Feed-forward Image-Based Visual Servoing을 이용한 UAV의 함상 자동 착륙 연구

Autonomous Ship Deck Landing of UAVs Using Feed-forward Image-Based Visual Servoing

초록

무인기가 모든 임무를 자동으로 수행하기 위해 자동 이착륙은 필수적이다. 하지만 GPS(Global Positioning System)등의 정확성이 낮은 센서는 배가 파도에 의해 요동치면서 빠르게 이동하는 상황에서 함상 자동 이착륙에 사용하기에는 제한적이다. 본 연구에서는 기존의 정지된 목표지점에 착륙하기 위해 사용되던 image-based visual servoing 기법에 이동하는 착륙지점인 배의 속도를 보상해주는 방법을 적용하였다. 이를 위해 배의 GPS로부터 얻은 속도정보와 무인기의 카메라로부터 얻은 영상정보 및 배의 다이나믹 모델을 융합하여 배의 속도를 추정하였다. 제안된 알고리즘은 배가 빠르게 이동하면서 파도에 의해 급격히 흔들리는 상황을 모사한 시뮬레이션과 다양한 실제 비행실험을 통해 검증되었다.

Key Words : Autonomous landing, Image-based visual servoing, Sensor fusion, Unmanned aerial vehicle

Introduction

Unmanned aerial vehicles (UAVs) have been used widely for surveillance, reconnaissance, searching, rescue, and inspection of wind turbines, bridges, in both military and industrial field. For all missions listed above, takeoff and landing of the UAV have to be performed before and after missions. Especially, if the UAV is operated at marine environments which are far from the land, the UAV must be able to land on a small and narrow area of the moving ship, often oscillated by sea waves. However, autonomous landing in this condition using traditional sensing system such as global positioning system (GPS) is not sufficient due to its large position error. In order to overcome the inaccuracy of GPS, vision sensor can be added to the sensing system, and in this research, visual servoing is exploited for autonomous landing of a UAV on a moving ship deck.

Visual servoing is the control method for an agent (especially robot manipulator[1,2]) to move to a target position using a vision sensor. It can be divided into image-based visual servoing (IBVS) and position-based visual servoing (PBVS)[3]. In PBVS, states are defined as the pose of the target in the Cartesian coordinate, and the target pose is estimated with respect to the camera frame. The error is expressed as the relative pose between the target and camera. PBVS allows the camera to move to the target in the optimal trajectory. But poor state estimation leads to instability of the pose of the camera such as perturbations in the trajectory and inaccuracy after convergence[20]. Visual servoing algorithm has been applied to tracking and landing of a UAV as well as robot manipulator. There exist many studies for autonomous landing of a UAV using PBVS. Yang et al. exploited PBVS to takeoff and landing of a UAV, and square root unscented Kalman filter is used to estimate the pose of the UAV[4]. Jose et. al. carried out the study of landing on a moving platform using PBVS[5]. In this study, the platform is moving with the maximum speed of 10km/h. Falanga et al. developed a fully autonomous quadrotor system which can land on a moving target using PBVS[6]. In the study of Robson et. al., autonomous landing using PBVS on a platform oscillating with the heave motion is presented[7], and simulations with the Virtual Robot Experimentation Platform (V-REP)[8] are carried out to verify the algorithm.

On the other hand, in IBVS, states are the position of the features in the image plane, and the error is defined as the pixel position error between the desired feature position and current feature position. From the pixel error in the image plane, the desired velocity command to move the camera to target pose is calculated with image Jacobian which explains the relationship between the velocity of the camera in 3D space and feature velocity in 2D image plane. Compared with PBVS, IBVS is robust to pixel measurement error(20). Research for vision-based autonomous landing using IBVS also has been carried out actively. Hamel et. al. applied IBVS to under-actuated system for the first time with robust backstepping technique. They considered the full dynamics of the camera motion fixed to the rigid body(9). As an extension of this work, Guenard et. al. carried out the hovering experiment of a quadrotor UAV(10). Lee et. al. applied the virtual image plane to IBVS for tracking and landing of the UAV to compensate the effect of the attitude of the UAV and designed an adaptive sliding mode controller(11,12). Serra et.al. proposed control law for landing on a platform with the heave motion, and conducted simulations and indoor experiments(13). In Quang et. al.(14), a controller for ship landing of helicopter using combination of IBVS and translational rate command is introduced and simulations are carried out. In Thomas et. al. (15), to land on a moving ship deck, velocity of the ship is estimated using the response amplitude operator and the auto regressive with moving average model. The motion of the ship is compensated in the IBVS controller and its performance is verified with simulations. Borshchova et. al. conducted simulations and experiments of autonomous landing on a ship deck(16,17). They exploited the color detection method as features for IBVS to reduce the computational load. Simulations and experiments are conducted for a moving target with V-REP simulations. Wynn et. al. proposed feed-forward IBVS (FF-IBVS) to compensate the velocity of the moving ship(18). Velocity of the ship is estimated by extended Kalman filter (EKF) which fuses visual and GPS measurement and estimated velocity is added as feed-forward term of IBVS. They also proposed the whole process for autonomous landing on a moving target starting from the approach phase. In the experiment, velocity of the target is set around 1m/s and it has heave motion, and precision landing performance is verified.

As mentioned above, there are a lot of studies to land the UAV on a ground vehicle or a ship. However, there is a lack of study when the landing target is moving fast as well as oscillating. In terms of the autonomous landing system, a study on the entire landing procedure starting from the approach phase to the touchdown phase is also not sufficient. Therefore, this paper proposes the autonomous landing system of the UAV based on FF-IBVS to land on a small ship deck which is moving fast and oscillating by sea waves. In order to make the landing system robust and stable, the adaptive IBVS gain is applied and feature shape is compensated. Besides, to improve the landing performance, reliable velocity estimation of the ship is newly introduced and the entire landing procedure is made fully autonomous.

The main contribution of this study is as follows. First, we enhanced the autonomous landing performance based on research of FF-IBVS(18) using several innovative techniques: adaptive IBVS gain, compensation of features for IBVS, and improved estimation using the Kalman filter and sensor fusion. The adaptive IBVS gain is used to keep the features in the field of view (FOV) by slowing down the altitude rate, and the features are compensated to remove the unnecessary IBVS command occurred by the changing attitude of the ship. In the Kalman filter, the ship motion is modeled as high order linear motion and a pseudo measurement technique is used to improve the estimation performance(22). Second, a landing system for robust and safe autonomous landing is designed. To detect the features and land to the target from long distances, the size and placement of AR tags (marker for the landing) are carefully determined. The landing procedure starting from approach phase to touchdown phase is also designed with the state machine structure. It allows that if the marker is missed, the UAV to hold the position near the target or to move the position to find the marker again. Lastly, realistic simulations and flight experiments in harsh conditions are conducted. In the simulations, the environment is set to very harsh condition where the ship is moving at a fast speed of 5m/s while oscillating at sea state 4. To the best of our knowledge, speed of 5m/s is the fastest environment setup for vision-based autonomous landing of a quadrotor UAV. Sea State 4 refers to the height of sea waves between 1.25 and 2.5 meters. Under such severe circumstances, it is very difficult for UAVs to land on a moving ship deck.

This rest of this paper is organized as follows. In Section image-based visual servoing, IBVS is introduced briefly and additional process for under-actuated system such as quadrotor UAV are explained. Next, feed-forward IBVS which is for compensate the velocity of a moving target for precision landing is presented in Section feed-forward IBVS. In landing system, whole autonomous landing system marker for IBVS setup and landing procedure is proposed. The performance of the proposed controller and landing system is verified with simulations and experiments in the next sections.

Image-Based Visual Servoing

In this section, in order to help readers understand, image-based visual servoing (IBVS) is briefly reviewed. As mentioned in previous section, the aim of the IBVS is to reduce the pixel error between the desired and measured feature position in the image plane. The error $\boldsymbol{e}$ defined in the IBVS is:

$$\boldsymbol{e} = \boldsymbol{s}_d - \boldsymbol{s}$$

where $\boldsymbol{s}_d$ is the desired feature position and $\boldsymbol{s}$ is the measured feature position. The relationship between the camera and feature velocity is given by:

$$\dot{\boldsymbol{s}} = \boldsymbol{L}_s \boldsymbol{v}_c$$

where

$$\boldsymbol{L}_s = \begin{bmatrix} -\boldsymbol{f}/\boldsymbol{Z} & \mathbf{0} & \boldsymbol{x}'/\boldsymbol{Z} & (\boldsymbol{x}'\boldsymbol{y}')/\boldsymbol{f} & -(\boldsymbol{f}+\boldsymbol{x}'^{2})/\boldsymbol{Z} & \boldsymbol{y}' \\ \mathbf{0} & -\boldsymbol{f}/\boldsymbol{Z} & \boldsymbol{y}'/\boldsymbol{Z} & (\boldsymbol{f}+\boldsymbol{y}'^{2})/\boldsymbol{f} & -\boldsymbol{x}'\boldsymbol{y}'/\boldsymbol{Z} & -\boldsymbol{x}' \end{bmatrix}$$

is the image Jacobian and camera velocity $\boldsymbol{v}_c \in \mathbb{R}^6$ which consists of linear three motions and angular three motion ($\boldsymbol{v}_c = [v_{c,x} \quad v_{c,y} \quad v_{c,z} \quad \omega_{c,x} \quad \omega_{c,y} \quad \omega_{c,z}]^\top$). The image Jacobian matrix can be obtained from pinhole camera model described in Fig. 1. Arbitrary point $\boldsymbol{P} = [x \quad y \quad z]^\top$ which is expressed in camera frame ($\boldsymbol{O}_{XYZ}$) is projected to image plane as $s = [x' \quad y']^\top$, then:

$$x' = fx/z, \tag{1}$$

$$y' = fy/z, \tag{2}$$

where f is camera focal length. By taking time derivative of Eq. (1) and Eq. (2):

$$\dot{x'} = f(\dot{x} - x'\dot{z})/z, \tag{3}$$

$$\dot{y'} = f(\dot{y} - y'\dot{z})/z. \tag{4}$$

When the camera is moving in 3-D space, velocity of the arbitrary point in the camera frame is:

$$\dot{\boldsymbol{P}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = -\boldsymbol{v}_{c,lin} - \boldsymbol{v}_{c,ang} \times \boldsymbol{P} = \begin{bmatrix} -v_{c,x} - \omega_{c,y} z + \omega_{c,z} y \\ -v_{c,y} - \omega_{c,z} x + \omega_{c,z} z \\ -v_{c,z} - \omega_{c,x} y + \omega_{c,z} x \end{bmatrix}, \tag{5}$$

where $\boldsymbol{v}_{c,lin} = [v_{c,x} \quad v_{c,y} \quad v_{c,z}]^\top$ and $\boldsymbol{v}_{c,ang} = [\omega_{c,x} \quad \omega_{c,y} \quad \omega_{c,z}]^\top$ are linear and angular velocity of the camera, respectively. From Eq. (3) ~ Eq. (5), relationship between the velocity of the feature in the image plane and the arbitrary point $\boldsymbol{P}$ can be obtained.

