A Cooperative Framework for Autonomous Landings of Quadrotors using Vision on a Moving UGV

Rishab Balasubramanian*
NIT Trichy, Tiruchirappalli – 620015, India.

P.B. Sujit†
IISER Bhopal, Bhopal – 462066, India.

Unmanned ground vehicles (UGVs) have become popular for deploying and recharging quadrotors in various persistent applications. In order to accurately land, the UAV on the UGV efficient landing controllers must be designed. Further, a single UGV may need to service several quadrotors and hence the UGV and UAV need to coordinate with each other for landing and take-off. In this paper we present a State Dependent Riccati Equation (SDRE)-based landing controller with vision in the loop and a cooperative scheduling scheme for sequential landing of the UAV on the UGV with active interaction with the vehicles. Simulations results are presented to show the efficacy of the landing and scheduling for different trajectories.

I. Nomenclature

W	=	Inertial frame (ECEF)
B	=	Body Fixed Frame of the UAV
x_d	=	position of the UAV along X axis in W
y_d	=	position of the UAV along Y axis in W
z_d	=	position of the UAV along Z axis in W
v_{dx}	=	velocity of the UAV along X axis in W
v_{dy}	=	velocity of the UAV along Y axis in W
v_{dz}	=	velocity of the UAV along Z axis in W
a_{dx}	=	acceleration of the UAV along X axis in W
a_{dy}	=	acceleration of the UAV along Y axis in W
a_{dz}	=	acceleration of the UAV along Z axis in W
ϕ_d	=	roll of UAV
θ_d	=	pitch of UAV
ψ_d	=	yaw of UAV
ω_d	=	angular velocity of UAV about Z axis
T	=	desired thrust
ϕ_{des}	=	desired roll of UAV
θ_{des}	=	desired pitch of UAV
ω_{des}	=	desired angular velocity for UAV about Z axis
m	=	mass of UAV
g	=	acceleration due to gravity
x_r	=	position of the target along X axis in W
y_r	=	position of the target along Y axis in W
z_r	=	position of the target along Z axis in W
v_{rx}	=	velocity of the target along X axis in W
v_{ry}	=	velocity of the target along Y axis in W
v_{rz}	=	velocity of the target along Z axis in W

*Under graduate, Department of Instrumentation And Control Engineeering. Email: rishab.edu@gmail.com
†Associate Professor, Department of Electrical Engineering and Computer Science. Email: sujit@iiserb.ac.in

a_{rx} = acceleration of the target along X axis in W
a_{ry} = acceleration of the target along Y axis in W
a_{rz} = acceleration of the target along Z axis in W
ψ_r = yaw of target
ω_r = angular velocity of target about Z axis
R_b = rotation matrix corresponding to transformation from B to W
R_w = rotation matrix corresponding to transformation from W to B

II. Introduction

Small unmanned aerial vehicles like quadrotors are highly popular to conduct surveys. We are interested in using these vehicles for cargo delivery during these pandemic times like COVID-19, where the interaction between human needs to be minimized while ensuring supply chain is maintained. This requires the UAV to take off from the cargo vehicle, deliver the package and return to the vehicle for landing. These sequence of actions can be automated to delivery application. Another similar application could be habitat mapping or convoy protection. Further, if the UAV carrying vehicle is an UGV then the operation can be autonomous without any harm to humans while performing operations in disease prone regions or remote regions.

Most of the works focus on either route planning problem for the UAV and the UGV to meet the requirements with minimal refueling locations [1, 2] or on robust landing of the UAV on the UGV [3–5]. In these two works, a single UAV is used for planning paths. The mission can be accomplish faster and cover large regions in the presence of multiple UAVs. However, with multiple UAVs, either the platforms needs to be multiple landing pads or in many cases they need share a single landing platform. In order to minimize logistics, the vehicles need to share a single lading platform and hence there is a need for the vehicles to coordinate with the UGV on the availability of the landing pad. The focus of this paper is to develop accurate landing controller for the quadrotors and also develop a cooperative mechanism between UAV and UGV to ensure sequential and safe landings.

Autonomous landing of aerial vehicles on moving targets involves (a) detection of the landing target using computer vision techniques and (b) determining guidance or controller commands to the vehicle for accurate landing. The are several computer vision techniques that can be used to detect the landing like optical flow [6], stereo vision [7] [8], image segmentation [9][10][11] and edge detection [12]. Once detected, the target information is then used to determine appropriate guidance or control commands for landing on the target.

Over the years, several landing controllers have been developed. Serrra et al. [4] developed an image-based visual servo control to land a quadrotor on a moving target. Falanga et al. [13] use vision to detect a tag for quadrotor landing on a moving rover. Cocchioni et al. [14] developed different vision algorithms to detect the landing pad patterns for robust landing. Li et al. [15] developed a vision algorithm to contain drift for a hover by taking the initial take-off image for a vertical landing. Ghommam and Saad [5] developed an adaptive tracking control scheme with backstepping and dynamic surface control for landing a moving target whose position is assumed to be known. Vlantis et al. [16] proposed a nonlinear model predictive controller for a quadotor to land on moving robot having inclined landing surface. Hoang et al. [17] use visual-bag-of-words method to detect the landing pad and develop a controller for tracking and landing using feedback linearization. Wang and Bai [18] use red LEDs to mark the landing pad and use color segmentation to detect the landing pad using vision. A PID control law is used to land on the vertically on stationary target whose pose is simulated as that of a ship. Most of the controllers are nonlinear controller with no optimality guarantees.

Model Predictive Controller (MPC) is the most common optimal controller that is used for quadrotor landing [19, 20]. Although, MPC determines optimal solutions, it is computationally intensive and hence difficult to implement for real-time application in the autopilot. Hence, an off-board computer computes the controls and commands the vehicle. Alternatively, the dynamics can be linearized for real-time implementation on a small onboard companion computer [21]. In this paper, we consider the complete nonlinear model and determine optimal control inputs by using the nonlinear optimal control framework on State-Dependent Ricatti Equation approach (SDRE) [22, 23].

Scheduling vehicles to land on the UGV enables multiple UAVs to perform the task while meeting payload and fueling constraints. Matthew et al. [24] developed an approximation algorithm to meet the schedule of different vehicles. The location to refuel is the same. Sujit and Beard [25] developed a landing schedule

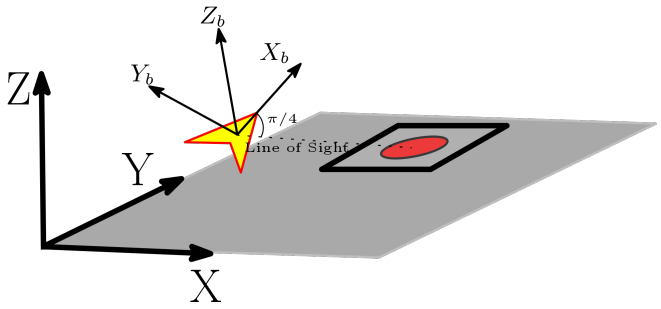


Fig. 1 Representation of W and B frames, and the general landing problem

based on slots for a fixed landing locations. Rucco et al. [26] proposed an optimal control framework for cooperative landing of a UAV on to a moving UGV. However, the optimal controller is time consuming and not robust to UGV trajectory changes. In this article, we develop a cooperative scheduling strategy for the UAVs to land sequentially on the moving UGV as it meets its schedule. Due to change in terrain and UGV motion, it can communicate the UAV to not land at certain locations. During this process, the UAV tracks the UGV and when it obtains a signal to land, the UAV proceeds to land. Unlike the other works, there is a continuous interaction between the UAV and the UGV for landing.

