Autonomous Quadrotor Landing on a Moving Platform in Outdoor Environments

Chris Choi*, Stanley Brown† and Steven L. Waslander‡

***Abstract*—**

I. Introduction

II. Related Work

III. Contribution

- Robust Landing
- AprilTag estimation with Kalman Filter
- AprilTag windowing to speed up detection
- Light Invariant AprilTag detection
- Tracking Controller
- Trajectory Planning

IV. System Overview

In this section we describe the setup of our problem, from the quadrotor, ground landing target to coordinate systems used to represent positions in body and inertial frames.

For the autonomous landing problem we assume that the quadrotor has direct line of sight to the landing target and that the AprilTag is detectable. The target position and velocity estimation is performed on-board the quadrotor, and using the estimations of the target the quadrotor tracks and lands onto the target.

- I: Inertial Frame
- C: Camera Frame
- G: Gimbal Frame
- B: Quadrotor Body Frame
- BP: Quadrotor Body Planar Frame

We describe five different main coordinate frames used throughout this paper, in general we have followed the ROS coordinate frame conventions where the inertial frame I is in ENU coordinate frame, the camera frame C is in EDN coordinate frame, gimbal frame G, quadrotor body frame B and quadrotor body planar frame BP uses NWU coordinate frame. The quadrotor body planar frame BP is defined as the body centered and inertial horizon x-y aligned frame, to reiterate the body planar frame BP ignores the roll and pitch (but not yaw) normally considered in a body frame B.

* M.A.Sc. Candidate, Mechanical and Mechatronics Engineering, University of Waterloo; c33choi@uwaterloo.ca

2 M.A.Sc. Candidate, Mechanical and Mechatronics Engineering, University of Waterloo; s52brown@uwaterloo.ca

† Assistant Professor, Mechanical and Mechatronics Engineering, University of Waterloo; stevenw@uwaterloo.ca

V. Target Tracking

A popular visual fiducial system called AprilTag [1] was chosen, the system allows us to identify the landing target along with the full 6 dof estimation. There exists multiple implementations of AprilTag, [2] was chosen because it is the most used implementation. In the following we describe steps taken to optimize the AprilTag detection in-order to run the processing at a higher rate.

A. Adaptive Image Processing

The standard implementation of the AprilTag library when processing images of 640 by 480 pixels results with an update rate of approximately 3 to 5 fps on the on-board computer too low for quadrotor controls. To improve performance we adapted the image size depending on the estimated distance between camera and AprilTag.

B. AprilTag Windowing

In [3] an attempt was made to optimize the AprilTag library by reducing the brightness of the image such that the majority of the image is black, we went one step further and masked the lastest image using the last detected AprilTag camera coordinates, rendering everything other than the AprilTag black, with this modification we assumed the image coordinates between frames did not vary too wildly, else the mask is removed and the full image is inputed to the AprilTag detector for processing.

Once the center of the AprilTag in camera frame is found the top left and bottom right corners of the AprilTag can be calculated with:

$$X_{\text{top left}} = X - (l/2) - X_{\text{padding}} \tag{1}$$
$$Y_{\text{top left}} = Y - (l/2) - Y_{\text{padding}} \tag{2}$$
$$X_{\text{bottom right}} = X + (l/2) + X_{\text{padding}} \tag{3}$$
$$Y_{\text{bottom right}} = Y + (l/2) + Y_{\text{padding}} \tag{4}$$

Where X, Y are positions of the AprilTag in the camera frame, $X_{\text{padding}}, Y_{\text{padding}}$ are the mask padding in the camera frame, l is the AprilTag side length in meters, and finally $X_{\text{top left}}, Y_{\text{top left}}, X_{\text{bottom right}}, Y_{\text{bottom right}}$ represent the top left and bottom right corners in the camera frame. Adding the mask padding at this point allows the padding to be defined in the camera frame which increases and decreases proportionally in the image frame depending on the depth-distance between camera and AprilTag.

Using the pin-hole camera model, we can convert the AprilTag's corners from camera frame into image frame with the following:

$$x = \frac{f_x * X}{Z} \tag{5}$$

$$y = \frac{f_y * Y}{Z} \tag{6}$$

Where f_x and f_y are the focal length in pixels in the x and y axis, and finally x, y are image pixel coordinates. Using the image coordinates of the top left and bottom right corners of the AprilTag a mask can be created.

C. Nested AprilTag

To address visibility issues of the AprilTag as the field of view of is reduced during landing, we found a unique combination of AprilTag ids that allows us to place a smaller AprilTag in the center of a larger AprilTag (see Fig **??**). By shear coincidence the AprilTag implementation [2] always prefers the smaller secondary tag when both are detected.

D. Illumination Invariant AprilTag

A common issue in computer vision is illumination changes in the environment which in turn interferes with the performance of detection algorithms. From our experience the standard black and white AprilTag fails to be detected if a shadow is casted partially or fully upon the tag features, this is because the AprilTag detection relies heavily on the edges and lines of the tag to be able to estimate the 6 dof pose. Depending on the time of day or weather conditions, this can have a significant impact as the quadrotor approaches the target, as its own shadow over the AprilTag can render the tag undetectable.