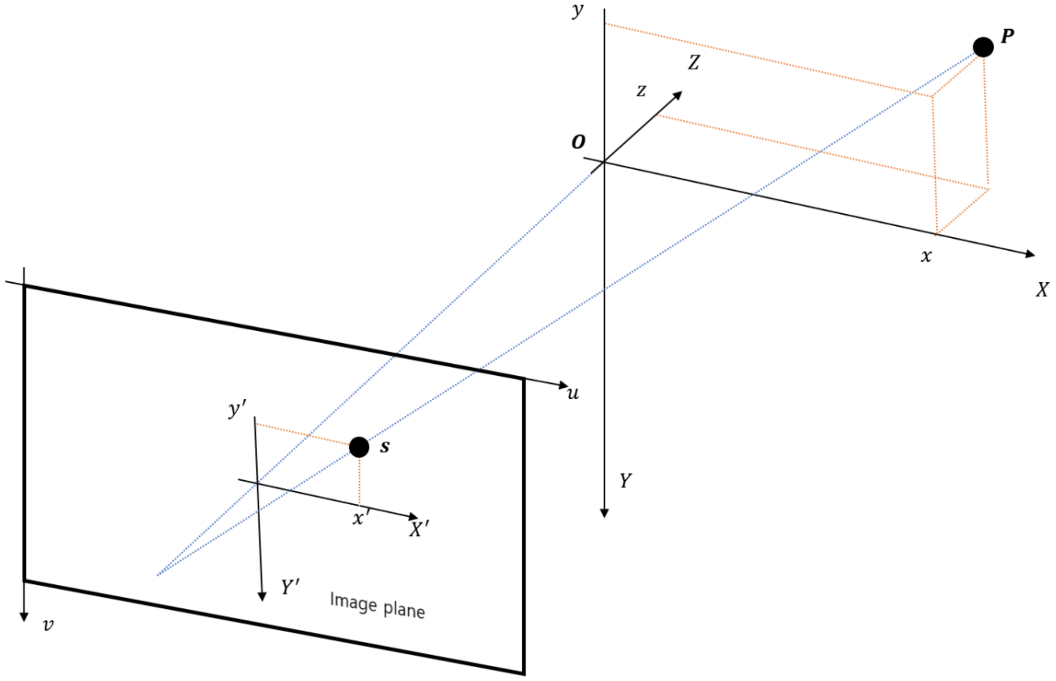


Fig. 1. Pinhole camera model.

If the desired feature position $\boldsymbol{s}_d$ is static ($\dot{\boldsymbol{e}} = \dot{\boldsymbol{s}}$) and ensuring feature error is exponentially decrease ($\dot{\boldsymbol{e}} = -\lambda\boldsymbol{e}$), then relationship between camera velocity and feature error can be obtained as:

$$\boldsymbol{v}_c = -\lambda \boldsymbol{L}_s^+ \boldsymbol{e}$$

where λ is positive gain and $\boldsymbol{L}_s^+$ is pseudo inverse matrix of $\boldsymbol{L}_s$. In fact, knowing the exact value of the $\boldsymbol{L}_s$ is impossible, so IBVS controller is designed as:

$$\boldsymbol{v}_d = -\lambda \hat{\boldsymbol{L}}_s^+ \boldsymbol{e}$$

where $\hat{\boldsymbol{L}}_s^+$ is estimation of $\boldsymbol{L}_s^+$ and $\boldsymbol{v}_d$ is desired camera velocity. If assume that center of the UAV coincides with camera center, $\boldsymbol{v}_d$ can be considered as desired UAV velocity.

IBVS for under-actuated system

From the IBVS controller, six-DOF(Degree Of Freedom) desired velocity ($\boldsymbol{v}_\mathrm{d} \in \mathbb{R}^6 = [v_{d,x} \quad v_{d,y} \quad v_{d,z} \quad \omega_{d,x} \quad \omega_{d,y} \quad \omega_{d,z}]$) is calculated. But at the under-actuated system such as the quadrotor UAV, roll and pitch rate are coupled with linear velocity for y and x direction, respectively. In other words, the UAV cannot make roll and pitch rate motion independent of y and x velocity. Furthermore, the UAV will make opposite velocity command to the target position depending on the situations. For example, as described in the Fig. 2(a), the target position is in left of the UAV, but the target is projected on the right half plane of the image plane because of the angle of the UAV, so that the UAV will move to right direction which is the opposite to the target position. To overcome these problems, features are transformed to a virtual coordinate frame[11]. The virtual coordinate frame is defined that the center of the origin coincides to the camera coordinate frame and its z axis is parallel to the z axis of the inertial frame. The virtual image plane is also defined as the plane which the features projected to the virtual coordinate frame. It means that virtual image plane is always parallel to the ground and if the landing pad is parallel to the ground, roll and pitch rate are always zero so that under-actuated system can be decoupled.

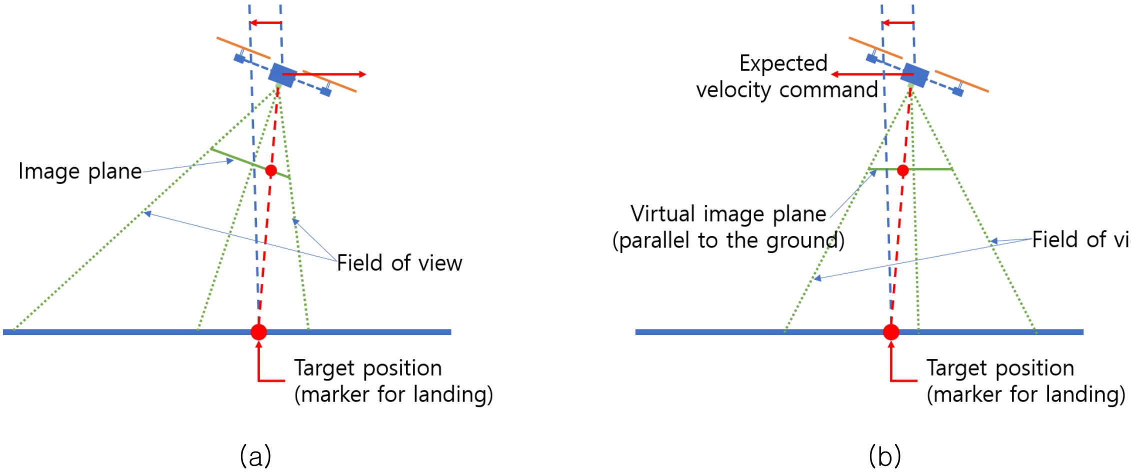


Fig. 2. Projection of the target position on the image plane and virtual image plane: (a) on the image plane; and (b) on the virtual image plane.

Virtual image plane transform can be conducted using the roll and pitch angle of the UAV. The ith point in the camera frame $\boldsymbol{P}_i$ is expressed in the imaginary camera frame as $\boldsymbol{P}_i^r$, and its corresponding coordinate in the image plane and virtual image plane are $\boldsymbol{p}_i$ and $\boldsymbol{p}_i^r$. The relationship between $\boldsymbol{P}_i$ and $\boldsymbol{P}_i^r$ can be expressed as:

$$\boldsymbol{P}_i^r = R(1,\phi)R(2,\theta)\boldsymbol{P}_i \tag{6}$$

where

$$R(1,\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & sim(\phi) & \cos(\phi) \end{bmatrix}, R(2,\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}.$$

Using Eq. (6), transform equation from image frame to virtual image frame can be expressed as:

$$\boldsymbol{p}_i^r = \frac{f}{z_i^r}\begin{bmatrix} x_i^r \\ y_i^r \end{bmatrix} = f\begin{bmatrix} \dfrac{u_i\, cos(\theta) + f sin(\theta)}{-u_i\, cos(\phi)\, sin(\theta) + v_i\, sin(\phi) + f cos(\phi) cos(\theta)} \\ \dfrac{u_i\, sin(\phi)\, sin(\theta) + v_i\, sin(\phi) + f cos(\phi) cos(\theta)}{-u_i\, cos(\phi)\, sin(\theta) + v_i\, sin(\phi) + f cos(\phi) cos(\theta)} \end{bmatrix}$$

where x_i^r, y_i^r, and z_i^r are the position of the marker in X', Y', and Z' axis, respectively.

However, even if a virtual image plane is applied to IBVS, the limited camera field of view (FOV) can be a problem. If the UAV decrease the altitude by IBVS control command when its horizontal position is far from the center of landing target, the features will out of the image plane because of limited FOV as described in the Fig. 3. In order to operate the IBVS keeping the features inside of the image plane is important. To leave the features into the image plane, Lee et. al. suggested adaptive IBVS gain which use inverse tangent function(11). In this research, IBVS gain for altitude rate is adjusted by the feature error in the image plane to have small control input if the UAV has a large horizontal distance error. Adaptive gain is designed with sigmoid function as:

$$\mathrm{ad_z} = 1 - \frac{1}{1 + e^{-kc}}$$

where c is center feature coordinate in the image plane, and k is gain of the sigmoid function. The IBVS control command with adaptive gain is given as:

$$\boldsymbol{v}_d = -\lambda \begin{bmatrix} 1 \\ 1 \\ ad_z \\ 1 \end{bmatrix} \hat{\boldsymbol{L}}_s^{+}\boldsymbol{e}.$$

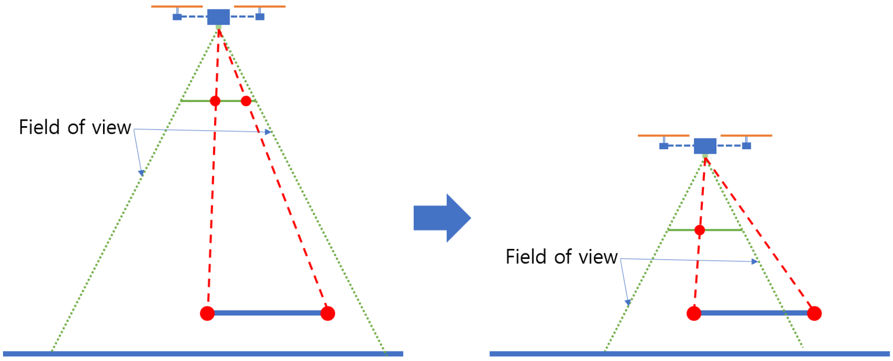


Fig. 3. Out of FOV of the landing features due to the low altitude.

Square compensation

After transformation of the features to the virtual image plane, decoupling between speed for x and y axis, and roll and pitch rate is conducted. However, if the landing target is oscillating, in other words, plane made up of features is not parallel to the image plane, and coupling cannot be removed. As described in the Fig. 4, for autonomous landing using IBVS, there are four markers and distance between camera center and right two markers is closer than left two markers. Then, distance between right part of the features is longer than another and desired velocity to left direction command will be made. To remove the effect of orientation of the landing pad, four features are fitted to square using least square method. If the four features form a square, it satisfies that the plane made up of four features treated as parallel to the image plane, and IBVS control command will not make unnecessary command depending on the attitude of the landing target like Fig. 4.

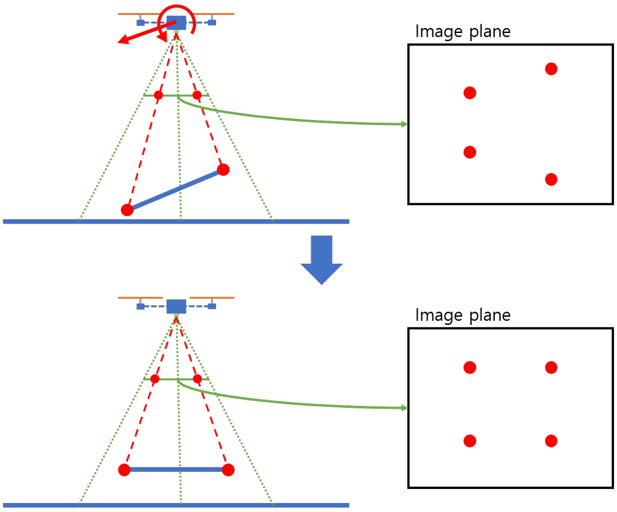


Fig. 4. Square fitting of the features.

After transformation to virtual image plane and square fitting of the features, the IBVS controller for underactuated system makes the four-DOF ($\boldsymbol{v}_{d,4DOF} = [v_x \quad v_y \quad v_z \quad \omega_z]^{\top}$) velocity command. Then, the image Jacobian for quadrotor-UAV is expressed as:

$$\boldsymbol{L}_{s,4DOF} = \begin{bmatrix} -f/Z & 0 & x'/Z & y' \\ 0 & -f/Z & y'/Z & -x' \end{bmatrix},$$

and generated velocity command using $\boldsymbol{L}_{s,4DOF}$ is:

$$\boldsymbol{v}_{d,4DOF} = -\lambda\, \hat{\boldsymbol{L}}^{+}_{s,4DOF}\boldsymbol{e}$$

where $\hat{\boldsymbol{L}}^{+}_{s,4DOF}$ is pseudo inverse of the estimation of the $\boldsymbol{L}_{s,4DOF}$.