The main contribution of this paper is the development of SDRE-based accurate landing controller and a cooperative landing framework for quadrotors to land on the UGV. The rest of the paper is organized as follows. The landing problem formulation is presented in Section III and the SDRE-based landing controller is designed in Section IV. In Section V, we describe the position and velocity estimation of the target using vision. The cooperative strategy is described in Section VI. The evaluation of the proposed framework is carried out through simulations which is described in Section VII and the conclusions are presented in Section VIII.

III. Problem Formulation

Consider a UAV that needs to land on a target as shown in Figure 1. A camera is mounted on the vehicle which is inclined at an angle of $\pi/4$ with respect to the body of the UAV. We use the image information from the camera with a Kalman filter to estimate the position and velocity of the moving landing target on the ground. The goal is to drive the UAV from its current co-ordinates $(x_d(t), y_d(t), z_d(t))$ to the coordinates of the target $(x_r(t), y_r(t), z_r(t))$, while simultaneously ensuring the relative velocities between the UAV and target approaches zero. The notations d denotes drone (UAV) and r represents UGV (rover). The objective is

$$\lim_{t\to\infty} x - x_r = y - y_r = z - z_r = v_{dx} - v_{rx} = v_{dy} - v_{ry} = v_{dz} - v_{rz} = 0. \tag{1}$$

We consider the landing problem as an infinite horizon regulator problem. The errors in the body frame B are considered as the states for the regulatory system, which are used to generate the controller commands. The desired pitch (θ_{des}), roll (ϕ_{des}), thrust (T), and angular velocity (ω_{des}) of the UAV needs to be determined using the developed optimal guidance strategy based on SDRE formulation. We assume the lateral and longitudinal accelerations are decoupled. The architecture of the control strategy is shown in Fig.2, where develop only the SDRE control and vision algorithms in this paper. We assume that the autopilot converts the thrust, roll, pitch, and z-angular velocity commands to the desired motor commands. We do not explicitly consider the UAV dynamics in this article.

Let us define the errors in position and velocity between the target and UAV in the inertial frame W as

$$e^W_{q,ax} = r^W_{q,ax} - d^W_{q,ax} \tag{2}$$

where $q \in \{pos, vel\}$ is the state used for error calculation (position or velocity), ax is the direction in $W(x, y, or\ z)$. The errors in W frame is converted to B frame as

$$\bar{e}^B_q = R_w \bar{e}^W_q \tag{3}$$

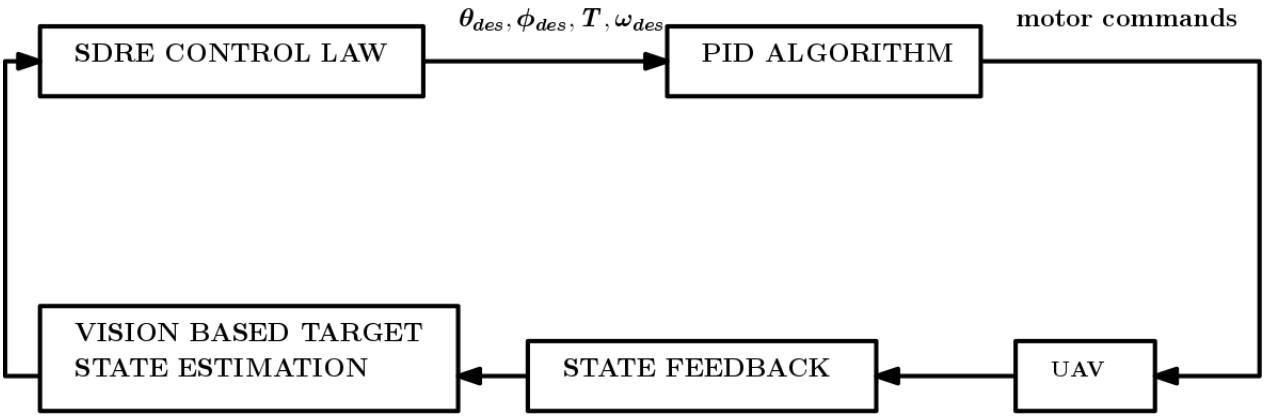


Fig. 2 The figure shows the architecture of the control, feedback, and estimation model used.

where

$$\bar{e}_q^B = \begin{bmatrix} e_{q,x}^B \\ e_{q,y}^B \\ e_{q,z}^B \end{bmatrix},$$

$$\bar{e}_q^W = \begin{bmatrix} e_{q,x}^W \\ e_{q,y}^W \\ e_{q,z}^W \end{bmatrix},$$

$$R_w = \begin{bmatrix} \cos(\psi_d)\cos(\theta_d) & -\sin(\psi_d)\cos(\phi_d) + \sin(\phi_d)\sin(\theta_d)\cos(\psi_d) & \sin(\psi_d)\sin(\phi_d) + \cos(\phi_d)\cos(\psi_d)sin(\theta_d) \\ \sin(\psi_d)\cos(\theta_d) & \cos(\psi_d)\cos(\phi_d) + \sin(\phi_d)\sin(\theta_d)\sin(\psi_d) & -\sin(\phi_d)\cos(\psi_d) + \sin(\psi_d)\sin(\theta_d)\cos(\phi_d) \\ -\sin(\theta_d) & \cos(\theta_d)\sin(\phi_d) & \cos(\theta_d)\cos(\phi_d) \end{bmatrix}. \tag{4}$$

We know that

$$\dot{e}_{pos,ax} = e_{vel,ax}. \tag{5}$$

Therefore, we will refer to errors in positions along x, y, z in B as e_x, e_y, e_z and their respective error velocity in B as $\dot{e_x}, \dot{e_y}, \dot{e_z}$. We shall also define the error in yaw between the target and UAV as

$$e_\psi = \psi_r - \psi_d. \tag{6}$$

The only forces impacting the motion of the UAV are the weight, and thrust due to the rotors. The thrust acts along the z direction in B and, gravity along $-z$ direction in W. Thus the force due to gravity acting on the UAV in W is:

$$Fg^W = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{7}$$

We get the force due to gravity in B as:

$$Fg^B = R_w Fg^W = \begin{bmatrix} mg\sin\theta_d \\ -mg\sin\phi_d\cos\theta_d \\ -mg\cos\phi_d\cos\theta_d \end{bmatrix} \tag{8}$$

