Simultaneous Tracking of Multiple Ground Targets from a Multirotor Unmanned Aerial Vehicle

Nathaniel Miller
Texas A&M University, College Station, Texas 77843

and

Jonathan Rogers
Georgia Institute of Technology, Atlanta, Georgia 30332

DOI: 10.2514/1.1010345

An algorithm for simultaneous tracking of multiple ground targets by an unmanned aerial vehicle is presented. The algorithm is specifically tailored toward multirotor vehicles, and it consists of a particle filter to predict target motion, a reference trajectory generator, and a finite-horizon model predictive controller for trajectory tracking. Two versions of the algorithm are proposed: for a vehicle equipped with a gimbaled camera, and for a vehicle equipped with a fixed camera. Furthermore, a target rejection algorithm is included to prune targets that inhibit accurate tracking of the majority of the target set. The tracking algorithm and multirotor vehicle dynamic model are first described, followed by example simulations for both the gimbaled and fixed-camera cases. Trade studies are presented, analyzing the effects of controller tuning parameters and camera field of view, as well as performance of the target rejection algorithm. In simulation experiments, real-world target data are used to improve simulation fidelity. Overall, results show that the algorithm is effective in capturing a set of ground targets within the field of view simultaneously when using either gimbaled or fixed-camera configurations, but performance is somewhat degraded in the fixed-camera case when target dynamics occur on timescales similar to tracking vehicle dynamics.

Nomenclature

\[ a_{lat}^i, a_{lon}^i = \text{lateral and longitudinal accelerations of the } i\text{th target} \]

\[ \hat{a}_{lat}^i, \hat{a}_{lon}^i = \text{estimated lateral and longitudinal accelerations of target } i \]

\[ c_d, c_r = \text{multirotor induced drag coefficients} \]

\[ F_{Dx}, F_{Dy}, F_{Dz} = \text{induced drag force components in the inertial frame} \]

\[ g = \text{gravitational acceleration} \]

\[ H = \text{prediction horizon} \]

\[ I_B, J_B, K_B = \text{unit vectors of the body-fixed reference frame} \]

\[ I_x, J_x, K_x = \text{unit vectors of the inertial reference frame} \]

\[ M_x, M_y, M_z = \text{body-frame components of the total external moment acting on the multirotor} \]

\[ M_C = \text{total external moment on the multirotor about its mass center} \]

\[ m = \text{mass of the multirotor vehicle} \]

\[ p, q, r = \text{body-frame components of the angular velocity of the body frame with respect to the inertial frame} \]

\[ Q, R = \text{model predictive control weighting matrices} \]

\[ r_i = \text{distance from } i\text{th target to centroid of other targets} \]

\[ S = \text{rotor disk area} \]

\[ T = \text{thrust force} \]

\[ u_x, u_y, u_z = \text{inertial-frame components of the velocity of the mass center with respect to the inertial frame} \]

\[ \bar{v}^{i}_b = \text{velocity of the multirotor mass center} \]

\[ u^i = \text{measured velocity of the } i\text{th target} \]

\[ x_{CG}, y_{CG}, z_{CG} = \text{multirotor mass center position in inertial frame} \]

\[ x_{cam}, y_{cam}, z_{cam} = \text{position of center of camera frame in inertial frame} \]

\[ x^*_0, y^*_0, z^*_0 = \text{desired multirotor position} \]

\[ x^*_0(0), y^*_0(0) = \text{measured position of target } i\text{ at beginning of planning horizon} \]

\[ w_x, w_y, w_z = \text{wind velocity components in inertial frame at the current multirotor position} \]

\[ w_1, w_2, w_3 = \text{weighting values for target rejection score} \]

\[ z_i = \text{center of discretized altitude interval} \]

\[ \rho = \text{air density} \]

\[ \sigma_{lat}^i, \sigma_{lon}^i = \text{estimated standard deviation of uncertainty in } \hat{a}_{lat}^i \text{ and } \hat{a}_{lon}^i \]

\[ \sigma_{v}^i = \text{estimated standard deviation of uncertainty in } u^i \]

Presented as Paper 2014-2670 at the AIAA Atmospheric Flight Mechanics Conference, Atlanta, GA, 16–20 June 2014; received 6 October 2014; revision received 19 February 2015; accepted for publication 20 February 2015; published online 17 March 2015. Copyright © 2015 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 2327-3097/15 and $10.00 in correspondence with the CCC.

*Graduate Research Assistant, Department of Aerospace Engineering.

