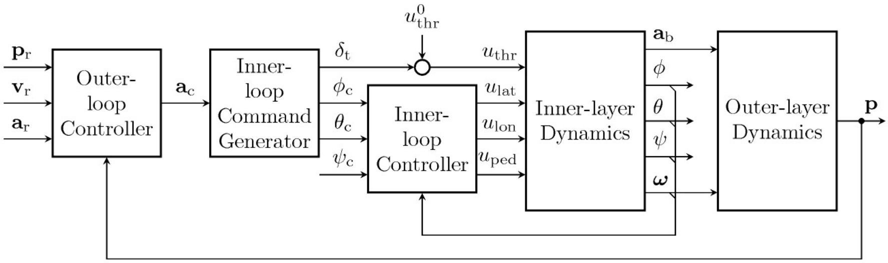

Fig. 4. Dual-loop Structure of Flight Control System

7.1 Inner-Loop Control Design

The inner-layer dynamics is a 8$^{\text{th}}$-order MIMO system with three control inputs, namely u_{lat}, u_{lon}, and u_{ped}. The uninvolved fourth input, u_{thr}, is reserved for control of vertical motion and needs be set at its trimming value (denoted as u^0_{thr}) at this stage. For the measurement part, IMU gives ϕ, θ, ψ, p, q, and r. The other two state variables (i.e. the flapping angles b_s, a_s) have to be estimated by an observer. Therefore, the linearized inner-layer controlled object can be formulated from Eqn. 1 as

$$\begin{cases} \dot{\mathbf{x}} = A_{\text{in}}\mathbf{x} + B_{\text{in}}\mathbf{u} + E_{\text{in}}\mathbf{w} \\ \mathbf{y} = C_{\text{in},1}\mathbf{x} + D_{\text{in},11}\mathbf{u} + D_{\text{in},1}\mathbf{w} \\ \mathbf{z} = C_{\text{in},2}\mathbf{x} + D_{\text{in},2}\mathbf{u} + D_{\text{in},22}\mathbf{w} \end{cases} \tag{2}$$

where $\mathbf{x} = [\phi\ \theta\ \psi\ p\ q\ r\ b_s\ a_s]^{\mathrm{T}}$ is the inner-loop state variables. $\mathbf{y} = [\phi\ \theta\ \psi\ p\ q\ r]^{\mathrm{T}}$ is the measured output vector, $\mathbf{z} = [\phi\ \theta\ \psi]^{\mathrm{T}}$ is the controlled output vector, and all variables are the deviations from their trimming values. Note that the direct feed through matrices $D_{\text{in},11}$ and $D_{\text{in},2}$ are both zero. No external disturbance is considered for this part of model at the current stage, so the disturbance input matrix E_{in} and the feed through matrices $D_{\text{in},1}$, $D_{\text{in},22}$ are all empty. They are reserved in the expression for integrity so that external disturbances such as wind gusts can be considered in future. The controlled subsystem characterized by quadruple (A_{in}, B_{in}, $C_{\text{in},2}$, $D_{\text{in},2}$) is both observable and controllable. By transforming the quadruple into the special coordinate basis (SCB) form [8], we find that the subsystem is invertible and of minimum phase. Hence, we can design an H_∞ controller via the ATEA method using state feedback to obtain robust stability. Matrix F_{i} is the state feedback gain, Matrix G_{i} is the corresponding reference feed forward gain to make sure the ratio between output and reference is unity.