Using small angle approximations, we obtain

$$Fg^B = \begin{bmatrix} mg\theta_d \\ -mg\phi_d \\ -mg \end{bmatrix} \tag{9}$$

Thus, the net force acting on the UAV in B is

$$F^B = \begin{bmatrix} mg\theta_d \\ -mg\phi_d \\ T - mg \end{bmatrix} \tag{10}$$

where T is the thrust due to the motors. We can find the body acceleration by dividing F^B by m. We shall assume here, we know the accelerations of the rover, and thus we can get the relative acceleration (error) between the target and quadrotor. We define our new virtual control inputs u_1, u_2, u_3 as

$$e_{q,ax} = \begin{bmatrix} a_{rx} - a_{dx} \\ a_{ry} - a_{dy} \\ a_{rz} - a_{dz} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \tag{11}$$

From Equation (11), we obtain

$$\theta_{des} = (a_{rx} - u_1)/g \tag{12}$$
$$\phi_{des} = (u_2 - a_{ry})/g \tag{13}$$
$$T = m(a_{ry} - u_3) + mg. \tag{14}$$

Therefore, we define the linearized state model in the form of:

$$\dot{X} = A(X)X + B(X)U \tag{15}$$

where:

$$X = \begin{bmatrix} e_x \\ \dot{e_x} \\ e_y \\ \dot{e_y} \\ e_z \\ \dot{e_z} \\ e_\psi \end{bmatrix}, A(X) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B(X) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \omega_{des} \end{bmatrix} \tag{16}$$

Using the linearized state model, we will design the SDRE controller for landing in the next section.

IV. SDRE Control Law

A general infinite time optimization problem can be defined as:

$$min\ J = \int_0^\infty X^T Q(X) X + U^T R(X) U \tag{17}$$

where X and U are the states and inputs to our system respectively, related by (15). The optimal control solution is given by:

$$U^* = -R(X)^{-1} B(X)^T P X \tag{18}$$

where P is obtained by solving the Algebraic Riccati Equation (ARE)

$$A(X)^T P + P A(X) - P B(X) R(X)^{-1} B(X)^T P + Q(X) = 0 \tag{19}$$

The weight $Q(X)$ provides weights for the states in X. If the UAV descends too quickly, there is a high possibility it loses sight of the target, which results in landing failiure. Thus, the error in position along z should reduce slower than along x and y. It is critical to penalize the error in velocity as the UAV approaches the target. This prevents overshoot, and allows for the UAV to match its speed with the target at the time of landing. Finally, we would also like to place sufficient weight on the error in heading for vision-based landing, as this allows us to track the rover from behind. As stated previously, to ensure we are constantly behind the rover, the error in velocity is heavily penalized. Once $R(X)$ is defined, we adjust $Q(X)$ so that we can acheive swift landing, without losing vision of the target (for the vision-based landing scheme).

A. Stability Analysis

We shall now consider the stability of the proposed controller. Let us define the function $f(X)$ as:

$$f(X) = AX \tag{20}$$

Let us assume that the states are completely observable. Thus, from [23], we can say that the controller is locally stable about the origin if it satisfies

1) $f(X) \in C^1$
2) $f(0) = 0$
3) $B \neq \emptyset$
4) $\{A, B\}$ is controllable
5) $Q(X) \geq 0$ and $R(X) > 0$

We will now prove the above conditions for the given controller

1) We can observe that:

$$f(X) = AX = \begin{bmatrix} \dot{e_x} \\ 0 \\ \dot{e_y} \\ 0 \\ \dot{e_z} \\ 0 \\ 0 \end{bmatrix} \in C^1 \tag{21}$$

2) When $X = 0$,

$$f(X) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{22}$$

3)

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \neq \emptyset \tag{23}$$

4) Let

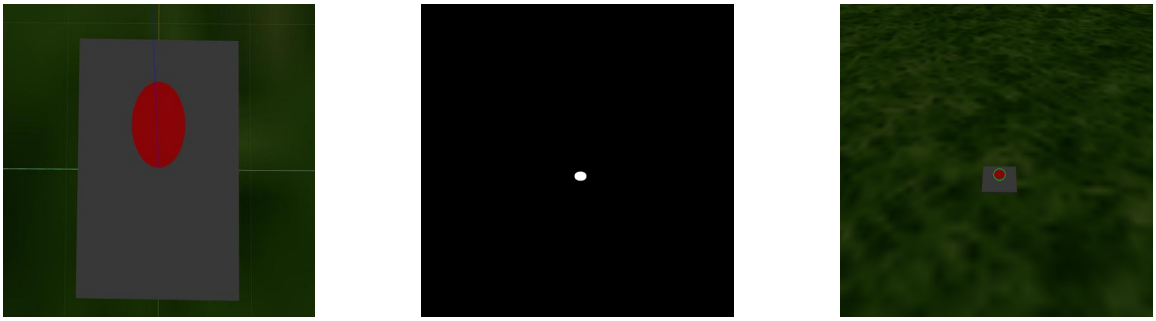
$$Q_c = [B, AB, A^2B, ...A^6B] \tag{24}$$

We use MATLAB ctrb function to verify that the rank$(Q_c) = 7$ = number of states.

5) This criteria is satisfied by choosing the diagonal elements of Q and R to be greater than 0. We shall observe that this is satisfied in the following sections, where the Q and R matrices are given for specific applications. Thus the system is controllable.

V. Position And Velocity Estimation Using Vision

To detect and track the target ground vehicle, a camera is mounted atop the UAV at an angle of $\pi/4$ with respect to the X axis in B, as in Fig 1. A grey colored landing space of dimension $1m \times 1m$ is placed on top of the rover, on which a red colored target is attached. The input video from the camera is converted from



(a) **(b)** **(c)**

Fig. 3 a) The landing zone atop the rover with a red circular visual target b) Extracted mask of the red target c) The detected position of the target

RGB to HSV and YCrCb color spaces. This is because of the robustness of these color spaces to lighting conditions over RGB. In the HSV model, colors of each hue are arranged in a radial slice, around a central axis of neutral colors which ranges from black at the bottom to white at the top. Hue thus defines the dominant color of an area. The saturation dimension resembles various tints of bright color measuring the colorfulness, and the value dimension resembles luminance. The Hue, Saturation and Value are calculated as:

$$H = \frac{\arccos(0.5(R-G)+(R-B))}{\sqrt{(R-G)^2+(R-B)(G-B)}}, \tag{25}$$

$$S = 1-3\frac{\min(R,G,B)}{R+G+B} \tag{26}$$

$$V = \frac{R+G+B}{3} \tag{27}$$

In the YCrCb color space the color is represented by luma Y, constructed as a weighted sum of RGB values, and two color difference values Cr and Cb that are formed by subtracting luma from RGB red and blue components.