Feed-forward IBVS

Basic IBVS assumes that the target is stationary[3,20]. In order to apply IBVS to moving target, target velocity should be compensated. In this work, only horizontal velocity of the ship is compensated. The desired velocity command with target velocity compensation as feed-forward is expressed as:

$$\boldsymbol{v}_{d,ff} = \boldsymbol{v}_{d,4DOF} + \hat{\boldsymbol{v}}_{target}$$

where $\hat{\boldsymbol{v}}_{target} = [\hat{v}_{target,x} \quad \hat{v}_{target,y} \quad 0 \quad 0]^{\mathsf{T}}$ is estimated horizontal velocity of the ship.

For the measurement, a GPS/INS sensor and the camera are used. The GPS/INS sensor is attached on the ship deck and measures the velocity of the ship. The coordinate of the features in the image plane measured from camera calculates the relative pose of the landing pad with respect to the UAV and the relative pose is used for pose measurement. Using relative pose and velocity of the target, target velocity is estimated using KF (Kalman Filter).

Since the ship might oscillate due to wave in marine environment, the designed filter should be able to consider the effect of the wave. There have been various approaches to estimate pose of a ship in such condition. For instance, in [21], the Tristan et. al. applied a frequency domain-based method to estimate the motion of a ship. Nonetheless, aforementioned study requires a dynamic model of the ship, which is normally hard to achieve. In this paper, we simply utilize the KF with constant crackle (5th time derivative of position) model and pseudo-measurement as shown in [22], since we do not have any knowledge about ship's physical characteristic. Assuming the ship is moving with constant velocity while oscillating due to the wave, this model would be enough to represent minor rotation of the ship.

Furthermore, to estimate the ship states, we apply track-to-track fusion algorithm consists of two distinct KFs which is described in Fig. 5. Each filter updates the ship states via GPS/INS and camera, respectively. We assume that the position and velocity data are available for the GPS/INS and the camera provides only relative position.

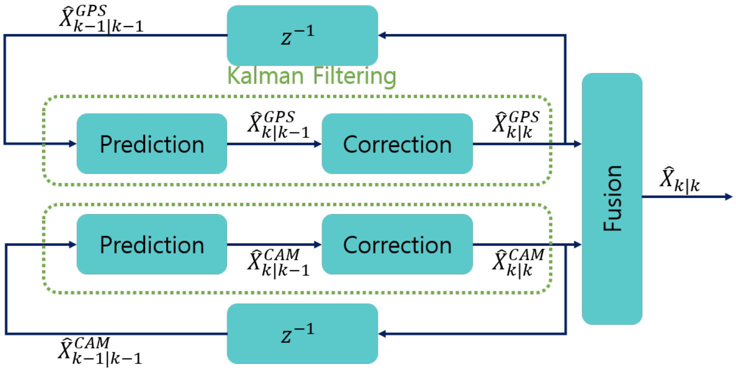


Fig. 5. Track-to-track fusion structure.

Let us define the true states of the ship as follow:

$$\boldsymbol{X} = \left[x_{ship} \quad y_{ship} \quad \dot{x}_{ship} \quad \dot{y}_{ship} \quad \cdots \quad \overset{(5)}{x} \quad \overset{(5)}{y}\right]^{\mathsf{T}}$$

where x and y are the position of the ship with respect to the vehicle-1 frame.

The dynamic matrix of the constant crackle model ($\boldsymbol{F}$) can be defined as:

$$\boldsymbol{F}=\begin{bmatrix}\boldsymbol{I}_2 & T\boldsymbol{I_2} & \frac{T^2}{2}\boldsymbol{I}_2 & \frac{T^3}{6}\boldsymbol{I}_2 & \frac{T^4}{24}\boldsymbol{I}_2 & \frac{T^5}{120}\boldsymbol{I}_2\\ \boldsymbol{0} & \boldsymbol{I}_2 & T\boldsymbol{I_2} & \frac{T^2}{2}\boldsymbol{I}_2 & \frac{T^3}{6}\boldsymbol{I}_2 & \frac{T^4}{24}\boldsymbol{I}_2\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_2 & T\boldsymbol{I_2} & \frac{T^2}{2}\boldsymbol{I}_2 & \frac{T^3}{6}\boldsymbol{I}_2\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_2 & T\boldsymbol{I_2} & \frac{T^2}{2}\boldsymbol{I}_2\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_2 & T\boldsymbol{I_2}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_2\end{bmatrix}, \tag{7}$$

for the both filters. In Eq. (7), T is the sampling time and $\boldsymbol{I}_2$ is the 2 × 2 identity matrix.

The estimated states of the ship at time step k for the filter with GPS/INS is represented as $\widehat{\boldsymbol{X}}_{k|k}^{GPS}$. Then, the prediction step of the filter can be expressed as:

$$\widehat{\boldsymbol{X}}_{k|k+1}^{GPS}=\boldsymbol{F}\widehat{\boldsymbol{X}}_{k|k}^{GPS},$$

$$\widehat{\boldsymbol{P}}_{k|k+1}^{GPS}=\boldsymbol{F}\widehat{\boldsymbol{P}}_{k|k}^{GPS}\boldsymbol{F}^{T}+\boldsymbol{Q}_k,$$

where $\widehat{\boldsymbol{X}}_{k|k+1}^{GPS}$ is the predicted state, $\widehat{\boldsymbol{P}}_{k|k}^{GPS}$ is the error covariance matrix, $\widehat{\boldsymbol{P}}_{k|k+1}^{GPS}$ is the predicted error covariance matrix, and $\boldsymbol{Q}_k$ is the system noise at time step k. Since the GPS/INS measures both position and velocity, the measurement matrix, $\boldsymbol{H}_{vel}$, is defined as follow:

$$\boldsymbol{H}_{vel}=\begin{bmatrix}\boldsymbol{I}_2 & 0 & 0 & 0 & 0 & 0\\ 0 & \boldsymbol{I}_2 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & \boldsymbol{I}_2\end{bmatrix}.$$

Accordingly, the correction step can be written as:

$$\boldsymbol{S}_{k+1}^{GPS}=\boldsymbol{H}_{vel}\widehat{\boldsymbol{P}}_{k|k+1}^{GPS}\boldsymbol{H}_{vel}^{T}+\boldsymbol{R}_{k}^{GPS},$$

$$\boldsymbol{v}_{k+1}^{GPS}=\boldsymbol{y}_{k+1}^{GPS}-\boldsymbol{H}_{vel}\widehat{\boldsymbol{X}}_{k|k+1}^{GPS},$$

$$\widehat{\boldsymbol{X}}_{k+1|k+1}^{GPS}=\widehat{\boldsymbol{X}}_{k|k+1}^{GPS}+\boldsymbol{K}_{k+1}^{GPS}\boldsymbol{v}_{k+1}^{GPS},$$

$$\boldsymbol{K}_{k+1}^{GPS}=\widehat{\boldsymbol{P}}_{k|k+1}^{GPS}\boldsymbol{H}_{vel}^{T}\boldsymbol{S}_{k+1}^{{GPS}^{-1}},$$

$$\widehat{\boldsymbol{P}}_{k+1|k+1}^{GPS}=(\boldsymbol{I}_{12}-\boldsymbol{K}_{k+1}^{GPS}\boldsymbol{H}_{vel})\widehat{\boldsymbol{P}}_{k|k+1}^{GPS},$$

where $\boldsymbol{S}_{k+1}^{GPS}$ is the innovation, $\boldsymbol{v}_{k+1}^{GPS}$ is the residual, $\boldsymbol{K}_{k+1}^{GPS}$ is the Kalman gain, and $\boldsymbol{y}_{k+1}^{GPS}$ is the measurement used at time step $k+1$ while $\boldsymbol{R}_{k}^{GPS}$ is the measurement noise at k. It is worth noting that $\boldsymbol{y}_{k+1}^{GPS}$ does not have to be measured exactly at time step $k+1$. In this research, we use the measurement achieved in between time step k and $k+1$ for the correction step at $k+1$. Although it might induce minor error because of the time gap, it would be negligible if the filters run fast enough.

Besides, we utilize the pseudo-measurement technique for the crackle measurement. Thus, the measurement at each time step k becomes

$$\boldsymbol{y}_{k}^{GPS}=[x^{GPS}\quad y^{GPS}\quad \dot{x}^{GPS}\quad \dot{y}^{GPS}\quad 0\quad 0]^{\mathsf{T}}$$

where x^{GPS} and y^{GPS} are measured ship location in x and y direction in the vehicle-1 frame, respectively and $\dot{x}^{GPS}$ and $\dot{y}^{GPS}$ are those of velocities. The pseudo-measurements are set to be 0 to represent the linear movement of the ship while considering the disturbances induced from the wave.

For the sake of simplicity, we describe the detailed equations for the GPS/INS case only. Nevertheless, the counterpart for the camera has the same form except the measurement matrix, i.e., the KF with the camera can be represented by substitute $\boldsymbol{H}_{pos}$ for $\boldsymbol{H}_{vel}$, where $\boldsymbol{H}_{pos}$ is

$$\boldsymbol{H}_{pos}=\begin{bmatrix}\boldsymbol{I}_2 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & \boldsymbol{I}_2\end{bmatrix},$$

and the dimension of corresponding matrices.

After the update of both filters are accomplished, their states are fused by the state-vector fusion method. Each estimation of KFs is weighted based on the magnitude of the corresponding error covariance matrix. Accordingly, the more accurate the estimation from one KF is, the more portion it will take in the fused states. In contrast, if one filter becomes defective, the error covariance increases and it would be automatically excluded from the fused states. The weighting factor can be written as:

$$\boldsymbol{P}_{Tk} = \left[\widehat{\boldsymbol{P}}_{k|k}^{GPS^{-1}} + \widehat{\boldsymbol{P}}_{k|k}^{CAM^{-1}}\right]^{-1},$$

where $\widehat{\boldsymbol{P}}_{k|k}^{CAM}$ is the error covariance matrix for the camera part at time step k. From $\boldsymbol{P}_{Tk}$, the fused state ($\widehat{\boldsymbol{X}}_k^T$) is represented as follow:

$$\widehat{\boldsymbol{X}}_k^T = \boldsymbol{P}_{Tk}\left[\widehat{\boldsymbol{P}}_{k|k}^{GPS^{-1}}\widehat{\boldsymbol{X}}_{k|k}^{GPS} + \widehat{\boldsymbol{P}}_{k|k}^{CAM^{-1}}\widehat{\boldsymbol{X}}_{k|k}^{CAM}\right].$$

From $\widehat{\boldsymbol{X}}_k^T$, we can extract the estimated ship velocity for the feedforward input.

Landing System

This section describes the autonomous ship deck landing system which exploits feed-forward IBVS algorithm. The entire autonomous ship deck landing system is illustrated in the Fig. 6. The ship moves with six-DOF motion (roll, pitch, yaw, surge, sway and heave) caused by sea wave and is also going forward with fast speed. The landing target is placed at the stern of the ship, and on the landing target, a GPS/INS sensor is attached. Position of the ship which is for approaching phase and velocity of a ship which is used as feed-forward data of the ship is measured by the GPS/INS sensor, and the data is sent to the UAV. The markers are placed on the landing target and are used for the features of IBVS. The mission of the UAV is to land to ship deck using only GPS/INS sensor data and vision data from camera under the UAV.

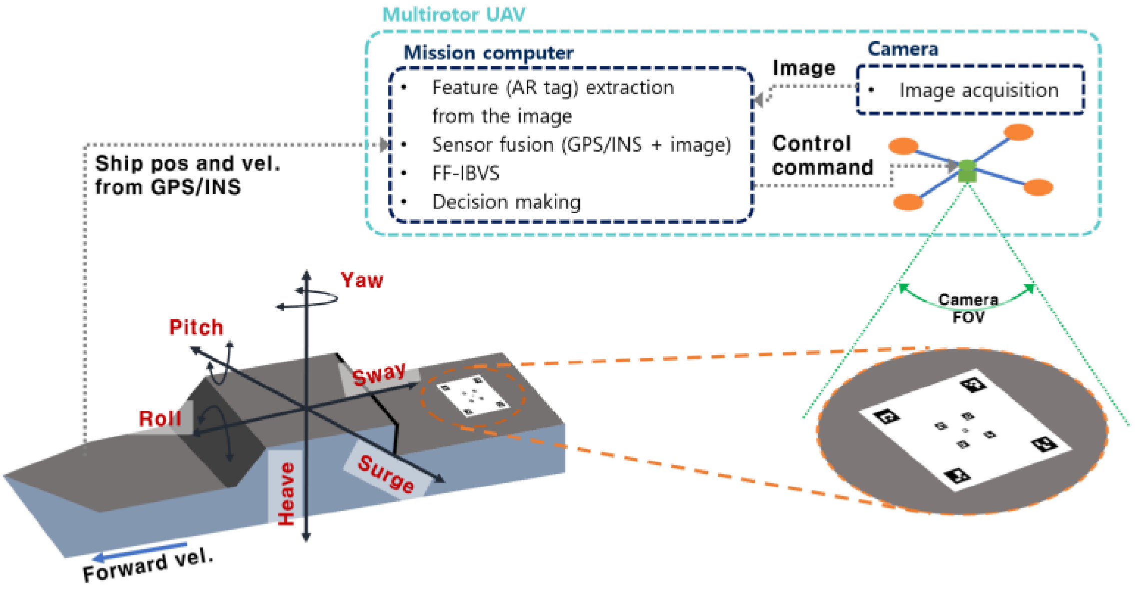


Fig. 6. Overview of the autonomous landing system.