7.2 Outer-Loop Control Design

The outer-loop control can then be designed separately and based on the dynamic model GremLion's translational motion only, provided that the outer loop is slow enough as compared to the inner loop. Furthermore, the outer-loop control signals are all defined in the North-East-Down (NED) frame and for all three directions, the dynamics are approximately formulated as double integrators. So,

$$\begin{cases} \dot{\mathbf{x}} = A_{\text{out}}\mathbf{x} + B_{\text{out}}\mathbf{u} + E_{\text{out}}\mathbf{w} \\ \mathbf{y} = C_{\text{out},1}\mathbf{x} \\ \mathbf{z} = C_{\text{out},2}\mathbf{x} \end{cases} \tag{3}$$

where,

$$\mathbf{x} = [x\ y\ z\ u\ v\ w]^{\mathrm{T}}\ ,\ \mathbf{y} = [x\ y\ z\ u\ v\ w]^{\mathrm{T}}\ ,\ \mathbf{z} = [x\ y\ z]^{\mathrm{T}}\ ,\ \mathbf{u} = [ac_x\ ac_y\ ac_z]^{\mathrm{T}}$$

Since the translational motion in these three directions are largely decoupled (inner-loop should have decoupled them if designed correctly), the RPT control laws for these three channels can be designed separately. Since they are all standard 2nd order systems, by choosing an appropriate natural frequency and damping ratio, they should be able to achieve desired performance. Of course, minor tuning is needed after trial flight tests have been carried out.

7.3 Inner-Loop Command Generator

We have designed the inner-loop and the outer-loop controllers separately to avoid the non-minimum phase problem and to relieve task complexity. To preserve the overall system stability, the closed outer loop should be slower than the closed inner loop. In this case, the closed inner loop can be seen as a static gain when combining with the outer loop. We approximated an inner-loop command generator from the outer-loop controller output $\mathbf{a}_{\text{c}}$,

$$\left[\delta_{\text{t}}\ \phi_{\text{c}}\ \theta_{\text{c}}\right]^{\mathrm{T}} = G_{\text{c}}\mathbf{a}_{\text{c}} = G_{\text{a}}^{-1}\mathbf{a}_{\text{c}} \tag{4}$$

Notice that G_{a} must be non-singular otherwise $\mathbf{a}_{\text{b}}$ cannot be manipulated by the control inputs u_{thr}, u_{lat}, u_{lon}. Flight tests show that this inner-loop command generator G_{c} is feasible.

7.4 Flight Results

One of the flight test results in the UAVForge competition has been shown in Fig. 4, which consists of several key flight modes, e.g. hovering, ascending, forward flight, etc. This test result verified the proposed flight control design and onboard avionics. The detailed path generation is as follows.

The proposed control law was sufficiently stable and GremLion should be able to finish the required 2 mile flight. Unfortunately, the electronic speed controller (ESC) overheated when GremLion began its forward flight. The ESCs cut off temporarily and started functioning after a short moment. The consequence was disastrous as the platform could not stand such a big and sudden disturbance. The top and bottom rotors

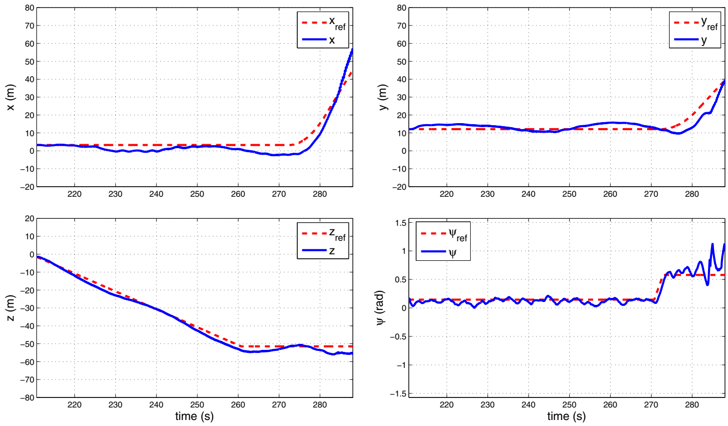

Fig. 5. Flight results

struck each other and followed by loss of lift. GremLion crashed into the trees and our attempt was suspended. Nonetheless, the on-board recorded data had been saved and transferred to PC and Fig. 5 show various signals just before the crash.