$$Y = K_r R + K_g G + K_b B, \tag{28}$$

$$Cr = R - Y, \tag{29}$$

$$Cb = B - Y. \tag{30}$$

In each computed color space, a mask is applied to remove only the regions which contain red colors. To remove noise, morphological transforms are used. The two resulting masks are then combined to obtain the final mask of the image. On this, blob detection algorithm is used to find the largest contour. The results are shown in Fig.3. After blob detection, the center of the contour is calculated, and the coordinates received in pixel values are transformed to coordinates in the body fixed frame using homography transformation and Single Value Decomposition (SVD). We assume the following

1) The target vehicle moves along a horizontal plane.
2) The altitude at which the quadrotor is flying is known to us.
3) The heading and angular velocity of the target vehicles is known.
4) The target moves with a constant acceleration

Assumption (3) arises because of the use of color detection algorithms to estimate the position of the target. We can overcome this problem by using QR tags or Aruco markers to determine the orientation of the rover, but that is beyond the focus of this work. Assumption (4) is a major limitation to the vision-only based landing scheme. This can be overcome if the vision system is fused with data from the rover, to get a full state observable system, or by using the cooperative strategy as proposed in Sec.VI.A.

A. Computing the Homography matrix

We can define a transformation from image coordinates to global coordinates by using homographic transformations. Let us define the homographic matrix M as:

$$M = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix},$$

and $M^{'}$ as:

$$M^{'} = \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{20} \\ m_{21} \\ m_{22} \end{bmatrix}.$$

Let the image co-ordinates be (a_1, b_1) and the global co-ordinates be (X_1, Y_1). Then we relate the two as:

$$\begin{bmatrix} a_1 \\ b_1 \\ 1 \end{bmatrix} \approx \begin{bmatrix} wa_1 \\ wb_1 \\ w1 \end{bmatrix} = M \begin{bmatrix} X_1 \\ Y_1 \\ 1 \end{bmatrix}.$$

To solve for the homography matrix, we use five directional vectors, in B that correspond to (0,0), (w,0), (0,h), (w,h), (w/2, h/2) in the image frame, where w and h are the width and height in pixels of the image. These vectors are multiplied with the rotational matrix, R_w to obtain the directional vectors in the world frame. Once the directional vectors are found in the world frame, their intersection with the $z = z_r$ plane is computed. Consider the intersection points as (X_i, Y_i, z_r) and the corresponding image coordinate as (a_i, b_i). The matrix C is then computed using each point i as follows:

$$C = \begin{bmatrix} X_i & Y_i & 1 & 0 & 0 & 0 & -a_iX_i & -a_iY_i & -a_i \\ 0 & 0 & 0 & X_i & Y_i & 1 & -b_iX_i & -b_iY_i & -b_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \tag{31}$$

Once generated the entries of the homography matrix are then computed by solving the homogeneous equation:

$$CH^{'} = 0. \tag{32}$$

This can be done by using Singe Variable Decomposition (SVD). Once the homography matrix is determined, the image coordinates can be transformed to determine the position of the target relative to the UAV as:

$$e_x^W = \frac{(m_{00}a + m_{01}b + m_{02})}{(m_{20} + m_{21} + m_{22})}, \tag{33}$$

$$e_y^W = \frac{(m_{10}a + m_{11}b + m_{12})}{(m_{20} + m_{21} + m_{22})}, \tag{34}$$

where (a, b) is the pixel coordinates of the center of the red part of the target.

B. Kalman Filter

A Kalman Filter is now applied over the coordinates obtained from the vision system. Lets say that our prediction of the rover's next states $\hat{X}_{k+1}$, based on current data is given as:

$$\underbrace{\begin{bmatrix} x_{k+1} \\ v_{x_{k+1}} \\ y_{k+1} \\ v_{y_{k+1}} \end{bmatrix}}_{\hat{X}_{k+1}} = \underbrace{\begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{\mathrm{F}} \underbrace{\begin{bmatrix} x_k \\ v_{x_k} \\ y_k \\ v_{y_k} \end{bmatrix}}_{X_k} \tag{35}$$

where $k+1$ denotes the next time step and x, v_x, y, v_y denote the expected position and velocity of the target relative to the UAV along the X and Y axis in W respectively. F is the co-efficient matrix, X_k is the previous states and $\hat{X}_{k+1}$ is the prediction matrix. Then the new expected covariance matrix $\hat{P}_{k+1}$ is:

$$\hat{P}_{k+1} = FP_kF^T + Q_k \tag{36}$$

where P_k is the co-variance of the filter output and Q_k is the noise matrix for the filter. Let the mean of the measured data be Z_k and the noise covariance be R_k, which we can define as:

$$Z_{k+1} = H_kX_k \tag{37}$$
$$R_{k+1} = H_kX_kH_k^T \tag{38}$$

where H_k is the sensor matrix. The Kalman gain is then computed as:

$$K = \hat{P}_{k+1}(\hat{P}_{k+1} + R_{k+1})^{-1} \tag{39}$$

The new states X_{k+1} and co-variance P_{k+1} are thus calculated as:

$$X_{k+1} = \hat{X}_{k+1} + K(Z_{k+1} - \hat{X}_{k+1}) \tag{40}$$
$$P_{k+1} = \hat{P}_{k+1}(1 - K) \tag{41}$$

Fig. 4 shows the results of the Kalman based prediction system. We can observe that although the prediction in target position is quite stable, the prediction in its velocity is oscillatory. This is caused by the oscillations of the UAV during its motion.

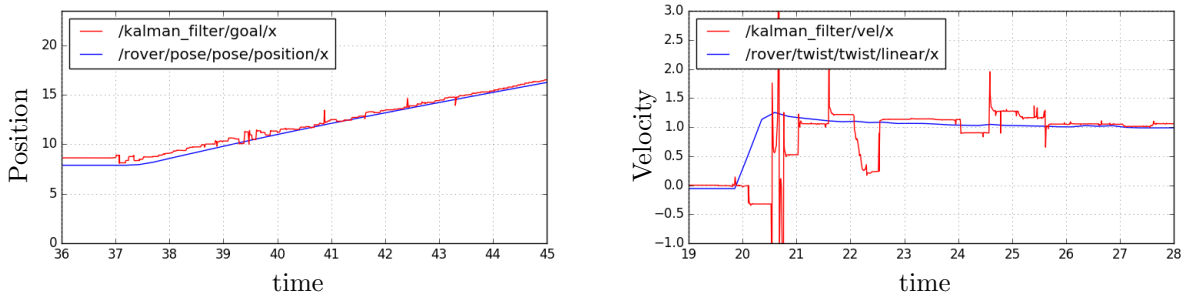


Fig. 4 The estimated position and velocity as the rover moves along the X axis with a speed of 1m/s

VI. Cooperative Strategy

When the target is maneuvering, it becomes difficult for the UAV to land on the target because of prediction and localization errors and sharp changes in accelerations and velocities, which are difficult to maintain at the time of landing. Even with heading estimations of the target with QR tags or Aruco markers,