The markers used in this research is described in the Fig. 7. The IBVS level is consist of three IBVS level and different size of AR tags are used for each IBVS level. For the AR tags, ArUco marker(19) is exploited. Source code of ArUco is open in OpenCV, so it is easy to use. The center point of AR tags at IBVS level 1 and 2 forms a square with a length of 2.2m and 0.68m respectively and marker for IBVS level 3 is located in center of the landing target. After recognition of AR tags, the center points of each AR tag for IBVS 1 and 2 and every corner points of the AR tag for IBVS3 are used as feature points of IBVS.

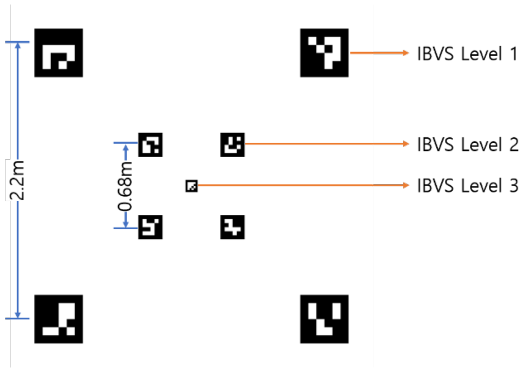


Fig. 7. Markers used for vision based autonomous landing.

Fig. 8 shows all landing procedure. Decision making algorithm for autonomous landing has state a machine structure. There are seven states which are Rendezvous, IBVS (level 1, 2 and 3), hold, rising and landing. At the Rendezvous state, the UAV is guided to around the ship using the GPS from a distance. The UAV approaches to Rendezvous point where is 15m above the center of the landing target. If the UAV is around the Rendezvous point, and AR tags for IBVS level 1 are detected, then, Rendezvous flag becomes true, and FF-IBVS is conducted at IBVS state. During the IBVS state, if the UAV misses the markers, landing state is changed to hold state. At the hold state, the UAV holds its altitude and feed-forward velocity which is the velocity of the ship estimated by GPS enters to the horizontal velocity reference. If the markers are not detected more than 3 seconds, the state enters to the rising state. The rising state is similar with hold state but the UAV rises its altitude with constant speed of 1m/s. while in the hold or rising state, if the UAV detect the markers again, state is changed to IBVS again. If the UAV cannot detect the markers continuously for more than 5 seconds, all parameters are initialized and state is changed to Rendezvous again. At the landing state which is the final state of the landing procedure, the UAV decreases its altitude with constant speed while maintaining feed-forward velocity for horizontal direction.

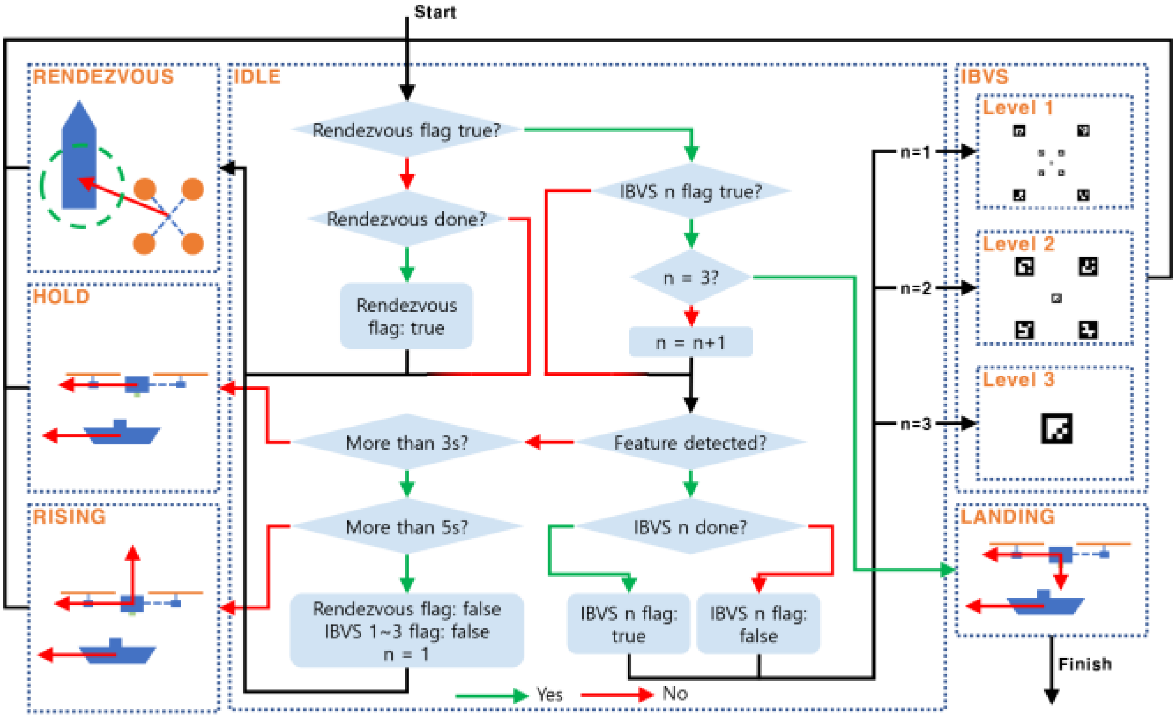


Fig. 8. State machine structure for autonomous landing.

At the Rendezvous state, the UAV should approach to the Rendezvous point while avoiding structures of the ship. For this reason, the Rendezvous point is set to 15m above the stern where is obstacle free. The Rendezvous zone is defined as a cylinder shape with a height of 2m and a radius of 5m, with the center of the Rendezvous zone being the Rendezvous point. If the UAV stays in the Rendezvous zone more than 3 seconds, Rendezvous flag changes to true and goes to IBVS state.

There are three state in the IBVS state which are level 1, 2 and 3. For each IBVS step, the completion of the IBVS is determined by the size of the square formed by the feature points. If the pixel error of the length of one side of the square remains less than the threshold for more than 3 seconds, IBVS $\mathrm{n}(\mathrm{n}=1,2,\mathrm{and}\ 3)$ flag becomes true and landing state goes next state. Finally, if IBVS 3 flag is true, landing state goes to landing state, and all landing procedure is finished.

Simulations

In order to verify the autonomous landing algorithm, simulations are conducted. For the simulator, PX4 simulator was exploited in Gazebo environment. Fig. 9 shows the visualization of the simulation on Gazebo environment using PX4 simulator. Bottom and top of the left side of each figure show the image acquired from camera and extracted feature from the image, respectively. In the feature view, rectangle and circle shape indicate the coordinates of the desired features and measured features, respectively. The horizontal FOV and resolution of the camera is set to 1.79 rad and 2048 × 1536 pixel, respectively. The simulation is carried out at the situation that the ship is going forward with speed of 10 knot (≈5.14 m/s) at the sea state 4 environment. The ship motion is simulated as a superposition of three sinusoidal functions. The ship motion according to the sea state 4 is shown in the

Table 1 and corresponding time history of the motion of the ship is shown in Fig. 10. Position error of the GPS of the platform is modeled as Gauss-Markov processes(23):

$$\nu_{k+1} = e^{-k_{GPS}T_s}\nu_k + \eta_k,$$

where ν is error being simulated, η is Gaussian white noise, $1/k_{GPS}$ is the time constant of the process, and T_s is the sampling time. Velocity error of the GPS of the platform is set to have standard deviation of 0.05 m/s(24).

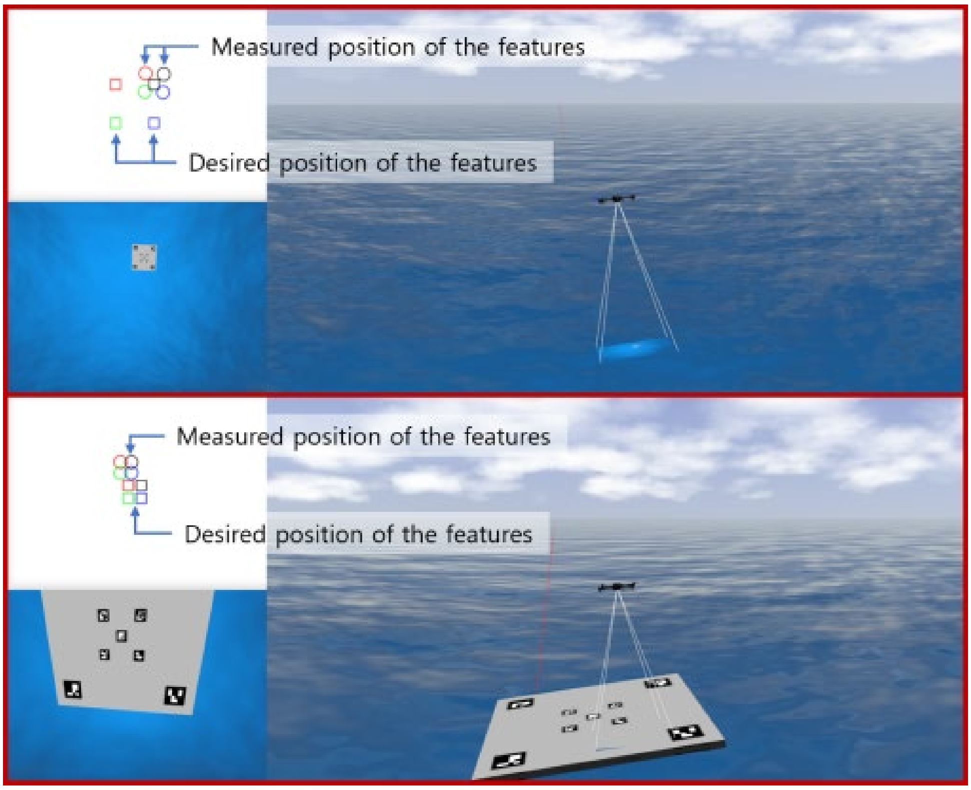
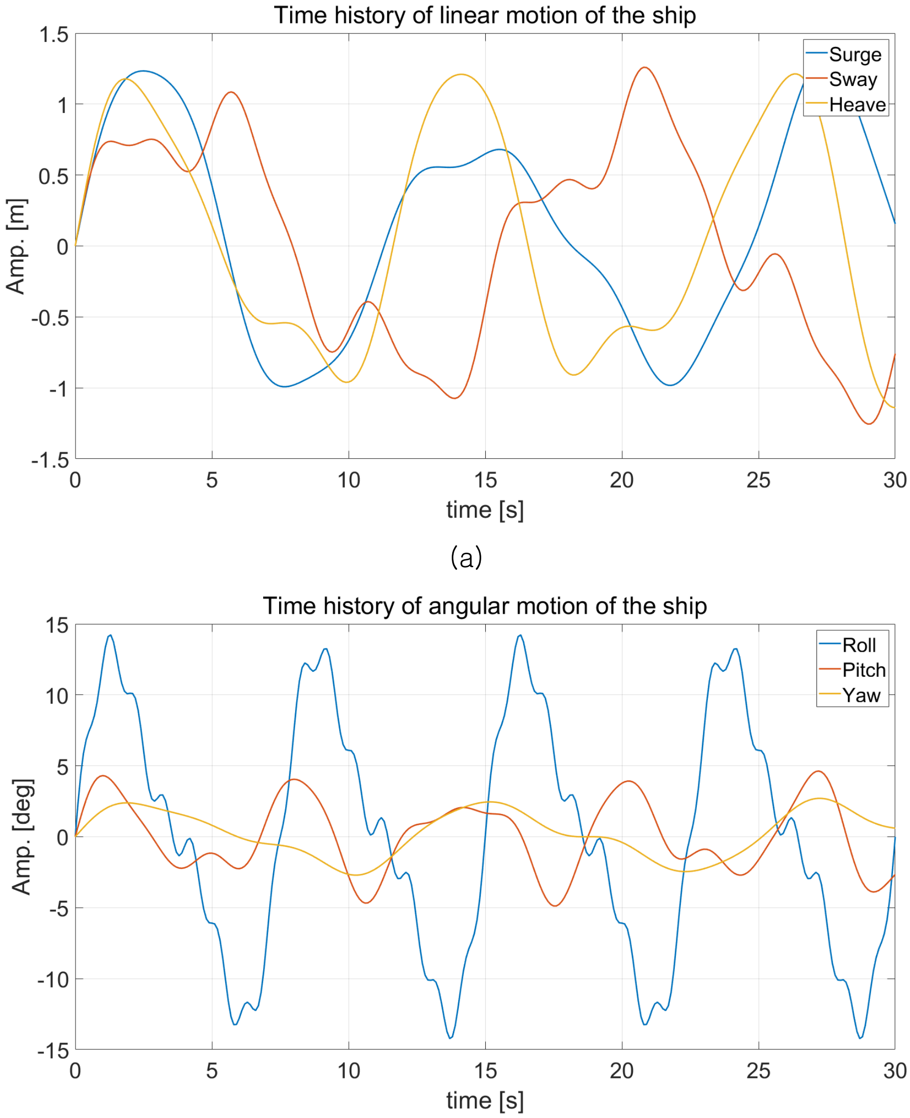