Reliable image processing is therefore paramount to the success of target detection, using methods described in [**?**] we have idenitified the best colours for an illumination invariant AprilTag to be blue and green (see Fig **??**), additionally

$$I = \log(R_2) - \alpha \log(R_1) - (1 - \alpha) \log(R_3) \tag{7}$$

Where R_1, R_2, R_3 are sensor responses (or image channels) corresponding to peak sensitivities at ordered wavelengths $\lambda_1 < \lambda_2 < \lambda_3$.

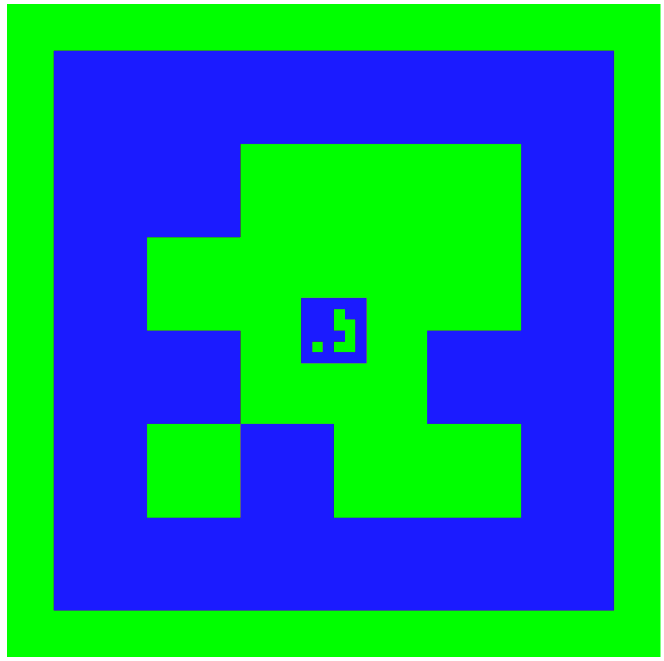

Fig. 1: Illumination Invariant AprilTag

VI. Target Estimation

The inertial target position, linear velocity, angular velocity and heading are estimated with an Extended Kalman Filter running at 200Hz, the estimation is in turn used for tracking and landing controllers on board the quadrotor.

A. Process Model

We chose to use a two-wheel robot motion model to approximate the forward kinematics of the target, which is given as:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{bmatrix} \\ &= \begin{bmatrix} x & y & z & \theta & v & \omega & v_z \end{bmatrix} \end{aligned} \tag{8}$$

$$\mathbf{u}(t) = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} v \\ \omega \\ v_z \end{bmatrix} = \begin{bmatrix} \mathcal{N}_v \\ \mathcal{N}_\omega \\ \mathcal{N}_{v_z} \end{bmatrix} \tag{9}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \\ \dot{x_7} \end{bmatrix} = \begin{bmatrix} x_5 \cos(x_4) \\ x_5 \sin(x_4) \\ x_7 \\ x_6 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{10}$$

Where the estimator estimates the inertial position in x, y and z direction, heading θ, wheel velocity v, steering angular velocity ω and linear velocity v_z in the z direction. It is worth noting the linear velocity in the x and y direction can be calculated from $v\cos(\theta)$ and $v\sin(\theta)$ respectively. The inputs to the process model are driven by Gaussian White Noise, since we do not possess input information to the two wheel robot motion model.

B. Measurement Model

The measurement inputted into the estimator is the target position (x, y, z) in the inertial frame, the target position has been transformed from the camera frame through to the gimbal frame to the inertial frame. Care was taken to ensure that the IMU measurements were synchronized with image of the AprilTag at the time it was captured.

To obtain the target position in the inertial frame, the detected target position is first transformed from image frame to gimbal joint frame as follows:

$$x_{\text{gjf}} = R_{\text{i}}^{\text{gjf}} R_{\text{cf}}^{\text{nwu}} x_{\text{i}} + t_c \tag{11}$$

Then from gimbal joint frame to body planar frame:

$$x_{\text{bpf}} = R_{\text{gjf}}^{\text{bpf}} x_{\text{gjf}} \tag{12}$$

Finally from body planar frame to inertial frame:

$$x_{\text{if}} = R_{\text{nwu}}^{\text{enu}} R_{\text{bpf}}^{\text{if}} x_{\text{bpf}} \tag{13}$$

This final target position in the inertial frame is then passed onto the estimator. The measurement model $h(t)$ and the linearized measurement model $H(t)$ at time t is then simply

$$h(t) = \begin{bmatrix} x, y, z \end{bmatrix}^T \tag{14}$$

$$H(t) = \begin{bmatrix} I_{3\times3} \end{bmatrix}^T \tag{15}$$

The Extended Kalman Filter updates the process model at 200 Hz, while the measurement is updated upon when the AprilTag is detected and transformed. The output of the estimator is in turn used for tracking and landing.