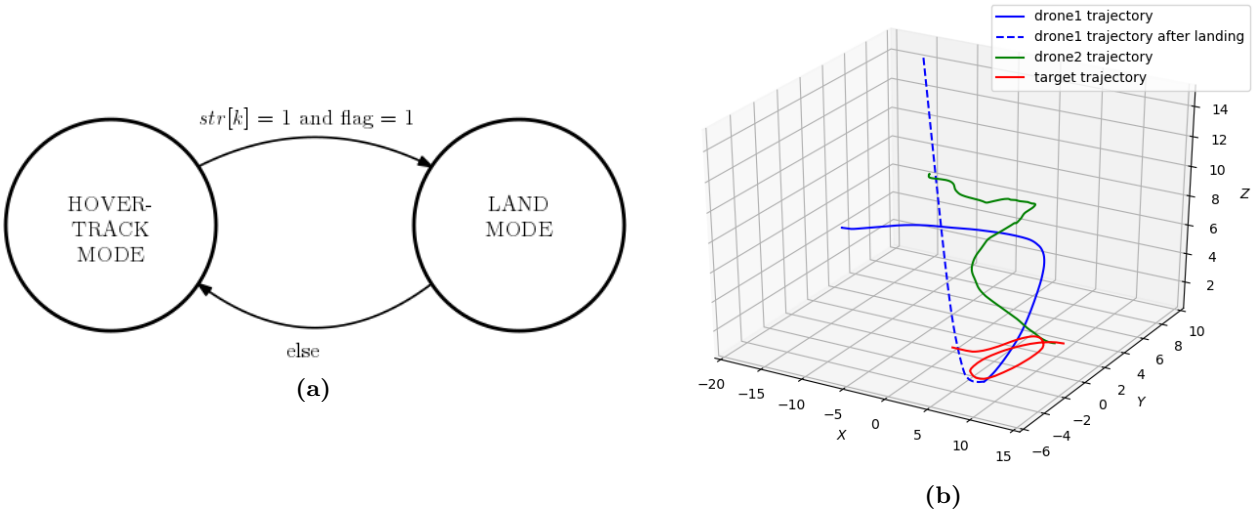
the visibility of such visual aids in diverse lighting conditions is questionable, which seriously hinders the ability of the vision-only model to land on targets which are moving along suddenly curving trajectories. Here we put forth a solution to this problem using a coordination strategy between the aerial and ground vehicles. We also extend the proposed strategies to a multi-UAV setting where the UAVs land sequentially on the target one after another without any collisions.

A. Cooperative strategy For maneuvering targets

We assume that there exists some mechanism to communicate between the vehicles either through radio or wifi or indirect communication. One such example of indirect communication is vision-based communication, where the color of the landing pad can be changed from red to another color (e.g blue), depending on whether the target would like for the UAV to land on it or not. The rover acts as the master, and the UAVs as slaves during this phase. The rover, while following a desired path, changes the value of the landing $flag$, based on whether it is safe for the UAV to land or not. In this paper, we consider it is unsafe to land when the rover is carrying out aggressive maneuvers, and safe to land when the rover moves along an approximate straight line. This $flag$ is used by the UAV(s) to transition between hover and tracking mode and landing mode as shown in Figure 5a. During the hovering-tracking mode, the UAV stays behind the target and tracks it, while maintaining it's current altitude. In the landing mode, the UAV decends onto the target.

B. Cooperative Strategy For Multi-UAV Landing

We extend the single vehicle landing to multiple vehicle landing framework. The vehicles need to land one after the another. Once a UAV is selected for landing, the UGV broadcast the landing sequence by announcing the current vehicle to be landed. Not that as the landing is a difficult task, it does not always happen within a given time instant. Thus, the vehicles need to cooperate for landing sequentially.



(a) (b)

Fig. 5 (a) A combined model showing both the collaborative strategies mentioned above for UAV index k (b) The final two-level cooperation allowing two UAVs land on a target moving along a curved path

VII. Results

This section demonstrates the single vehicle landing on the moving UGV and also cooperative landing of two UAVs on the UGV one after another. Initially, we first demostrate single vehicle landing through examples followed by multiple vehicle landing. Thee vehicle use the hybrd automata model as shown in figure 5a. We validate our results in simulation using Gazebo-ROS. In the single UAV scenarios, the UAV is 8m behind the rover at a height of 10m. While in the multi UAV scenarios, the UAVs are 8m behind the rover

and 4m away on either side, at a height of 10m. Once a UAV land, we then provide a different goal to the landed UAV. After it takes-off, the next UAV is tasked to land.

A. Vision-only model

The Figure 6 shows the SDRE-based landing using vision in the loop for different target trajectories. Each of the these landings have been successful as the vehicle landing inside the landing pad. Fig 7 shows the errors in position along X and Z axes, and the error in velocity along the X axis betweeen the UAV and target, as the rover moves with a velocity of 1m/s along the X axis. We can observe from 7a that the error in X decreases rapidly to 0, with some peaks. These peaks arise due to the deceleration of the UAV, so that the velocity of the target can be matched during the time of landing. The error in Z decreases along with decrease in error along X, although not as rapidly, as seen from Fig.7b. This is because as we descend, the FOV from the camera decreases, this constrains the motion of the vehicle such that the descended trajectory along Z is smooth and slower than along X. For the simulations, the values for $Q(X)$ and $R(X)$ are

$$Q = \begin{bmatrix} 1+\frac{100e_x^2}{|e_z^2|}+\frac{50}{||e_x|-1|}+\frac{100}{|e_x|} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{20\dot{e_x}e_z}{0.01|e_x|} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1+\frac{100e_y^2}{|e_z^2|}+\frac{50}{||e_y|-1|}+\frac{100}{|e_y|} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{20\dot{e_y}e_z}{0.01|e_y|} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+\frac{30e_z}{\sqrt{0.01(e_x^2+e_y^2)}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{e_z} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{10}{e_z} \end{bmatrix} \tag{42}$$

and

$$R = \begin{bmatrix} 75000 & 0 & 0 & 0 \\ 0 & 75000 & 0 & 0 \\ 0 & 0 & 800 & 0 \\ 0 & 0 & 0 & 500 \end{bmatrix} \tag{43}$$

B. Coordination Strategies

We will first show the cooperative strategy for maneuvering target and then extend the approach for multiple UAVs.

1. Cooperative strategy for maneuvering targets

Fig.8 shows the results of the coordination strategy used to land the UAV on the target as it moves along different maneuvers. To test the controller on challenging conditions, we make the UAV land when the target moves along the shorter path of its trajectory. In Fig.8a, Fig.8c, and Fig.8d, the UAV is made to land when the rover moves perpendicular to the X axis, and track otherwise. In Fig.8b the UAV is asked to land as the rover moves along the side of length 3m, and track as the target moves along the 8m side. We observe that by making the UAV only track the target while holding its altitude instead of attempting to land makes it possible for the UAV to adapt to the changes in the rover's path more easily. This allows the UAV to land on paths with length as short as 2m.