Measured position of the features
Desired position of the features
Measured position of the features
Desired position of the features

Fig. 9. PX4 simulation in Gazebo environment.

Table 1. Simulated ship motion.

Motion	1st order		2nd order		3rd order	
	Amp. [m or deg]	Period [s]	Amp. [m or deg]	Period [s]	Amp. [m or deg]	Period [s]
Surge	1	12	0.36	8.75	0.096	3.75
Sway	0.9	16.3	0.32	5	0.112	2.5
Heave	1	11.3	0.33	6.25	0.12	3.75
Roll	11	7.5	3.97	3.75	1.28	1
Pitch	3.1	6.3	1.54	3.75	0.5	2.28
Yaw	2.14	12.5	0.69	6.25	0.24	3.75



(a)

(b)

Fig. 10. Time history of the motion of the ship: (a) linear motion (b) angular motion.

In the simulations, whole of the landing procedure was conducted to verify the performance of the autonomous landing algorithm including Rendezvous guidance, FF-IBVS and landing procedure with state machine algorithm. The simulations are conducted 10 times. Fig. 11 and Fig. 12 show the results for one out of 10 trials, showing the altitude of the ship and the UAV, and the horizontal position error between the ship and the UAV, respectively. The initial error of altitude and horizontal position error are around 16.2m and 67.9m respectively. At the Rendezvous state, the UAV approaches to 15m above the ship using GPS/INS sensor, and horizontal position error is around 8m. At the IBVS 1 to 3 state, the UAV tracks the ship while descending its altitude according to the velocity command generated by the IBVS. For the final state, the UAV touches down to the ship using just velocity of the ship estimated by the GPS/INS sensor. Fig. 13 shows the touchdown error for all simulations. Through 10 simulations, an average touchdown error of 0.52m, standard deviation of 0.15m and a maximum touchdown error of 0.77m were obtained.

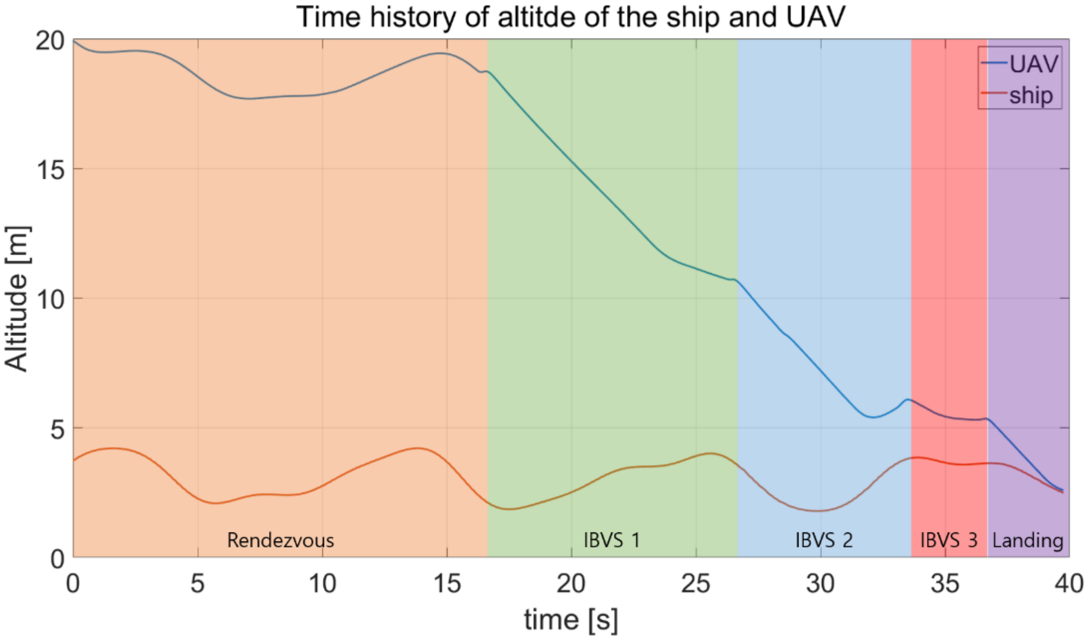


Fig. 11. Time history of the altitude of the ship and UAV.

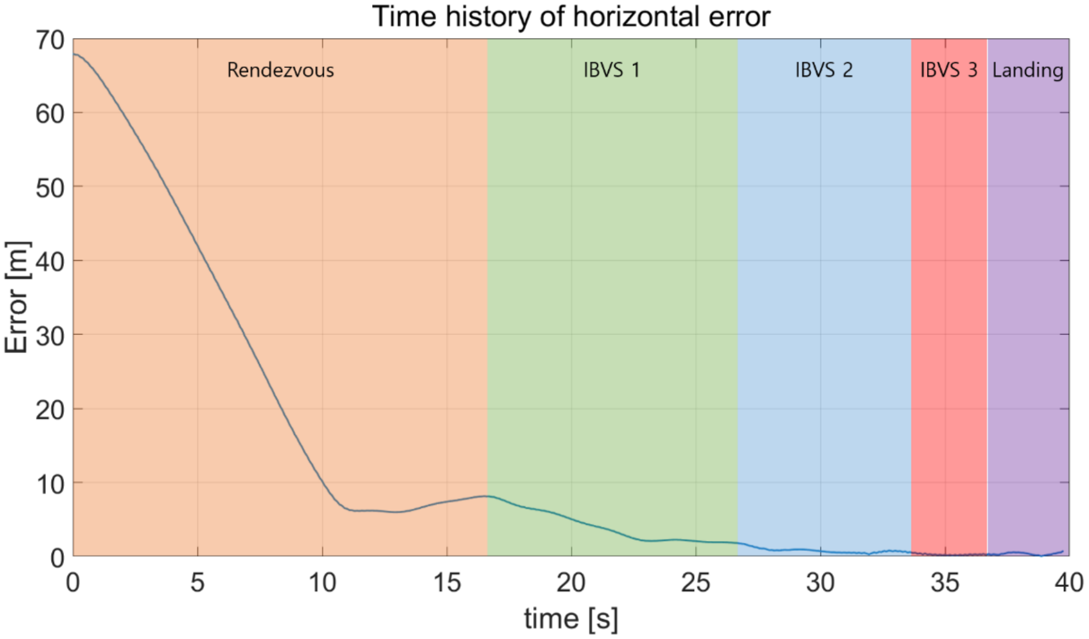


Fig. 12. Time history of horizontal error.

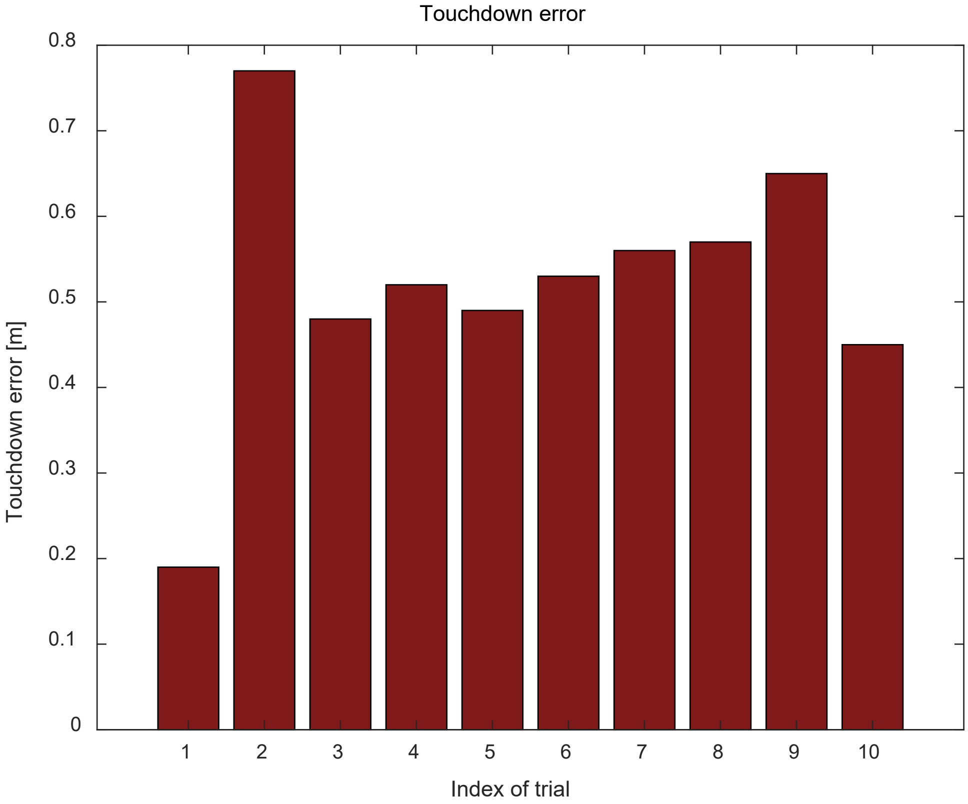
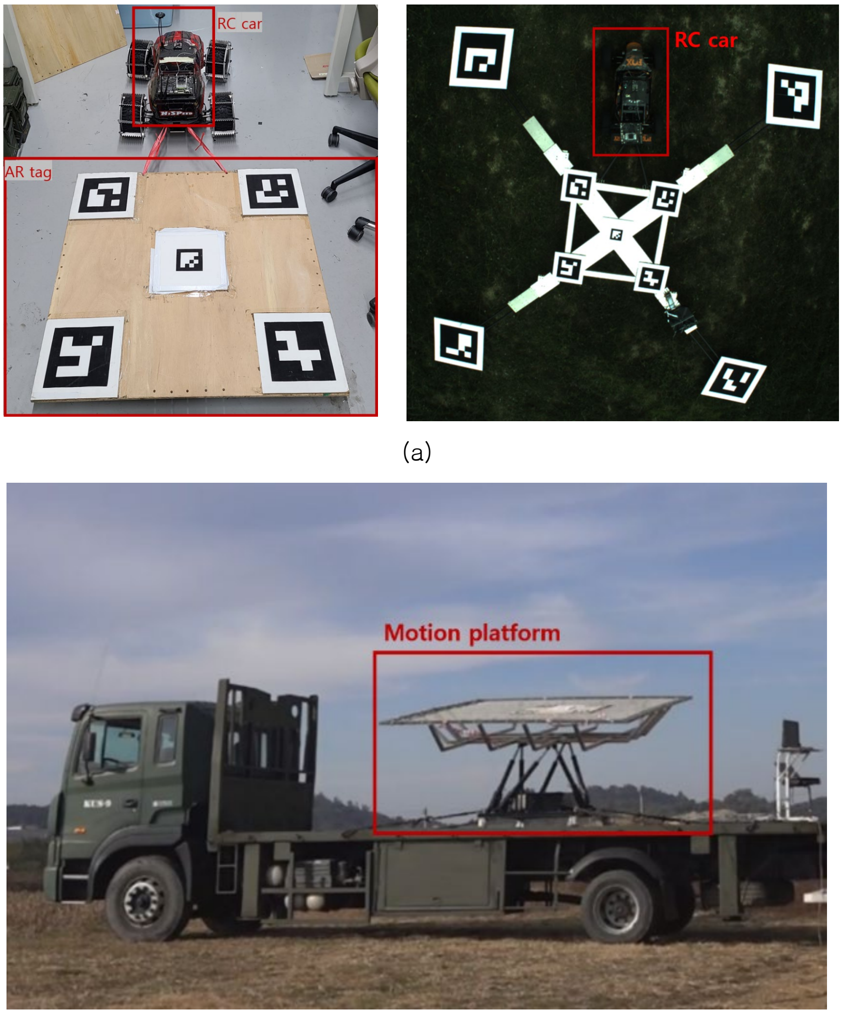