VII. Trajectory Planning

A. 2D Quadrotor Model

The model used for trajectory planning is a two dimensional three degrees of freedom (DOF) quadrotor in the x-z plane. By using this model we assume there will be small changes in the x-y plane where the controllers can correct without planning.

The 2D quadrotor state $\mathbf{x}$ includes both position and velocity in the x and z direction. Where x_0 and x_d are the initial and final states of the quadrotor. The inputs to the quadrotor model are the acceleration a_z in the z direction in body frame, and pitch angle θ in inertial frame.

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T \\ &= \begin{bmatrix} x & \dot{x} & z & \dot{z} \end{bmatrix}^T \in \mathbb{R}^4 \end{aligned} \tag{16}$$

$$\begin{aligned} \mathbf{x}_0 &= \begin{bmatrix} x_0 & \dot{x_0} & z_0 & \dot{z_0} \end{bmatrix}^T \\ \mathbf{x}_d &= \begin{bmatrix} x_d & \dot{x_d} & z_d & \dot{z_d} \end{bmatrix}^T \end{aligned} \tag{17}$$

$$\mathbf{u}(t) = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} a_z \\ \theta \end{bmatrix} \in \mathbb{R}^2 \tag{18}$$

The dynamics are thus given as:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} x_2 \\ u_1 \sin(u_2) \\ x_4 \\ u_1 \cos(u_2) - g \\ u_2 \end{bmatrix} \tag{19}$$

Where g is the gravitational constant.

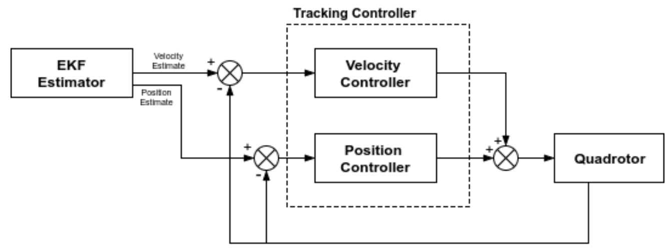

Fig. 2: Tracking Controller

B. Trajectory Problem Formulation

The objective of the trajectory is to achieve landing of the quadrotor onto a non-stationary target at $\mathbf{x}_d$, further the trajectory has to satisfy a number of desirable soft constraints, which are minimal deviation from desired path p_{diff}, minimal total input u_{total} and minimal input control difference between time steps u_{diff}.

The desired path $\mathbf{p}_{\text{desired}}$ vector is a straight-line from initial position x_0 to final position x_d.

$$p_{\text{diff}} = ||\mathbf{x} - \mathbf{p}_{\text{desired}}||^2 \tag{20}$$

$$u_{\text{total}} = ||\mathbf{u}||^2 \tag{21}$$

$$u_{\text{diff}} = \sum_{i=0}^{t_f-1} \|\mathbf{u}(i+1) - \mathbf{u}(i)\|^2 \tag{22}$$

$$J(\mathbf{x}, \mathbf{u}) = w_1 \cdot p_{\text{diff}} + w_2 \cdot u_{\text{total}} + w_3 \cdot u_{\text{diff}} \tag{23}$$

$$C_{\text{eq}} = \sum_{i=0}^{t_f-1} \mathbf{x}(i+1) - f(\mathbf{x}(i), \mathbf{u}(i)) \tag{24}$$

$$\begin{aligned} \text{minimize} \quad & J(\mathbf{x}, \mathbf{u}) && (25) \\ \text{s.t.} \quad & \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) && (26) \\ & \mathbf{x}(t=0) = \mathbf{x}_0 && (27) \\ & \mathbf{x}(t=t_f) = \mathbf{x}_d && (28) \\ & \mathbf{u} \in \mathbf{U} \quad \forall t \in [0, t_f] && (29) \end{aligned}$$

VIII. Quadrotor Control

A. Tracking Controller

For target tracking two PID controllers were implemented, first a position based controller that corrects positional errors of the quadrotor relative to the target. Second, a velocity based controller that uses estimated target velocity as a feed-forward signal to match the target's velocity. The two controllers are combined together to form the tracking controller.

$$\theta = k_p e + k_i \sum_{i=0}^{t} e_i dt + k_d \dot{e} \tag{30}$$

B. Landing Controller

During landing the controller will be switched from tracking to landing, the controller contains two PID controllers the first controller is a position based control to correct positional errors of the quadrotor relative to the target in the body frame.

IX. Experiment Results

X. Conclusions and Future Work

References

[1] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3400–3407, IEEE, 2011.

[2] "AprilTags C++ Library." `https://people.csail.mit.edu/kaess/apriltags/`. Accessed: 2016-01-12.

[3] K. Ling, U. of Waterloo. Department of Mechanical, and M. Engineering, *Precision Landing of a Quadrotor UAV on a Moving Target Using Low-cost Sensors*. University of Waterloo, 2014.