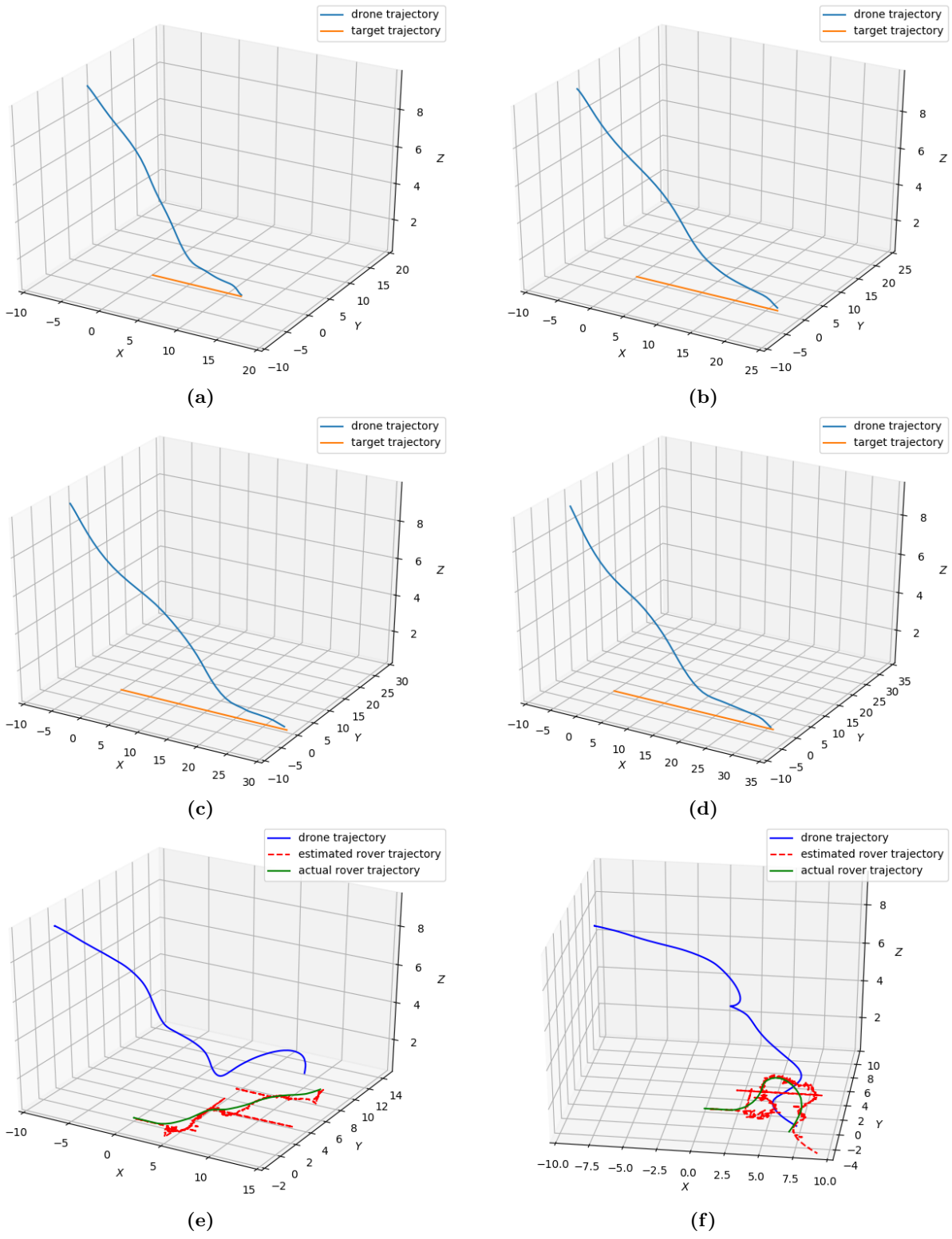
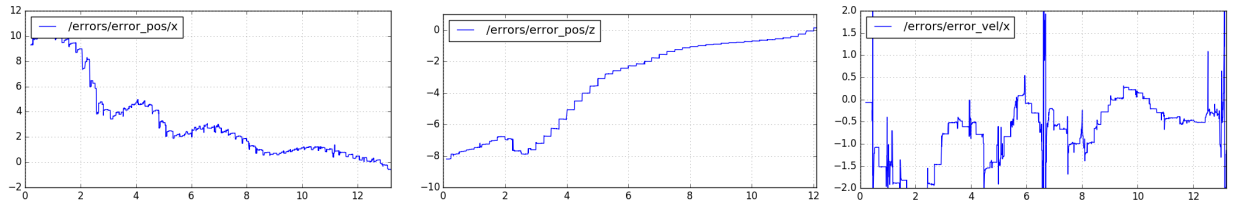
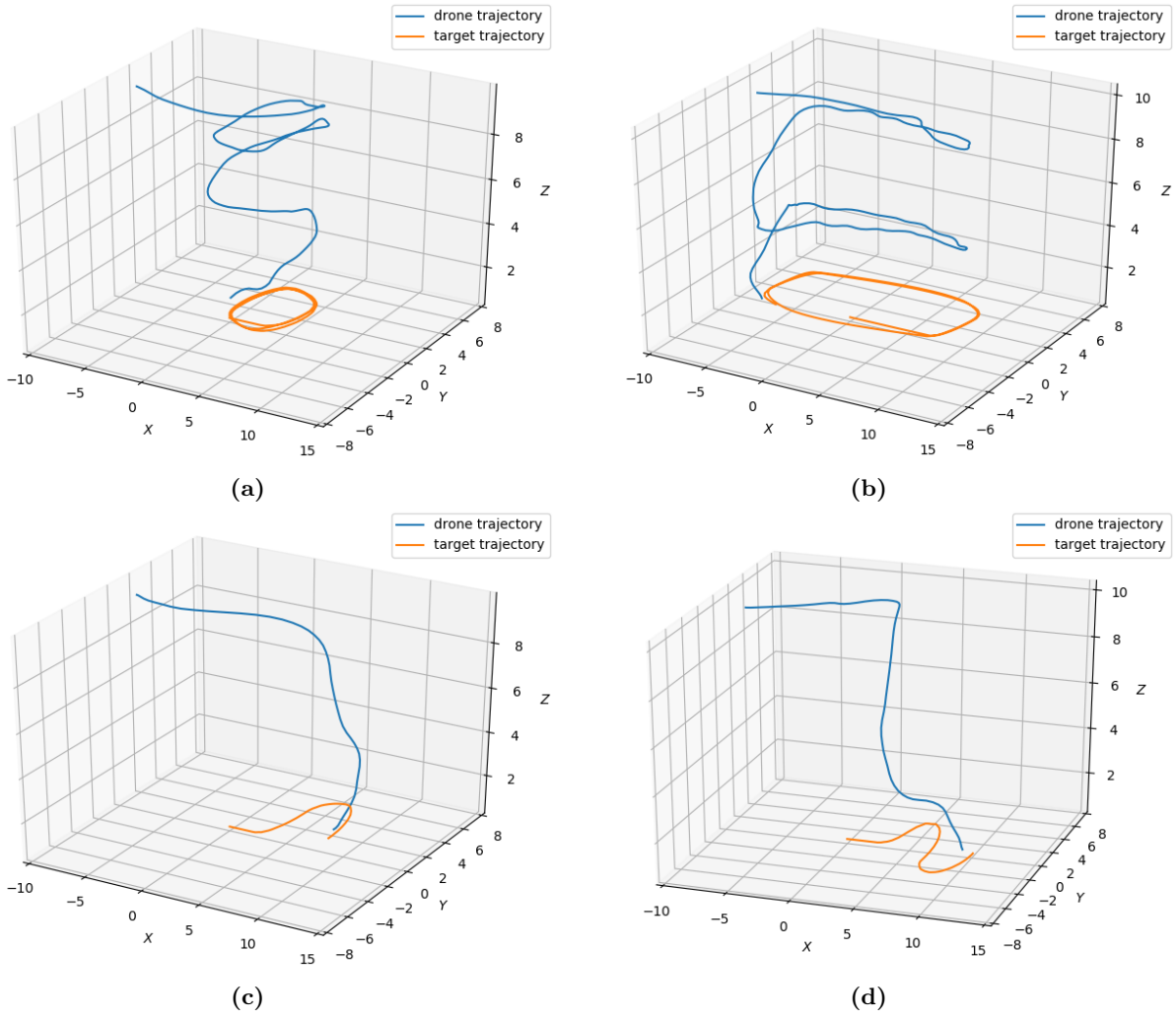