Fig. 13. Simulation result: touchdown error

Experiments

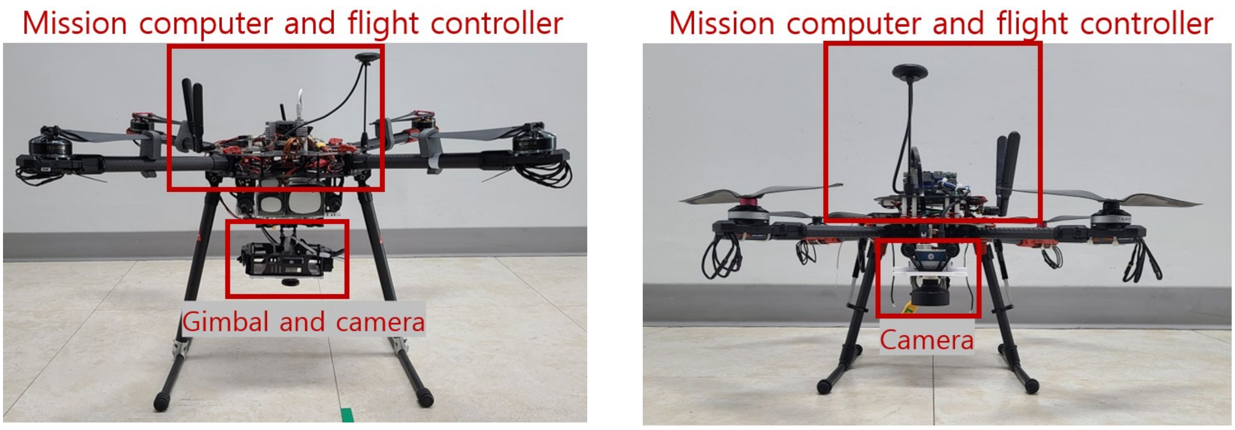
In order to verify the autonomous landing algorithm in real world, flight experiments were conducted. Landing platforms for flight experiments are designed in two types. The first one consists of a RC car and landing pad as described in Fig. 14(a). The RC car leads the landing pad that AR tags are attached. GPS/INS on the RC car estimates the velocity of the landing platform, and the data is transmitted to the UAV via Wi-Fi router. The second design of the landing platform is a combination of a truck and motion platform. Fig. 14(b) shows the landing platform setup. The motion platform on the truck is utilized to simulate motion caused by sea waves. The experiments are executed at various situations that: (a) the UAV equips the gimbal to look ahead to vertical direction to the earth to replace the transformation to the virtual coordinate frame, and starting from IBVS 2 phase; (b) simulated six-DOF ship motion (roll, pitch, yaw, surge, sway, and heave) caused by sea waves is added; and (c) all procedure starting from approach phase, and gimbal is removed so that the virtual coordinate frame transformation is adopted, but ship motion is not simulated. For all experiments, the platforms are set to move forward at a speed of around 5m/s. As shown in Fig. 15, at the situation (a) and (b), Tarot X4 is exploited for the UAV which is equipped with gimbal camera, and at the situation (c), Tarot 650 pro is used.



(a)

(b)

Fig. 14. Landing platforms for flight experiments: (a) RC car leading a landing pad; and (b) Motion platform on a truck for simulating motion of the ship.


Mission computer and flight controller
Gimbal and camera
Mission computer and flight controller
Camera

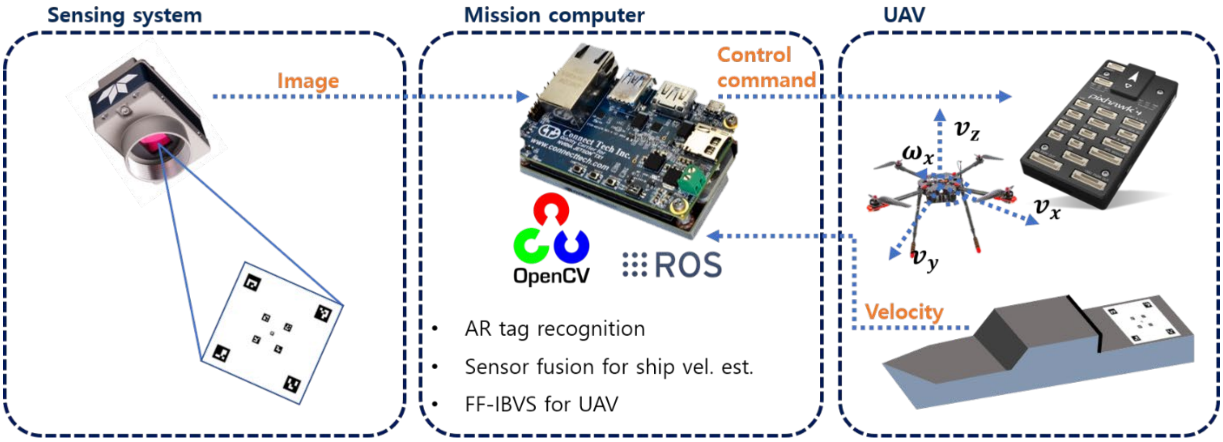
(a)

(b)

Fig. 15. UAV for experiments:

(a) Tarot X4 equipped with gimbal camera; and (b) Tarot 650 pro without gimbal camera

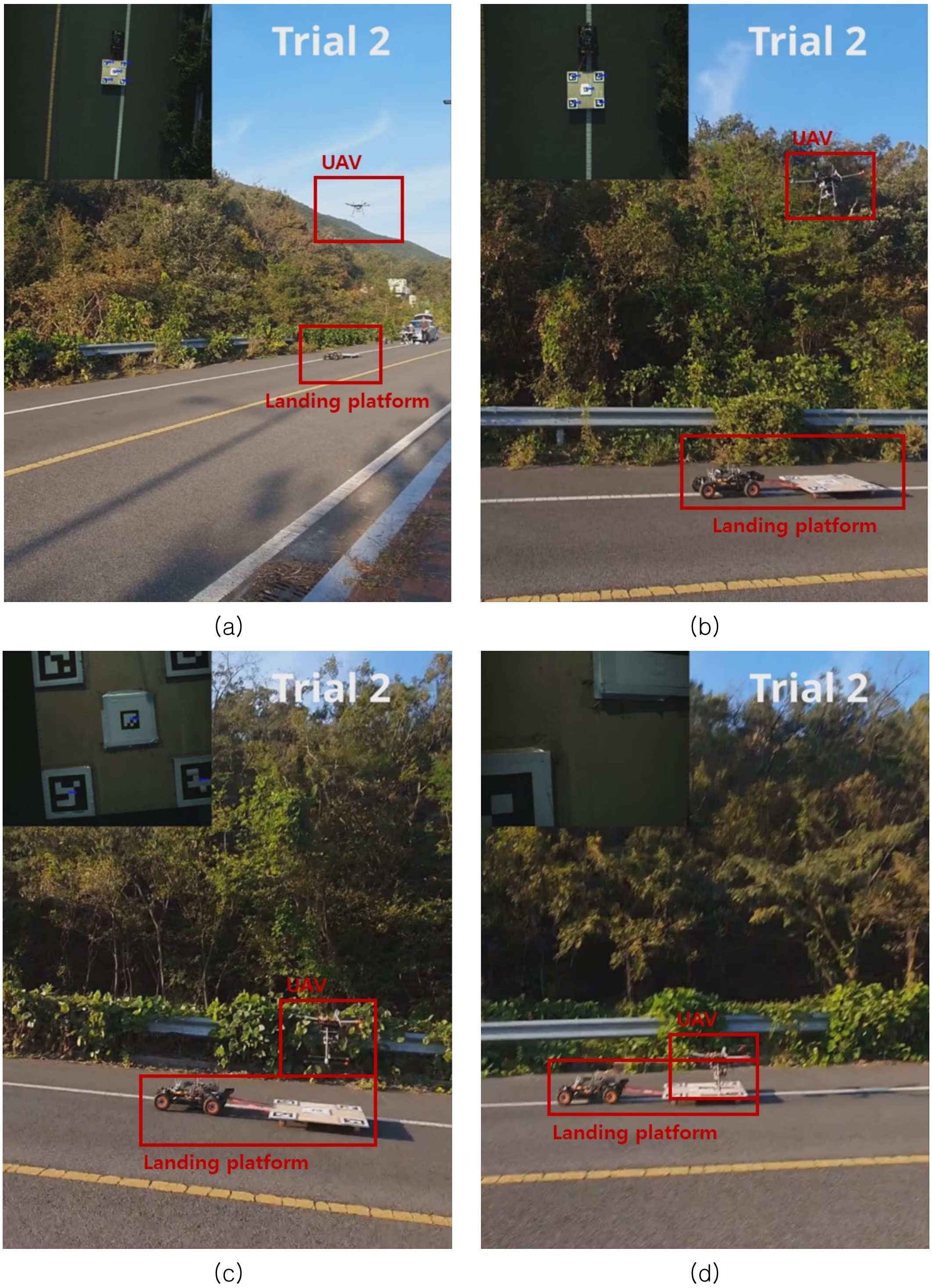
Fig. 16 shows the framework for flight experiments. For the camera and lens, a Teledyne Dalsa Genie Nano C2020 and a 3.5mm, and f/5.6 Cr Series Fixed Focal Length Lens from Edmund Optics were used, respectively. Combination of the selected camera and lens has HFOV (Horizontal Field Of View) of 102 deg and resolution of 2048 × 1536 pixels. For the GPS/INS on the ship, a Pixhawk4 is used. The image data and velocity data of the ship acquired from the camera and GPS/INS are transmitted to the Nvidia Jetson TX2 which is MC (Mission Computer) for autonomous landing via GigE (Gigabit Ethernet) and Wi-Fi communication. At the MC, the AR tag for the IBVS are extracted, and desired velocity command is calculated at 10 Hz. Generated control command is transmitted to the FC (Flight Controller) of the UAV. For the FC of the UAV, Pixhawk4 which is the same model of the GPS/INS on the ship is used.



- AR tag recognition
- Sensor fusion for ship vel. est.
- FF-IBVS for UAV

Fig. 16. Framework for flight experiments.

At the situation (a), the experiments are carried out three times. Fig. 17 shows the captures during the experiment for situation (a). Left side of the top for each capture shows the image acquired from the camera and result of AR tag extraction. The UAV start the autonomous landing when the features are recognized (Fig. 17 (a)), and then, the UAV executes IBVS. Fig. 17 (b) and Fig. 17 (c) show the captures when the UAV finishes the IBVS level 2 and 3, respectively. If the completion of the IBVS 3 is detected, the UAV descends the altitude and finishes the landing procedure (Fig. 17 (d)).



(a) (b) (c) (d)

Fig. 17. Captures during the experiment for situation (a).