†Assistant Professor, Woodruff School of Mechanical Engineering.


**I. Introduction**

There is increasing interest in the ability to task low-cost airborne assets to autonomously track ground targets: specifically through video feed. Such autonomous tracking capabilities may allow air vehicles to perform monitoring and surveillance activities without operator control, which can be especially burdensome in cases involving large numbers of targets and/or tracking vehicles. Consider a scenario in which multiple ground targets must be tracked and surveilled simultaneously by a fleet of unmanned air vehicles. Such a scenario may arise in urban combat settings, where a fleet of aircraft seeks to monitor enemy armored vehicles throughout a city. In an alternative scenario, a group of aircraft may wish to provide overhead surveillance for a convoy of ground vehicles. In these scenarios, one air vehicle may be responsible for simultaneously tracking multiple targets. This tracking problem primarily becomes one of vehicle guidance and path planning, since the vehicle flight path must permit onboard sensors to track multiple ground targets that may be uncooperative. As ground targets move farther apart, the aircraft must increase altitude to ensure the targets remain within sensor fields of view. In addition, the air vehicle must continuously search for the optimal position that minimizes the probability of targets being obscured or unexpectedly maneuvering out of the field of view. Thus, the vehicle path-planning problem and the tracking solution must be compatible with sensor characteristics, vehicle dynamic constraints, and runtime limitations. Few solutions currently exist for this type of multitarget tracking problem, and tactical unmanned aerial vehicles (UAVs) are often restricted to tracking and engaging one target at a time.

An extensive body of literature exists on automated airborne target tracking algorithms. Video-based tracking of a single target has been well studied [1,2] but such methods do not easily generalize to the simultaneous multitarget case. Similarly, the problem of tracking multiple targets from multiple aerial platforms has been studied somewhat extensively during the past decade (see, for instance, [3–7]). In many ways, these algorithms are generalizations of single-target single-tracker algorithms, in which the number of trackers needed is approximately the same as the number of targets. For high-altitude missions tasked with tracking multiple targets in a confined area, such as the predator scenario discussed in [8], the multitarget tracking problem from a single vehicle can be easily solved by flying standard surveillance patterns and determining optimal pointing solutions for onboard sensors. The problem becomes fundamentally different and far more difficult, however, when tracking is done with tactical low-altitude UAVs with onboard low-cost sensors of limited range and resolution. In this case, the vehicle must constantly climb, descend, and reposition to ensure adequate visual range and resolution with low-cost sensors. Limited work has also addressed the multiple target tracking problem when simultaneous tracking of all targets is not a priority [9,10]. Such solutions may fail if targets move unpredictably while not under observation

This paper presents a guidance and control algorithm for tracking an arbitrary number of ground targets from a single multirotor. For the purposes of algorithm development, it is assumed that target vehicle positions are known, and the control system is tasked to keep all targets within the video field of view to the maximum extent possible given dynamic constraints and uncertainty in target motion. Another critical assumption underlying the proposed work is that the tasks of pointing the sensor and determining target motion from the sensor feed are assumed to be completed by external algorithms separate from the controller derived here. Both of these topics have been well studied [11–14]. Given these assumptions, the algorithm described here focuses on vehicle guidance and control for the multirotor aircraft. The tracking algorithm is constructed by first generating a reference trajectory given predicted target motion. Based on the reference trajectory, a receding-horizon optimal control problem is formulated and a linear model predictive control algorithm is generated assuming linearized vehicle dynamics. A particle filter is used to predict target motion over the receding control horizon. In addition, a target rejection algorithm limits the tracking vehicle altitude below a prescribed ceiling, ensuring that tracking resolution of the total target set is not sacrificed in order to track a handful of rogue targets.

The paper proceeds as follows. First, the multitarget tracking problem is formulated and a linear dynamic model for the multirotor vehicle is described. The proposed guidance algorithm is outlined including the reference trajectory generator, model predictive controller, and target motion filter. A method of target rejection is also proposed in which a rejection score for each target is calculated as a weighted sum of various metrics. Following a brief discussion of system identification for the simulation model, example simulation results are presented using experimentally obtained data for walking, biking, and driving targets. Simulation results are provided for both gimbaled and fixed-camera cases. Finally, trade studies are performed examining the effects of camera field of view, target rejection, and gimbaled vs fixed-camera performance. Overall, results demonstrate that the algorithm enables simultaneous multitarget tracking performance for a broad range of ground target types and is robust to uncooperative target motion.