Fig. 6 Vision based landing of the UAV on the target moving along (a) a straight line with velocity 1m/s (b) a straight line with velocity 2m/s (c) a straight line with velocity 3m/s (d) a straight line with velocity 4m/s (e) a staircase like trajectory with step widht 3m and length 5m (f) a sine like curve with amplitude 5m



(a) (b) (c)

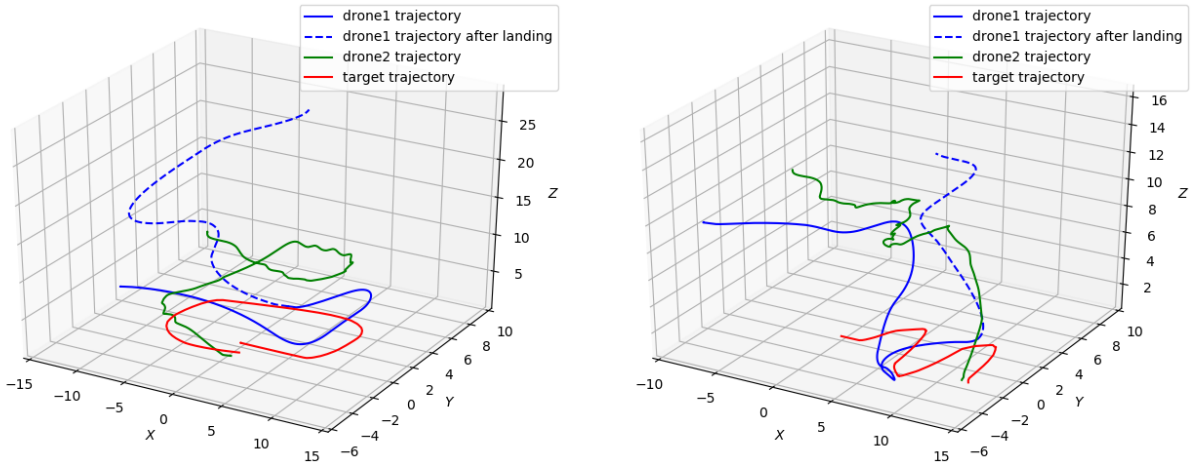
Fig. 7 Errors (a) in position along X direction (b) in position along Z direction (c) in velocity along X direction between the rover and drone while landing. The velocity of the rover is 1m/s.



(a) (b)

(c) (d)

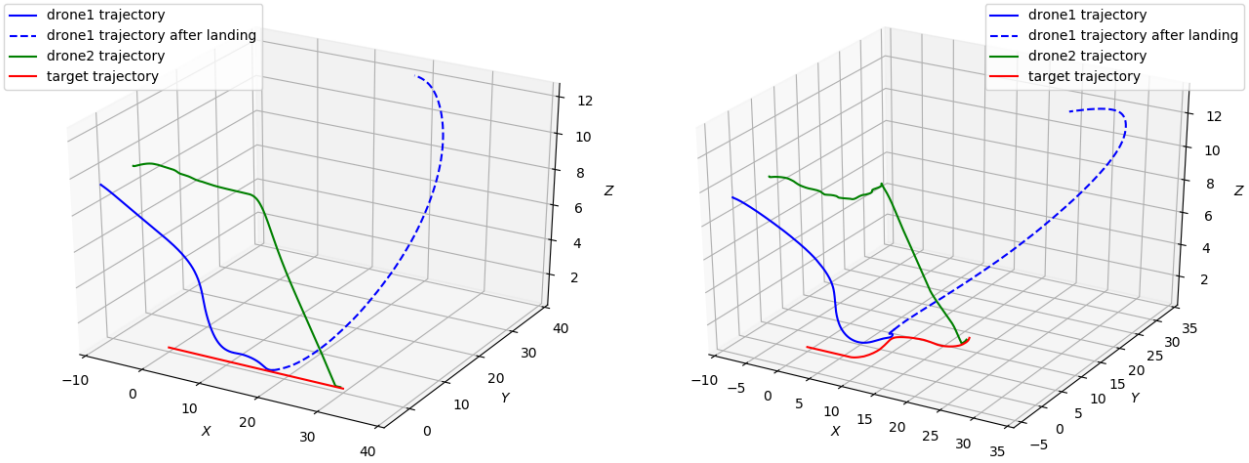
Fig. 8 Trajectories of UAV and target as the target moves along (a) a square of length 5m (b) a rectangle of length 8m and width 5m (c) a sine like curve with amplitude 5m (d) a sine like curve with amplitude 3m



(a) (b)

Fig. 10 Multiple agents landing on a target moving along (a) a rectangle of length 8m and with 5m (b) a sine like curve with amplitude 5m.

2. Multiple UAV Landing



(a) (b)

Fig. 9 Multiple agents landing on a target (a) moving in a straight line (b) moving along a step like trajectory with length 8m in either direction

Next, we show the cooperative strategy for landing multiple UAVs on the target as discussed in Sec.VI.B. Initially, drone1 is allowed to land as shown in Figure 9a. During the landing phase of drone1, the drone2 tracks the UGV. However, the can perform some other task until its lading time. After drone 1 lands and take off (shown in dotted line), the drone 2 is scheduled to land. Figure 9b shows the two vehicle landing for step trajectory of the UGV. In order to test the efficacy we tested the algorithm for highly maneuvrable environments in Figures 8b and 10b. From these simulations we can see that with coordination, the UAVs can safely land on the moving target.