The speed of the landing platform and altitude of the UAV at the situation (a) are shown in Fig. 18 and Fig. 19, respectively. The maximum speed of the landing platform is accomplished around 4.5m/s. The UAV starts landing at altitude of 7m from IBVS 2 phase. At around 7 second, IBVS 3 phase is started, and at that time, altitude of the UAV is around 4m. Around 22 second, IBVS 3 phase is finished, and at that time, altitude of the UAV is around 1m. The time from IBVS 2 phase to touchdown takes around 24 seconds to touchdown. As shown in Table 2, through three experiments, a mean touchdown error of 0.24m, standard deviation of 0.03m, and maximum touchdown error of 0.26m were obtained.

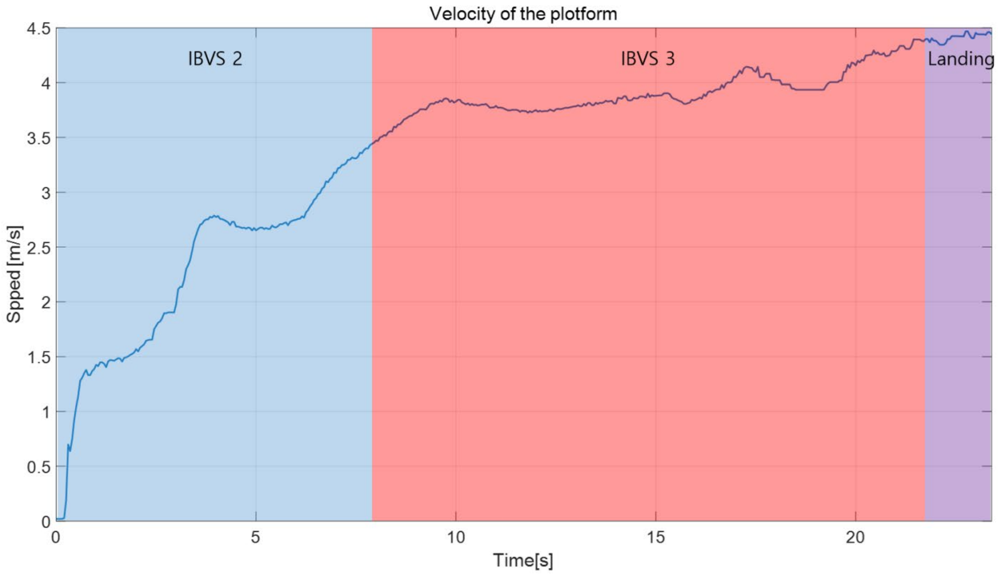


Fig. 18. Time history of the velocity of the landing platform at the situation (a)

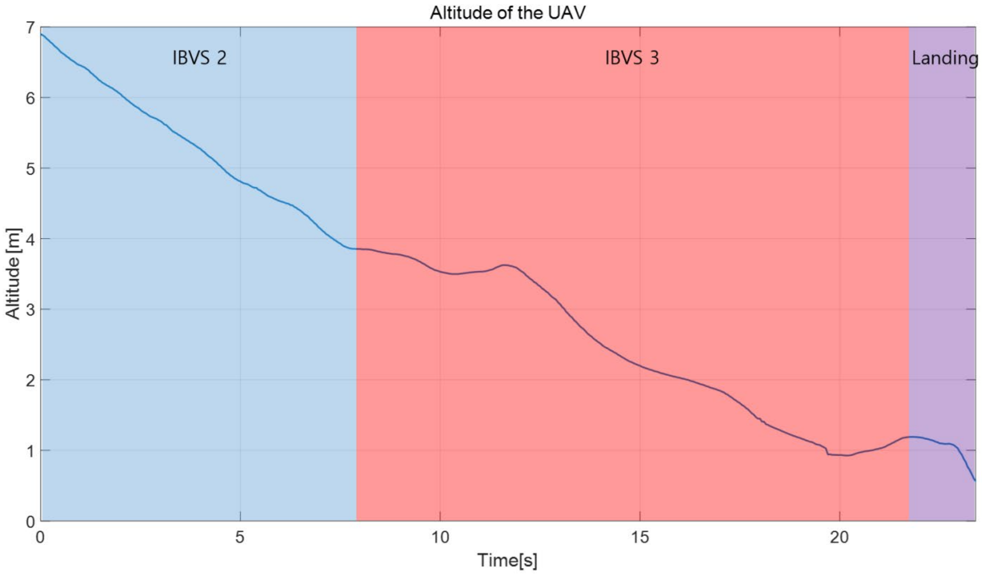
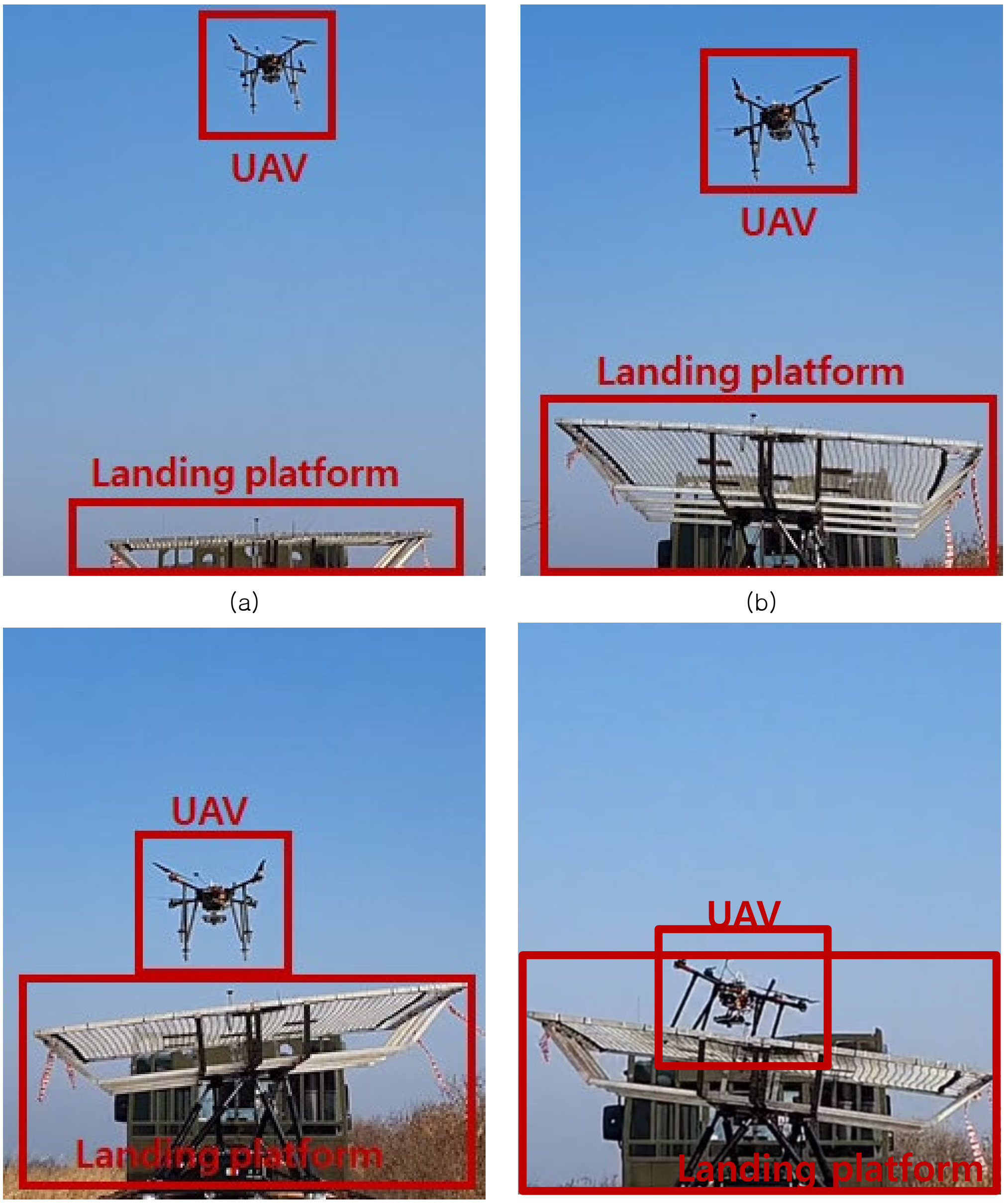


Fig. 19. Time history of the altitude of the UAV at the situation (a)

Table 2 experiment setup (a) result: touchdown error

Trial	1st	2nd	3rd	Mean error	Std. deviation
Touchdown error [m]	0.21	0.24	0.26	0.24	0.03

The situation (b) which use the motion platform is carried out. Fig. 20 shows the captures during the experiment for situation (b). The UAV uses IBVS to start landing on a landing platform that ship motion is simulated (Fig. 20(a)), and completes IBVS level 2 and 3 sequentially (Fig. 20 (b) and (c)). When IBVS 3 completion is detected, the UAV descends the altitude. Fig. 20 (d) shows the capture of touchdown moment. Fig. 21 and Fig. 22 show the time history of velocity the landing platform, and the altitude of the landing platform and the UAV, respectively. The maximum speed of the landing platform is around 5.5m/s. The altitude of the UAV at the starting time is set to same with situation (a) as 7m. In this experiment setup, touchdown error of 0.6m.



(a) (b)

(c) (d)

Fig. 20. Captures during the experiment for situation (b)

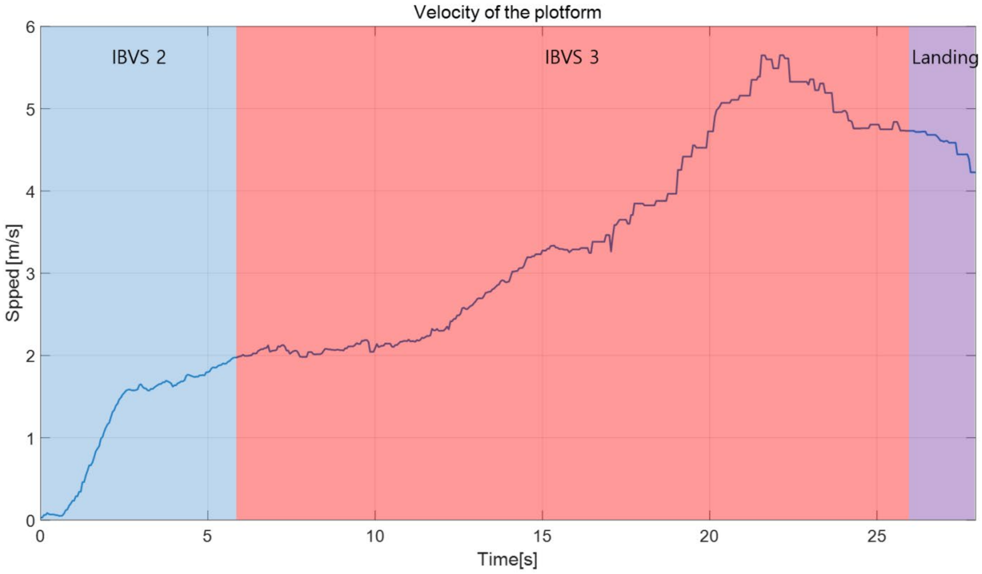


Fig. 21. Time history of the velocity of the landing platform at the situation (b)

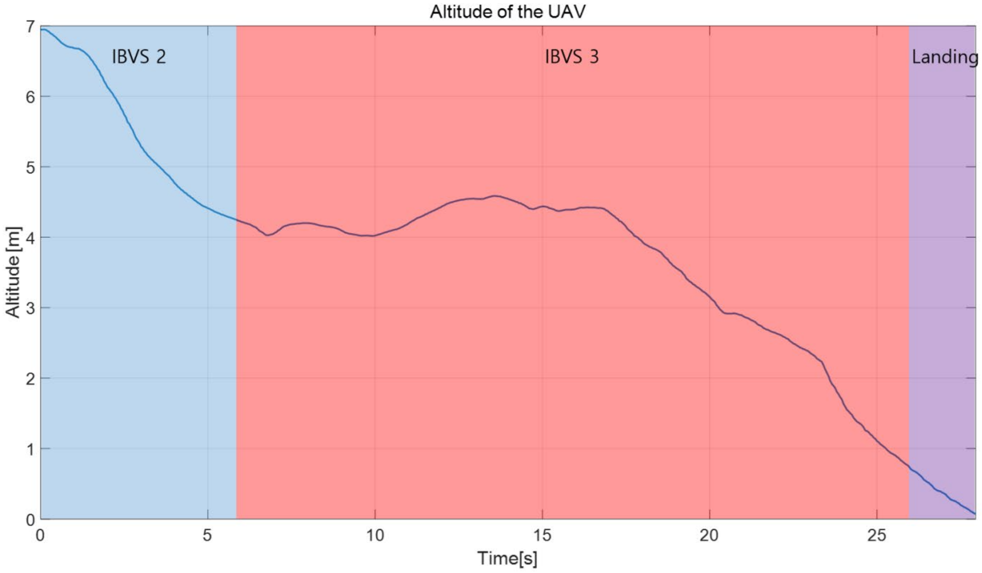


Fig. 22. Time history of the altitude of the UAV at the situation (b)

The experiments using RC car with landing pad where the situation (c) are conducted three times. Fig. 23 and Fig. 24 show the results for one out of three trials, showing the time history of velocity of the landing platform, and the altitude of the landing platform and the UAV, respectively. The landing platform maintains the reference velocity of around 5m/s while autonomous landing is executing. The Rendezvous phase start above 8m of the landing platform and approaches around the landing platform. When the state enters to the IBVS level 1, the UAV starts to descent its altitude by IBVS input. Table 3 shows the touchdown error for all experiments. Through three experiments, a mean touchdown error of 0.83m, standard deviation of 0.31m, and maximum touchdown error of 1.1m were obtained.

