Skoltech
Skolkovo Institute of Science and Technology

# Thesis Changes Log

**Name of Candidate:** Grigoriy Yashin

**PhD Program:** Engineering Systems

**Title of Thesis:** Development of group of flying robots with multifunctional robotic limbs aimed at operations in cluttered environments

**Supervisor:** Associate Professor Dzmitry Tsetserukou

*The thesis document includes the following changes in answer to the external review process.*

**Reviewer: Andrzej Cichocki**

**Question #1.** At this time, all developments are not available in Github. So software was not written using unified environment, for example Python, but it looks rather fragmented and seems to be not very user friendly?

**Answer.** The necessity to use different programming languages is caused by technical features of components and software packages. The Unity software package runs in C#, the Arduino Nano and Due microcontrollers support writing in C/C ++, while the code for controlling the robot's flight requires Python. Thus, it is not possible to unify the programming language within the framework of a scientific thesis.

Since I worked with a small number of colleagues, there was no need to store the code on Github. However, I plan to submit the code on Github in the near future with the necessary explanations.

**Question #2.** The author of the thesis should double check English, grammar, spelling and punctuation marks. In many places are missing dots and commas. For example, each figure and table caption should be finished by full stop (dot). Each sentence should be finished by full stop even it contains equations.

**Answer.** I revised the text of the thesis and made the necessary amendments.

**Reviewer: Hiroyuki Kajimoto**

**Question #1.** In Chapter 3, I would particularly like to see a discussion of viewpoint positions, i.e., third and first person viewpoints. Currently, it is explained that the camera attached to the end of the arm is the first-person view and the VR image is the third-person view, and only a limited comments from the users are listed.

However, the issue of viewpoint is quite important in teleoperation. In many cases, a robot that moves remotely can only create a first-person view. However, the first-person view is often insufficient for smooth work, and it has been pointed out that the "third person view from slightly behind" is more important. For example, Inami et al.'s study ensures a pseudo third-person viewpoint by using "images from the past in time".

https://www.youtube.com/watch?v=cu9w_UUaClY

In this experiment, a third-person viewpoint was mainly used, and I'm wondering if any inconsistencies occur with respect to the position and orientation of the hands. For example, when you move your hand back and forth, does the drone's hand move left and right in a coordinate rotation? In other words, I would like to know the relationship between the world coordinate system, the user's local coordinate system, and the drone's local coordinate system.

Also, I would like to hear "why" the visual system is the way it is now proposed. Overall, there seems to be little description of the visual system considerations. On the other hand, it is also necessary to discuss whether it is possible or realistic to use such a third person perspective in real scenarios.

**Answer.** The issue of choosing the correct viewing angle for teleoperation is one of the most important. In the thesis, we experimentally selected a convenient view for controlling the robot in a simulated environment (second user study, subchapter 3.9.2). It is a third-person view, slightly behind the robot and slightly higher. This field of view is realized by binding the HMD coordinate system to the robot's center with some offset. For the real experiment (chapter 3.8) in laboratory conditions (room with an area of 5 by 5 square meters), we used a motion capture system, which provides the precise positions of the robot and object. In this case, the operator could change his position in the room and inspect the robot from a comfortable viewing angle. For this purpose, the operator should use the UAV controller to switch to the mode of the holding position in order to disable control of the robot's body movement temporarily. Similarly, the operator could freely choose a viewpoint during the first user study (subchapter 3.9.1).

The third-person view is indeed important. This view allows the operator to see the whole robot when controlling its robotic limbs. The operator must be fully aware of the robot's size that the robot does not collide the robot into potential obstacles. If necessary, this function can be accomplished without the third-person view using additional algorithms and tactile feedback, giving an auxiliary signal that the desired position is unattainable (for example, vibration or a light signal in the VR interface). However, a convenient field of view provides more useful information about the surrounding environment. Using only a first-person view for aerial manipulation is generally insufficient for successful robot control. When the camera is only installed on the robot body, the operator can not grab the target object. If the camera is installed on the gripper, the operator can be disoriented at the manipulator moving and especially at the turning the gripper along the roll direction. Thus we used a camera on the gripper as an additional source of information useful for grasping operation. I added a description of viewpoint positions in chapter 3.2.

Rotation of the manipulator to the side is not provided when using a wearable device. The wearable interface only provides data on the angles at joints. At the same time, the manipulator is planar, and the rotation of the robot and movement in the direction perpendicular to the manipulator is carried out using the VR controller in the other hand (chapter 3.3).