VIII. Conclusions

This paper presents a SDRE-based controller for accurate autonomous landing of a UAV on to a moving UGV using vision. We propose cooperative strategies between the UGV and multiple UAVs for landing sequentially on to th UGV. The results shown that the proposed approach is effective in enabling the UAVs to maximally utilize the UGV. Future work involves experimentally validating the proposed approach and incorporating the UAV dynamics into the controller.

References

[1] Maini, P., Sundar, K., Singh, M., Rathinam, S., and Sujit, P., "Cooperative Aerial–Ground Vehicle Route Planning With Fuel Constraints for Coverage Applications," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 55, No. 6, 2019, pp. 3016–3028.

[2] Manyam, S. G., Casbeer, D. W., and Sundar, K., "Path planning for cooperative routing of air-ground vehicles," *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 4630–4635.

[3] Gautam, A., Sujit, P., and Saripalli, S., "Autonomous Quadrotor Landing Using Vision and Pursuit Guidance," *IFAC-PapersOnLine*, Vol. 50, No. 1, 2017, pp. 10501–10506.

[4] Serra, P., Cunha, R., Hamel, T., Cabecinhas, D., and Silvestre, C., "Landing of a quadrotor on a moving target using dynamic image-based visual servo control," *IEEE Transactions on Robotics*, Vol. 32, No. 6, 2016, pp. 1524–1535.

[5] Ghommam, J., and Saad, M., "Autonomous landing of a quadrotor on a moving platform," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 53, No. 3, 2017, pp. 1504–1519.

[6] Herissé, B., Hamel, T., Mahony, R., and Russotto, F. X., "Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow," *IEEE Transactions on Robotics*, Vol. 28, No. 1, 2012, pp. 77–89.

[7] Sereewattana, M., Ruchanurucks, M., Rakprayoon, P., Siddhichai, S., and Hasegawa, S., "Automatic landing for fixed-wing UAV using stereo vision with a single camera and an orientation sensor: A concept," *IEEE International Conference on Advanced Intelligent Mechatronics*, BEXCO, Busan, Korea, 2015, pp. 29–34.

[8] Sereewattana, M., Ruchanurucks, M., and Siddhichai, S., "Depth Estimation of Markers for UAV Automatic Landing Control Using Stereo Vision with a Single Camera," *International Conference on Information and Communication Technology for Embedded System*, Chennai, India, 2014.

[9] Jung, Y., Bang, H., and Lee, D., "Robust marker tracking algorithm for precise UAV vision-based autonomous landing," *International Conference on Control, Automation and Systems*, BEXCO, Busan, Korea, 2015, pp. 443–446.

[10] Saripalli, S., Montgomery, J. F., and Sukhatme, G. S., "Visually guided landing of an unmanned aerial vehicle," *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 3, 2003, pp. 371–380.

[11] Lange, S., Sunderhauf, N., and Protzel, P., "A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments," *IEEE International Conference on Advanced Robotics*, Munich, Germany, 2009, pp. 1–6.

[12] Abu-Jbara, K., Alheadary, W., Sundaramorthi, G., and Claudel, C., "A robust vision-based runway detection and tracking algorithm for automatic UAV landing," *International Conference on Unmanned Aircraft Systems*, Denver, Colorado, USA, 2015, pp. 1148–1157.

[13] Falanga, D., Zanchettin, A., Simovic, A., Delmerico, J., and Scaramuzza, D., "Vision-based autonomous quadrotor landing on a moving platform," *IEEE International Symposium on Safety, Security and Rescue Robotics*, Shanghai, China, 2017, pp. 11–13.

[14] Cocchioni, F., Frontoni, E., Ippoliti, G., Longhi, S., Mancini, A., and Zingaretti, P., "Visual based landing for an unmanned quadrotor," *Journal of Intelligent & Robotic Systems*, Vol. 84, No. 1-4, 2016, pp. 511–528.

[15] Li, P., Garratt, M., and Lambert, A., "Monocular Snapshot-based Sensing and Control of Hover, Takeoff, and Landing for a Low-cost Quadrotor," *Journal of Field Robotics*, Vol. 32, No. 7, 2015, pp. 984–1003.

[16] Vlantis, P., Marantos, P., Bechlioulis, C. P., and Kyriakopoulos, K. J., "Quadrotor landing on an inclined platform of a moving ground vehicle," *IEEE International Conference on Robotics and Automation (ICRA)*, Washington, USA, 2015, pp. 2202–2207.

[17] Hoang, T., Bayasgalan, E., Wang, Z., Tsechpenakis, G., and Panagou, D., "Vision-based target tracking and autonomous landing of a quadrotor on a ground vehicle," *American Control Conference (ACC), 2017*, Washington, USA, 2017, pp. 5580–5585.

[18] Wang, L., and Bai, X., "Quadrotor Autonomous Approaching and Landing on a Vessel Deck," *Journal of Intelligent & Robotic Systems*, Vol. 92, No. 1, 2018, pp. 125–143.

[19] Vlantis, P., Marantos, P., Bechlioulis, C. P., and Kyriakopoulos, K. J., "Quadrotor landing on an inclined platform of a moving ground vehicle," *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2202–2207. https://doi.org/10.1109/ICRA.2015.7139490.

[20] Feng, Y., Zhang, C., Baek, S., Rawashdeh, S., and Mohammadi, A., "Autonomous Landing of a UAV on a Moving Platform Using Model Predictive Control," *Drones*, Vol. 2, No. 4, 2018. https://doi.org/10.3390/drones2040034.

[21] Bouffard, P., Aswani, A., and Tomlin, C., "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results," *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 279–284.

[22] Shamma, J. S., and Cloutier, J. R., "Existence of SDRE stabilizing feedback," *IEEE Transactions on Automatic Control*, Vol. 48, No. 3, 2003, pp. 513–517.

[23] Çimen, T., "Survey of State-Dependent Riccati Equation in Nonlinear Optimal Feedback Control Synthesis," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 4, 2012, pp. 1025–1047. https://doi.org/10.2514/1.55821.

[24] Mathew, N., Smith, S. L., and Waslander, S. L., "Planning paths for package delivery in heterogeneous multirobot teams," *IEEE Transactions on Automation Science and Engineering*, Vol. 12, No. 4, 2015, pp. 1298–1308.

[25] Sujit, P., Kingston, D., and Beard, R., "Cooperative forest fire monitoring using multiple UAVs," *2007 46th IEEE Conference on Decision and Control*, IEEE, 2007, pp. 4875–4880.

[26] Rucco, A., Sujit, P., Aguiar, A. P., De Sousa, J. B., and Pereira, F. L., "Optimal rendezvous trajectory for unmanned aerial-ground vehicles," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 2, 2017, pp. 834–847.