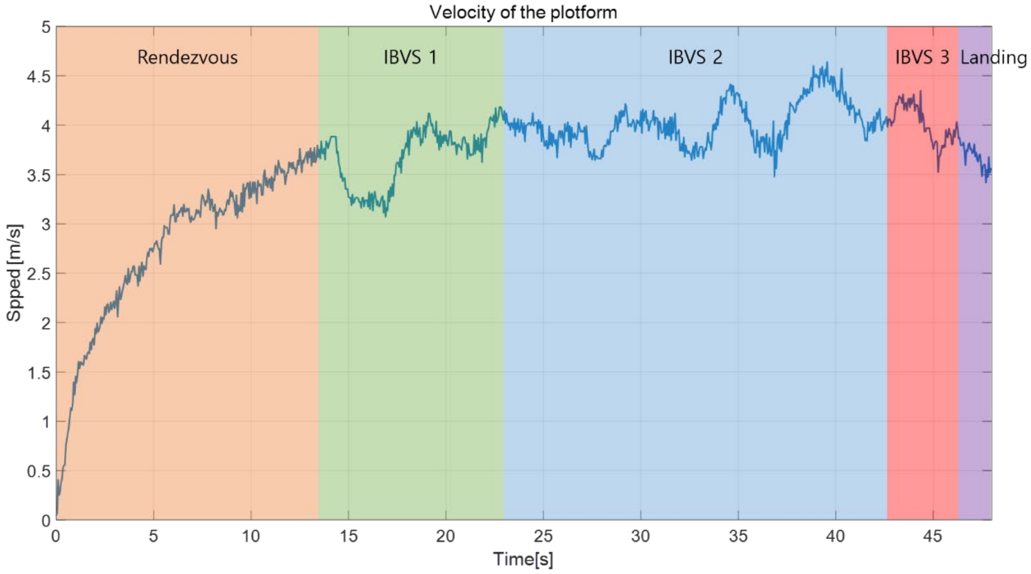


Fig. 23. Time history of the velocity of the landing platform at the situation (c)

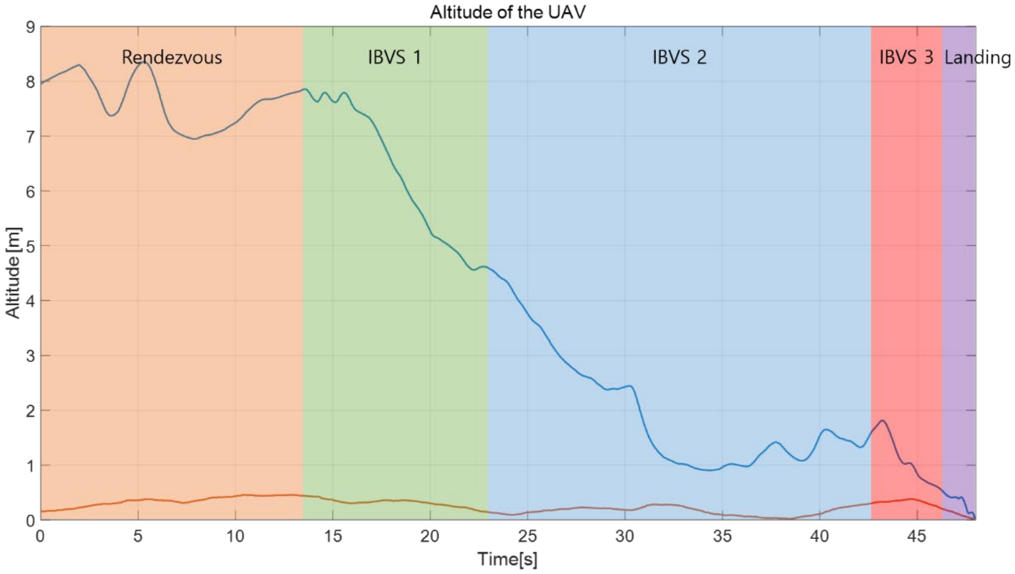


Fig. 24. Time history of the altitude of the UAV at the situation (c)

Table 3 experiment setup (c) result: touchdown error

Trial	1st	2nd	3rd	Mean error	Std. deviation
Touchdown error [m]	1.1	0.9	0.5	0.83	0.31

Conclusion

This paper proposes the vision-based autonomous ship deck landing strategy using feed-forward image-based visual servoing. Conventional IBVS scheme cannot guarantee if the target is not stationary. To overcome the convergence problem, velocity of the ship is added as feed-forward term. velocity of the ship is measured by GPS/INS on the ship deck and camera attached to the UAV and estimated using Kalman filtering. To accomplish whole of the landing procedure landing scheme is designed which including approach (Rendezvous) phase, three IBVS level according to the relative altitude between the ship and the UAV, holding, rising, and landing. As additional work, adaptive gain is adopted so that the features remain in the FOV, and the features are made to fit square to avoid the effect of the ship’s tilt. The landing scheme has state machine structure. Proposed autonomous landing algorithm is verified by simulations and experiments.

Proposed method in this paper, additional sensor (i.e. GPS/INS on the ship deck) is necessary to estimate velocity of the ship or target. However, the sensor attached to the ship complicates the system, and if there is a problem such as communication failure, autonomous landing is impossible. As a future work, a method for estimating the velocity of a ship without the aid of GPS/INS can be studied.

Reference

[1] Navid Shahriari, Silvia Fantasia, Fabrizio Flacco, and Giuseppe Oriolo. “Robotic visual servoing of moving targets.” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 77–82. 2013.

[2] Indrazno Siradjuddin, Laxmidhar Behera, T Martin McGinnity, and Sonya Coleman. “Image-based visual servoing of a 7-dof robot manipulator using an adaptive distributed fuzzy pd controller.” IEEE/ASME Transactions On Mechatronics, vol. 19, no.2, pp. 512–523, 2013.

[3] Seth Hutchinson, Gregory D Hager, and Peter I Corke. “A tutorial on visual servo control.” IEEE transactions on robotics and automation, vol. 12, no. 5, pp. 651–670, 1996.

[4] Shuo Yang, Jiahang Ying, Yang Lu, and Zexiang Li. “Precise quadrotor autonomous landing with srukf vision perception.” IEEE international conference on robotics and automation (ICRA), pp. 2196–2201, 2015.

[5] Jos´e Joaqu´ın Acevedo, Manuel Garc´ıa, Antidio Viguria, Pablo Ram´on, Bego˜na C Arrue, and Anibal Ollero. “Autonomous landing of a multicopter on a moving platform based on vision techniques.” Iberian Robotics conference, pp. 272–282, 2017.

[6] Davide Falanga, Alessio Zanchettin, Alessandro Simovic, Jeffrey Delmerico, and Davide Scaramuzza. “Vision-based autonomous quadrotor landing on a moving platform.” IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), pp. 200–207, 2017.

[7] Robson O de Santana, Leonardo A Mozelli, and Armando Alves Neto. “Vision-based autonomous landing for micro aerial vehicles on targets moving in 3d space.” 19th International Conference on Advanced Robotics (ICAR), pp. 541–546, 2019.

[8] Eric Rohmer, Surya PN Singh, and Marc Freese. “V-rep: A versatile and scalable robot simulation framework.” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1321–1326, 2013.

[9] Tarek Hamel and Robert Mahony. “Visual servoing of an under-actuated dynamic rigid-body system: an image-based approach.” IEEE Transactions on Robotics and Automation, vol. 18, no. 2, pp. 187–198, 2002.

[10] Nicolas Guenard, Tarek Hamel, and Robert Mahony. “A practical visual servo control for an unmanned aerial vehicle.” IEEE Transactions on Robotics, vol. 24 no.2, pp. 331–340, 2008.

[11] Daewon Lee, Hyon Lim, H Jin Kim, Youdan Kim, and Kie Jeong Seong. “Adaptive image-based visual servoing for an underactuated quadrotor system.” Journal of Guidance, Control, and Dynamics, vol. 35 no. 4, pp. 1335–1353, 2012.

[12] Daewon Lee, Tyler Ryan, and H Jin Kim. “Autonomous landing of a vtol uav on a moving platform using image-based visual servoing.” IEEE international conference on robotics and automation, pp. 971–976, 2012.

[13] Pedro Serra, Rita Cunha, Tarek Hamel, David Cabecinhas, and Carlos Silvestre. “Landing on a moving target using image-based visual servo control.” 53rd IEEE Conference on Decision and Control, pp. 2179–2184, 2014.

[14] Quang Huy Truong, Thomas Rakotomamonjy, Armin Taghizad, and Jean-Marc Biannic. “Vision-based control for helicopter ship landing with handling qualities constraints.” IFAC-PapersOnLine, vol. 49, no. 17, pp. 118–123, 2016.

[15] Thomas Rakotomamonjy and Quang Huy Truong. “Helicopter ship landing using visual servoing on a moving platform.” IFAC-PapersOnLine, vol. 50, no. 1, pp. 10507–10512, 2017.

[16] Iryna Borshchova and Siu O’Young. “Visual servoing for autonomous landing of a multi-rotor uas on a moving platform.” Journal of Unmanned Vehicle Systems, vol. 5, no. 1, pp. 13–26, 2016

[17] Iryna Borshchova and Siu O’Young. “Marker-guided auto-landing on a moving platform.” International Journal of Intelligent Unmanned Systems, 2017.

[18] Wynn, Jesse S and McLain, Timothy W, "Visual servoing with feed-forward for precision shipboard landing of an autonomous multirotor," American Control Conference (ACC), pp. 3928-3935, 2019

[19] Sergio Garrido-Jurado, Rafael Mu˜noz-Salinas, Francisco Jos´e Madrid-Cuevas, and Manuel Jes´us Mar´ın-Jim´enez. “Automatic generation and detection of highly reliable fiducial markers under occlusion.” Pattern Recognition, vol. 47 no. 6, pp. 2280–2292, 2014.

[20] Franc¸ois Chaumette and Seth Hutchinson. “Visual servo control. i. basic approaches. IEEE Robotics & Automation Magazine,” vol. 13 no. 4, pp. 82–90, 2006.

[21] Tristan Perez and Mogens Blanke. “Simulation of ship motion in seaway.” Computer Science; Technical Report; The University of Newcastle: Callaghan, Australia, pp. 1–13, 2002.

[22] Jose Luis Sanchez-Lopez, Srikanth Saripalli, Pascual Campoy, Jesus Pestana, and Changhong Fu. “Toward visual autonomous ship board landing of a vtol uav.” International conference on unmanned aircraft systems (ICUAS), pp. 779–788, 2013.

[23] James Rankin. “An error model for sensor simulation gps and differential gps.” In Proceedings of IEEE Position, Location and Navigation Symposium-PLANS, pp. 260–266, 1994.

[24] Randal W Beard and Timothy W McLain. “Small unmanned aircraft.” Princeton university press, 2012.