**II. Tracking Algorithm**

Figure 1 depicts a general multiagent, multitarget tracking problem formulated by considering \( M \) agents and \( N \) targets. Classical solutions to this problem typically involve two stages: a so-called asset assignment stage, in which targets are assigned to (or selected by) agents for tracking;
and the tracking stage, in which agents actually carry out the tracking task. The asset assignment problem has been well studied, and numerous methods now exist to perform target assignment in a decentralized manner, including market-based approaches \cite{15}, distributed optimization \cite{16}, and game theory \cite{17}, among others. In this paper, it is assumed that target assignment has already taken place (or continues to take place externally), leaving the individual vehicle to track its specific set of targets. Thus, the focus of this paper is the control algorithm onboard each agent that allows it to track multiple targets simultaneously. An implicit assumption invoked here is that asset assignment and target tracking can be completely decoupled. In practice, some coupling between these two layers may be advantageous, since targets that are difficult to track by one agent may be transferred to another, improving overall tracking of the target set. Such coupling considerations are not explored in this work.

The tracking algorithm proposed here consists of four components. First, a particle filter estimates target motion over a finite horizon using a simple constant-acceleration kinematic model. Given these predicted target paths, a trajectory generator constructs a desired three-dimensional reference trajectory for tracking by the airborne agent. A finite-horizon linear model predictive control (MPC) scheme is used to track this trajectory. Finally, a target rejection algorithm determines if any targets are driving the tracking vehicle to altitudes in excess of a prescribed ceiling or reducing tracking performance for the majority of targets. Such rogue targets are removed from the target set through a mathematical rejection criterion. A diagram of this control architecture is shown in Fig. 2. Detailed descriptions of each algorithm component are provided in this section following a discussion of the multirotor vehicle dynamic model.

A. Nonlinear Multirotor Dynamic Model

The multirotor dynamic model considered in this paper is a five-degree-of-freedom model consisting of three translational degrees of freedom and two rotational degrees of freedom. Due to the inherent symmetry of the multirotor vehicle, yaw is neglected in the equations of motion. Figure 3 shows a diagram of the reference frames used in development of the dynamic model, namely, the inertial frame $I$ and the body-fixed frame $B$. The inertial frame makes use of a flat-Earth approximation, and the $B$ frame is centered at the multirotor mass center. The $B$ frame is obtained from the $I$ frame by rotating first about $J_I$ by the pitch angle $\theta$, and then, about $I_B$ by the roll angle $\phi$. This geometry is depicted in Fig. 3.

Three forces are assumed to act on the multirotor: weight $-mgK_I$, thrust $T K_B$, and drag. Let the velocity of the mass center with respect to the $I$ frame be given by $v_{0/I} = v_x I_x + v_y I_y + v_z I_z$, and let the angular velocity of the $B$ frame with respect to $I$ be given by $\omega_{B/I} = p I_B + q J_B + r K_B$. Drag acts at the mass center and is given by $F_D = F_{Dx} I_x + F_{Dy} I_y + F_{Dz} I_z$. Furthermore, assume that control is available through pure torques generated by differential thrust, leading to moments $M_C = M_x J_B + M_y J_B + M_z K_B$ about the mass center. Therefore, the nonlinear equations of motion for the multirotor are given by

\begin{equation}
\begin{bmatrix}
\dot{x}_{CG} \\
\dot{y}_{CG} \\
\dot{z}_{CG}
\end{bmatrix}
= \begin{bmatrix}
v_x \\
v_y \\
v_z
\end{bmatrix}
\end{equation}

\begin{equation}
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta}
\end{bmatrix}
= \begin{bmatrix}
p + \sin(\phi) \tan(\theta) q + \cos(\phi) \tan(\theta) r \\
\cos(\phi) q - \sin(\phi) r
\end{bmatrix}
\end{equation}

![Fig. 2 Multitarget tracking guidance and control architecture.](image)

![Fig. 3 Reference frame definitions for multirotor dynamic model.](image)
\[
\begin{aligned}
\begin{bmatrix}
\dot{v}_x \\
\dot{v}_y \\
\dot{v}_z 
\end{bmatrix} &= \frac{1}{m} \begin{bmatrix}
T \cos(\phi) \sin(\theta) + F_{Dx} \\
-T \sin(\phi) + F_{Dy} \\
T \cos(\phi) \cos(\theta) + F_{Dz} - mg 
\end{bmatrix} \\
\dot{p} \\
\dot{q} \\
\dot{r} &= [I_B]^{-1} \begin{bmatrix}
M_x \\
M_y \\
M_z 
\end{bmatrix} \begin{bmatrix}
0 & -r & q \\
-\theta & 0 & -p \\
-\theta & -q & 0 
\end{bmatrix} [I_B]^{-1} \begin{bmatrix}
p \\
q \\
r 
\end{bmatrix}
\end{aligned}
\] 

where \( I_B \) is the moment of inertia matrix of the vehicle about its mass center expressed in the \( B \) frame.