8 Vision-Based Obstacle Detection and Avoidance

In order to perform autonomous navigation in unknown outdoor environments, the UAV should have the capability to detect and avoid obstacles, e.g. trees, electrical cables, buildings, etc., in its flight path. We propose a depth-based obstacle detection algorithm using vision sensing. We chose to use vision sensors because monocular cameras are light-weight and low priced and can provide rich information of the environment. Our obstacle detection method is based on 3D vision techniques, more specifically, optical flow. Optical flow can provide the velocity of features on the images.

The main steps in our algorithm are feature extraction, feature matching, and depth estimation. The purpose to achieve feature extraction and matching is to obtain the feature position on the image, and to calculate the feature velocity on the image. The idea behind our depth estimation is based on structure from motion. More precisely, if the state (i.e., position, velocity and attitude) of the UAV can be measured, e.g., by GPS and inertial sensors, and the image of a 3D point can be matched between two images, then the 3D position of the 3D point can be determined. In our work, since we are more interested in the distance between the obstacles and the UAV, we focus on how to estimate the depth of the feature points since we can obtain the state of the UAV using GPS and inertial sensors. If the depth of certain parts of the scene is less than a prescribed threshold, then that part will be classified as an obstacle and obstacle avoidance procedures will be activated.

9 Vision-Based Target Tracking

The UAVForge competition requires UAVs to perform a series of advance behaviors and one of them is to execute the “Follow Me” task, which requires the vehicle to autonomously follow the ground operator from the Starting Point back to the designated Flight Preparation Area and maintain a safe altitude.

The strategy used to solve the above problem statement is to implement mono-camera target tracking of the required vehicle. As the target mentioned in the above statement is that of a vehicle traveling at about 15-30 mph, it is necessary to firstly mark the target that is required to be tracked. It is achieved by using a mono-camera sensor to capture an image of the target vehicle and drawing a rectangular target box around the required target. Next, the selected target box is extracted from the image and training of the target is implemented using the Continuously Adaptive Mean-SHIFT(CAMSHIFT) algorithm. Assuming that the mono-camera sensor on the GremLion UAV is stable, the movement of the target could be tracked by observing the new object center calculated from the algorithm. The GremLion UAV is then controlled to maintain the tracked object in the center of the image captured from the camera.

10 Conclusion and Future Work

In this paper, the development and implementation of GremLion has been presented. The flight test results of the GremLion in the UAVForge competition have been presented too. Further experiment and research work is required to obtain a reliable and accurate dynamic model of GremLion in full envelope condition. That is also important for the automatic control law design. In addition, we have implemented the vision-based algorithms on the onboard vision computer, including obstacle avoidance and target tracking. These algorithms will be tested in further experiments, though we did not test them in the competition.

References

[1] Cai, G., Chen, B.M., Lee, T.H.: Unmanned Rotorcraft Systems, 1st edn. Springer, New York (2011)

[2] Cai, G.W., Lin, F., Chen, B.M., Lee, T.H.: Development of fully functional miniature unmanned rotorcraft systems. To be presented at the 29 Chinese Control Conference, Beijing, China (2010)

[3] Sarris, Z., Atlas, S.: Survey of UAV applications in civil markets. In: Proceedings of the 9th Mediterranean Conference on Control and Automation, Dubrovnik, Croatia, vol. WA2-A (2001)

[4] Enderle, B.: Commercial Applications of UAV's in Japanese Agriculture. In: Proceedings of the AIAA 1st UAV Conference, Portsmouth, Virginia (2002) AIAA-2002-3400

[5] Ludington, B., Johnson, E., Vachtsevanos, G.: Augmenting UAV autonomy. IEEE Robotics & Automation Magazine 13, 63–71 (2006)

[6] Campbell, M.E., Whitacre, W.W.: Cooperative tracking using vision measurements on seascan UAVs. IEEE Transactions on Control Systems Technology 15, 613–626 (2007)

[7] Dong, X., Chen, B.M., Cai, G., Lin, H., Lee, T.H.: A comprehensive real-time software system for flight coordination and cooperative control of multiple unmanned aerial vehicles. International Journal of Robotics and Automation 26(1), 49–63 (2011)

[8] Chen, B.M.: Robust and H_∞ Control. Springer, London (2000)