For outdoor conditions, I suppose to apply computer vision methods. Using this technology, we can perform mapping of the surrounding environment in real-time. Thus, we can use a tested approach for free-viewpoint when the operator uses the holding position to perform aerial manipulation operations.

**Question #2.** Three tactile interfaces were proposed for remote control: the first one using a Vive controller, the second one using a transducer on the fingertip + IMU on the instep and arm, and a bending sensor on the hand, and the third one using an IMU sensor on the shoulder and bending sensors on the elbow and wrist. However, there are no photos available for this third one. Also, there does not appear to be any mention of the third one in the user study. Could you explain which one was used for the user study.

**Answer.** Photo of the first interface is presented in Fig. 3-1 (c), photo of the third interface is shown in Fig. 3-3 (b), and photos of IMU-based interface are presented in Fig. 3-3 (c, d) and Fig. 3-12 (a). In subchapter 3.4.3, I wrote that after preliminary tests we decided not to use the third interface with bending sensors in the user study due to significant dependence of data reading quality from the sensor position on the operator's arm. Thus we chose an IMU-based interface to compare with the VR controller in the user study and the smart glove with trackers for flight experiments.

**Question #3.** In the conclusion of chapter 3, it stated that one of the major drawbacks of the VR controller compared to the IMU-based interface is the need for an external tracking system. However, even though it is certainly true for the Vive, there are already many products that do not require an external tracker, such as the Oculus Quest, so this argument should be weakened. Rather, if possible, discuss how being IMU-based is "inherently" advantageous.

**Answer.** Yes, not all VR devices require stationary base stations nowadays. Advantages of the IMU-based interface are a more natural method of manipulator control and higher manipulation speed (faster by 27% in comparison with HTC Vive controller) with better matching of the manipulator position to the target set-point, that was confirmed by ANOVA results of the first user study. The ease of usage of the wearable interface is explained by the fact that the manipulator simply repeats the movements of the operator's hand. The operator intuitively controls the manipulator joints directly, which provides precise positioning and greater awareness. That is, the operator subconsciously associates his hand with the robotic hand.

**Question #4.** With respect to chapter 4, the illustration of the whole paper in chapter 1 (Fig. 1.5) seems to state that chapter 4 also contains design related to human factors, namely GUI and Visualization. However, Chapter 4 does not seem to contain such a topic. If this is the case, I think it is necessary to revise Fig. 1.5 (or to add an experiment on human factors in Chapter 4).

**Answer.** GUI for DroneGear was developed for the heuristic analysis of the robot locomotion. I added the description and screenshot of this GUI in subchapter 4.5.1. This interface was used as an auxiliary tool for comfortable research of locomotion algorithm by controlling each servomotor's positions, representing the actual state of the robot, its joint angles, and sensor data. It makes sense to mention that GUI for the AeroVR robot was developed for the same goal. However, the VR-based teleoperation method can be used to control DroneGear using position-velocity control.

**Reviewer: Gonzalo Ferrer**

**Question #1.** I find particularly valuable the fact that the student had to master many diverse methods to succeed on his thesis, probably supported by colleges dividing the workload.

**Answer.** In the framework AeroVR project, I developed and assembled the flying robot with a 4-DoF manipulator. The lightweight manipulator was designed in CATIA; manufacturing of the manipulator and control electronics was implemented by myself; the control software is written in Arduino IDE using C/C++ and in MATLAB (GUI). I also designed three prototypes of wearable devices and wrote the control code for them. In the framework of the LocoGear project, I researched the possible movement of landing gear using heuristic analysis. Based on this analysis results, I developed a novel locomotion algorithm by trajectory analysis of the robot CoM, calculation of the dynamic loads using the Lagrangian formulation, and kinetostatic methods. The analysis of experimental data for both projects was also carried out by myself.

Daria Trinitatova and Ruslan Agishev contributed to the research part dedicated to remote control and tests of aerial manipulation. Daria created a VR scene in Unity for teleoperation, and we together wrote the code in Unity for communication between a real robot, wearable interface, and its digital twin. Ruslan was responsible for the flight part of the experiments and wrote the code for exchanging information between the UAV's autopilot and the robot's onboard computer for further sending the data to Unity. I discussed with Yuri Sarkisov the developed locomotion strategy and the dynamic modeling of the landing platform for the multirotors. Also, the first prototype of the landing platform for the multirotors was designed and assembled by Yuri.