For multirotor vehicles, the primary contribution to drag in forward flight is caused by the thrust imbalance on the advancing rotors, giving rise to a phenomenon known as induced drag [18]. Induced drag is typically modeled as a linear function of translational velocity components in the body frame. Define \( v^v_x \equiv v_x - v_w \), with analogous definitions for \( v^v_y \) and \( v^v_z \). Following [18], induced drag is given by

\[
\begin{bmatrix}
F_{Dx} \\
F_{Dy} \\
F_{Dz} 
\end{bmatrix} = -T \begin{bmatrix}
c_{dl}(c_d^2 v^v_x - s_d s_b v^v_y) + c_{dl}(s_d^2 c_b^2 v^v_y + s_d s_b c_b v^v_z) + s_d^2 s_b c_b v^v_z \\
c_{dl}(s_d s_b c_b^2 v^v_y + c_b^2 v^v_z) + c_{dl}(s_d^2 s_b^2 v^v_y + s_d s_b c_b v^v_z) + c_{dl}(s_d^2 s_b^2 v^v_y + s_d s_b c_b v^v_z) + s_d^2 s_b c_b v^v_z \\
c_{dl}(-s_d s_b c_b v^v_x + c_b^2 v^v_z) + c_{dl}(s_d^2 s_b^2 v^v_x + s_d s_b c_b v^v_z) + s_d^2 s_b c_b v^v_z 
\end{bmatrix}
\]

where \( c_{dl} \) and \( c_{dr} \) are the induced drag coefficients in the \( I_x \) and \( I_y \) directions, and \( s_b \equiv \sin(\alpha_b), c_b \equiv \cos(\alpha_b) \). The trigonometric quantities in Eq. (5) arise from the fact that \( v^v_x, v^v_y, v^v_z \) and \( F_{Dx}, F_{Dy}, F_{Dz} \) are defined as components in the inertial frame for compatibility with the model in Eq. (3). Although other aerodynamic effects act on the vehicle (i.e., parasite drag on the vehicle body, etc.), these components are usually neglected in multirotor modeling, as they are typically dominated by high-gain control action [18]. Note that several authors have demonstrated high-performance control of quadrotor vehicles using a simple static thrust model without accounting for any sources of drag [19,20].

For the case of a fixed (nongimbled) camera, one additional quantity of interest for the tracking control law is the location of the center of the camera field of view projected in the ground plane. As shown in Fig. 3, this location is given by the coordinates \( (x_{\text{cam}}, y_{\text{cam}}) \), such that

\[
x_{\text{cam}} = x_{\text{CG}} - z_{\text{CG}} \tan(\theta)
\]

\[
y_{\text{cam}} = y_{\text{CG}} + z_{\text{CG}} \tan(\phi)/\cos(\theta)
\]

The next section linearizes this model about a given vehicle state, allowing for synthesis of a linear model predictive controller for multiple target tracking.

**B. Linearized Multirotor Dynamic Model**

To obtain linearized equations of motion for the multirotor, first define nondimensionalized thrust as \( \hat{T} \equiv T/mg \). Then, Eq. (3) becomes

\[
\begin{bmatrix}
\dot{v}_x \\
\dot{v}_y \\
\dot{v}_z 
\end{bmatrix} = \begin{bmatrix}
g \hat{T} \cos(\phi) \sin(\theta) + \frac{F_{Dx}}{m} \\
-g \hat{T} \sin(\phi) + \frac{F_{Dy}}{m} \\
g \hat{T} \cos(\phi) \cos(\theta) + \frac{F_{Dz}}{m} - g 
\end{bmatrix}
\]

Several assumptions are invoked during the linearization process as follows:

1. The vehicle is equipped with an inner-loop stabilization system capable of tracking commanded Euler angles with a first-order lag. Thus, \( \dot{\phi} = r_\theta (\phi_k - \phi) \) and \( \dot{\theta} = r_\theta (\theta_k - \theta) \), where \( r_\phi \) and \( r_\theta \) are the time constants of the stabilization controller and the plant.
2. Vehicle thrust \( \hat{T} \) responds to commanded thrust \( \hat{T}_c \) according to a first-order lag such that \( \hat{T} = \tau_r (\hat{T}_c - \hat{T}) \), where \( \tau_r \) is a time constant accounting for motor controller and rotor inflow dynamics.
3. The vehicle operates at relatively small pitch and roll angles. Thus, it is assumed that \( \sin(\theta) \approx \theta, \cos(\phi) \approx 1 \) and \( \tan(\theta) \approx \theta \) with similar expressions for \( \phi \). Likewise, terms involving products of Euler angles are assumed to be negligible.

Define the vehicle state vector and control vector as

\[
x = \begin{bmatrix}
x_{\text{CG}} & y_{\text{CG}} & z_{\text{CG}} & \phi & \theta & v_x & v_y & v_z & \phi_c & \theta_c & \hat{T}_c 
\end{bmatrix}^T
\]

\[
u = \begin{bmatrix}
\phi_c & \theta_c & \hat{T}_c 
\end{bmatrix}^T
\]

In Eqs. (9) and (10), two control inputs \( \phi_c \) and \( \theta_c \) are included as elements of the state vector rather than the control vector. The reason for this, as shown in Sec. II.E, is that both the control and control rate can be penalized separately within the model predictive controller.

Given the aforementioned assumptions, Eq. (8) can be linearized about a hover flight condition given by \( \hat{T}_0 = 1 \) and \( \phi_0 = \theta_0 = 0 \), where hover means that the mass center velocity with respect to the wind is zero. Let \( \Delta \hat{T}, \Delta \phi, \Delta \hat{T}_c, \Delta v_x, \Delta v_y, \) and \( \Delta v_z \) represent small perturbations about this flight condition. This leads to the linearized translational dynamics given by

\[
\begin{bmatrix}
\Delta \hat{v}_x \\
\Delta \hat{v}_y \\
\Delta \hat{v}_z 
\end{bmatrix} = \begin{bmatrix}
g \Delta \theta - g c_{dl} \Delta v_x \\
-g \Delta \phi - g c_{dl} \Delta v_y \\
g \Delta \hat{T}
\end{bmatrix}
\]
For the remainder of the paper, the Δ symbols are dropped for convenience. Defining $\tilde{c}_{dx} \equiv gc_{dx}$ and $\tilde{c}_{dy} \equiv gc_{dy}$, Eq. (11) becomes

$$\begin{pmatrix}
\dot{v}_x \\
\dot{v}_y \\
\dot{v}_z
\end{pmatrix} = \begin{pmatrix}
g\theta - \tilde{c}_{dx}v_x \\
g\phi - \tilde{c}_{dy}v_y \\
gT
\end{pmatrix}$$

(12)

Given Eqs. (1) and (12), and the assumptions outlined previously regarding closed-loop dynamics, the following linear dynamic model can be established:

$$\dot{x} = Ax + Bu$$

(13)

where

$$A = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & -\tau\phi & 0 & 0 & 0 & 0 & \tau\phi & 0 \\
0 & 0 & 0 & 0 & -\tau\theta & 0 & 0 & 0 & 0 & \tau\theta \\
0 & 0 & 0 & g & -\tilde{c}_{dx} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -g & 0 & 0 & -\tilde{c}_{dy} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\tau_T
\end{bmatrix}$$

(14)

$$B = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tau_T
\end{bmatrix}^T$$

(15)

Although the state vector dynamics in Eq. (13) are the same for both the gimbaled and fixed-camera cases, two separate output vectors are defined for each of these vehicle configurations. These output vectors are given by

$$y_G = [x_{CG} \quad y_{CG} \quad z_{CG} \quad \phi \quad \theta]^T$$

(16)

$$y_{NG} = [x_{CG} \quad y_{CG} \quad z_{CG} \quad \phi \quad \theta \quad x_{cam} \quad y_{cam}]^T$$

(17)

where $y_G$ and $y_{NG}$ represent the gimbaled and fixed-camera cases, respectively. To linearize the output equations for $x_{cam}$ and $y_{cam}$ in Eqs. (6) and (7), a small angle approximation is invoked. Furthermore, a class of linear models is created for 10 logarithmically spaced altitude intervals centered at altitudes $z_i$, where $i = 1, \ldots, 10$. The linear output equations for the gimbaled and fixed-camera configurations are given, respectively, by

$$y_G = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} x = C_G x$$

(18)

$$y_{NG} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -z_i & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & z_i & 0 & 0 & 0 & 0 & 0
\end{bmatrix} x = C_{NG}^i x$$

(19)

where $z_i$ is the center of the discretized altitude band.

Collectively, Eqs. (13), (18), and (19) represent the linearized multirotor model for both the gimbaled and fixed-camera configurations. These continuous equations are transformed to a discrete-time system using a zero-order-hold transform [21] with a time step of 0.2 s. The resulting discrete-time equations of motion are given in Eqs. (20–22):

$$x_{k+1} = A_d x_k + B_d u_k$$

(20)
\[ y_{G,k} = C_G x_k \]  
\[ y_{NG,k} = C_{NG} x_k \]

### C. Ground Target Model

The proposed guidance algorithm requires predictions of target motion over a finite time horizon \( H \). Targets are modeled as point masses and, for the purposes of this paper, their motions are restricted to the \( I_x - I_y \) plane, as shown in Fig. 4. A constant acceleration model is assumed, where targets hold constant longitudinal and lateral accelerations over the planning horizon. It is further assumed that lateral acceleration is reasonably used. Let \( _i \) the mean and standard deviation of \( 6i \) for the purposes of this paper, their motions are restricted to the four preceding quantities, perturbed dynamic model given by

\[ \dot{x}_i(t) = (a_{l0} + v_0^i) \sin \psi_i(t) \]

(23)

\[ \dot{y}_i(t) = (a_{l0} + v_0^i) \cos \psi_i(t) \]

(24)

The heading angle \( \psi_i(t) \) is given by

\[ \psi_i(t) = \psi_i(0) + \int_0^t \frac{a_{l0} \, \mathrm{d}t}{v_0^i + a_{l0}^2} = \left\{ \begin{array}{ll} \psi_i(0) - \frac{a_{l0}^2 \ln v_0^i}{a_{l0}^2} + \frac{a_{l0}^2 \ln(v_0^i + a_{l0}^2) - a_{l0}^2}{a_{l0}^2} & \text{if } a_{l0}^2 \neq 0 \\ \psi_i(0) + \frac{a_{l0}^2}{v_0^i} & \text{if } a_{l0}^2 \approx 0 \end{array} \right. \]

(25)

For the purposes of this paper, it is assumed that measurements of the target’s heading and speed are available over a prior time horizon. Given these measurements, estimates of \( a_{l0}^i, a_{l0}^i, v_0^i \), and \( \psi_i(0) \) may be obtained using online accumulation algorithms such as those given in [22] (the accumulator equations are omitted here for brevity). Furthermore, assuming the uncertainty associated with each estimate is Gaussian, the standard deviation associated with each estimated quantity is also available through these accumulators. Thus, at the start of the planning horizon, the mean and standard deviation of \( a_{l0}^i, a_{l0}^i, v_0^i \), and \( \psi_i(0) \) for each target are assumed to be available. Let these values be denoted by \( a_{l0}^i, a_{l0}^i, v_0^i, \psi_i(0) \) and \( \sigma_{a_{l0}}^i, \sigma_{a_{l0}}^i, \sigma_{v_0}^i, \sigma_{\psi_0}^i \), respectively.

A particle filter is implemented for each target to predict target motion over the planning horizon. Using the mean and standard deviation for the four preceding quantities, \( N_p \) particle trajectories are generated over the interval \( t = [0, H] \) for each target in a Monte Carlo fashion using a perturbed dynamic model given by

\[ \dot{x}_i(t) = (a_{l0}^i(t) + v_0^i) \sin \psi_i(t) \]

(26)

\[ \dot{y}_i(t) = (a_{l0}^i(t) + v_0^i) \cos \psi_i(t) \]

(27)

\[ \psi_i(t) = \tilde{\psi}_i(0) + \int_0^t \frac{a_{l0}^i \, \mathrm{d}t}{v_0^i + a_{l0}^i} = \left\{ \begin{array}{ll} \tilde{\psi}_i(0) - \frac{a_{l0}^i^2 \ln v_0^i}{a_{l0}^i^2} + \frac{a_{l0}^i^2 \ln(v_0^i + a_{l0}^i^2) - a_{l0}^i^2}{a_{l0}^i^2} & \text{if } a_{l0}^i \neq 0 \\ \tilde{\psi}_i(0) + \frac{a_{l0}^i}{v_0^i} & \text{if } a_{l0}^i \approx 0 \end{array} \right. \]

(28)

for \( j = 1, \ldots, N_p \), where the \( j \)th perturbed acceleration, velocity, and heading values are sampled according to

\[ a_{l0}^i \sim N(a_{l0}^i, \sigma_{a_{l0}}^i), \quad v_0^i \sim N(v_0^i, \sigma_{v_0}^i), \quad \tilde{\psi}_i(0) \sim N(\tilde{\psi}_i(0), \sigma_{\tilde{\psi}_0}^i) \]

(29)

![Fig. 4 Ground target motion diagram.](image-url)
An example of these Monte Carlo trajectories for four targets is shown in Fig. 5 in the form of black trajectory paths emanating from each target at the current instant in time. In Fig. 5, individual lines show the observed paths of the targets, and multiple lines emanating from the current target locations are possible target paths predicted by the particle filter. Circles represent the expected location of each target at a future time. A bounding rectangle is drawn that contains the expected locations of all targets at time step \( k \). The desired multirotor location is the center of this square camera field of view. At the beginning of each planning time step, particles are completely regenerated from measurements rather than resampled from prior trajectories; thus,

\[
\begin{align*}
x_i^*(0) &= x_i^0(0) \\
y_i^*(0) &= y_i^0(0)
\end{align*}
\]

(30)

(31)

where \( x_i^0(0) \) and \( y_i^0(0) \) are the measured position of target \( i \) at the beginning of the planning horizon. Note that this scheme is equivalent to a particle filter in which the measurement error covariance is zero. However, the preceding algorithm requires significantly less computational effort than a complete particle filter, since resampling and likelihood computation is not performed in light of the assumption of perfect measurements. It should also be noted that the target prediction algorithm used here is provided strictly to evaluate performance of the proposed tracking algorithm. More complex target motion predictors can certainly be used and are unlikely to affect the performance of the tracking scheme.

D. Reference Trajectory Generation

Let the prediction horizon be divided into \( K \) equally spaced time intervals such that \( t_{k+K} - t_k = H \). The goal of the reference trajectory generator is to create a sequence of system outputs such that the predicted expected values of all target locations remain within the camera field of view over the prediction horizon. It is further desired that the multirotor maintain the minimum altitude possible to keep all targets’ expected locations within the field of view. Let \((x_{ik}, y_{ik})\) be the \( i \)th target’s expected location at time step \( t_k \). Furthermore, let the surface of the ground be given in general form by \( z = f(x, y) \) and the set of all points on the ground be given by set \( \mathbb{C} = \{ (x, y, z) \in \mathbb{R}^3 | z = f(x, y) \} \). Note that \( f \) may be either a smooth function in the case of terrain or a discontinuous function in the case of buildings, etc. Define the line-of-sight vector between target \( i \) and the multirotor commanded location as \( \mathbf{L}_i = (x_{ik} - x_i)\mathbf{J}_i + (y_{ik} - y_i)\mathbf{K}_i + (z_{ik} - z_i)\mathbf{J}_i \). The intersection of this line-of-sight vector with the ground surface is given by the set \( S_i = \mathbf{L}_i \cap \mathbb{C} \), which contains at least one point (the \( i \)th target location) but can potentially contain more in the case of targets obscured by buildings or terrain features. Finally, define all points that lie within the projection of the camera field of view on the ground as \( \mathbb{C}_F \subset \mathbb{C} \). Then the commanded multirotor location at time step \( k \) can be found by solving the following constrained nonlinear optimization problem:

Find \( (\tilde{x}_k, \tilde{y}_k, \tilde{z}_k) \) that minimizes \( J = |\tilde{z}_k| \)

subject to \( \tilde{z}_k > f(\tilde{x}_k, \tilde{y}_k) \),

\[
\mathbb{S}_i \setminus (x_{ik}, y_{ik}, z_{ik}) = \emptyset \quad \text{for all} \ i
\]

\[
(x_{ik}, y_{ik}, z_{ik}) \subset \mathbb{C}_F \quad \text{for all} \ i
\]

The solution of the optimization problem framed as shown ensures that the multirotor will track all targets from a position at minimum altitude such that line-of-sight vectors do not pass through obstacles (second constraint) and all targets remain within the camera field of view (third constraint). For general terrain, a nonlinear optimization solver may be used to determine the commanded multirotor location at each time step.
For the purposes of this paper, flat terrain is assumed, and thus the line-of-sight vector constraint is satisfied for any choice of \((\hat{s}_k, \hat{y}_k, \hat{z}_k)\). To solve this optimization problem for two-dimensional (2-D) terrain, bounding targets are defined at the edges of the camera field of view and the optimal multirotor location \((\hat{s}_k, \hat{y}_k, \hat{z}_k)\) is computed as that which contains the expected location of all targets at minimum altitude. Figure 5 presents a graphical representation of this process at an example time step. Note that, for simplicity, it is assumed that the camera field of view is equal in the \(I_g\) and \(J_R\) directions, and that the vehicle yaw angle is always zero. Also, it is assumed that the vehicle pitch and roll angles are small enough that the skewing of the field of view is neglected.

Optionally, a buffer zone can be designated at the edge of the camera field of view to keep targets from the approaching the edge, guarding against modeling errors and target prediction uncertainty. In this case, bounding targets are measured against the edge of the buffer zone rather than the edge of the field of view. Use of this buffer zone has the effect of increasing the multirotor altitude, which is generally undesirable. A trade study involving this buffer distance is described in the Simulation Results section (Sec. IV).

Let \((\hat{s}_k, \hat{y}_k, \hat{z}_k)\) be the desired multirotor position and \(\hat{\theta}_k\) and \(\hat{\phi}_k\) be the desired roll and pitch angles, respectively, at time step \(k\). The following stacked output vectors are defined over the prediction horizon for the gimbaled and fixed-camera cases, respectively:

\[
y^G_k = [\hat{x}_k \ \hat{y}_k \ \hat{z}_k \ \hat{\phi}_k \ \hat{\theta}_k \ \ldots \ \hat{x}_{k+K} \ \hat{y}_{k+K} \ \hat{z}_{k+K} \ \hat{\phi}_{k+K} \ \hat{\theta}_{k+K}]^T
\]

\[
y^{NG}_k = [\hat{x}_k \ \hat{y}_k \ \hat{z}_k \ \hat{\phi}_k \ \hat{\theta}_k \ \ldots \ \hat{x}_{k} \ \hat{y}_{k} \ \hat{z}_{k} \ \hat{\phi}_{k} \ \hat{\theta}_{k} \ \ldots \ \hat{x}_{k+K} \ \hat{y}_{k+K} \ \hat{z}_{k+K} \ \hat{\phi}_{k+K} \ \hat{\theta}_{k+K}]^T
\]

Throughout the rest of this work, \(\hat{\phi}_k\) and \(\hat{\theta}_k\) are set to zero, which has the effect of forcing the model predictive controller to track the desired velocity with as close to a level attitude as possible. Furthermore, in Eq. (33), \(\hat{x}_{\text{cam}, k} = \hat{x}_k\) and \(\hat{y}_{\text{cam}, k} = \hat{y}_k\) for all \(k\), since, for the fixed-camera case, it is desired that the multirotor be positioned directly above the target set as much as possible.

### E. Model Predictive Control

A linear model predictive controller is constructed to track the reference trajectory. Given the control vector [Eq. (10)] at time step \(k\), a stacked sequence of control vectors may be defined as

\[
U = [u^T_k \ u^T_{k+1} \ \ldots \ u^T_{k+K}]^T
\]

For this control sequence, the estimated output \(Y\) over the planning horizon may be computed from the discrete system model using block matrices according to the following equations [23]:

\[
y = k_{\text{cam}} x_k + k_{\text{cab}} U
\]

\[
k_{\text{cab}} = \begin{bmatrix} C_d A_d & 0 & 0 & \ldots & 0 \\ C_d A_d^2 & C_d B_d & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ C_d A_d^{K-1} B_d & \ldots & C_d A_d^{K-2} B_d & \ldots & C_d A_d B_d & C_d B_d \\
\end{bmatrix}
\]

\[
k_{\text{cam}} = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ C_d A_d & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots \\ C_d A_d^{K-2} & \ldots & C_d A_d B_d & 0 \\
\end{bmatrix}
\]

Given the predicted output vector \(Y\) and the desired output vector \(Y_k\) from Eqs. (32) or (33), a quadratic cost function is constructed to penalize both state error and controls:

\[
J = (Y - Y_k)^T Q (Y - Y_k) + U^T R U
\]

where \(Q\) and \(R\) are appropriate positive definite weighting matrices. Setting the derivative of Eq. (38) with respect to \(U\) equal to zero and solving for \(U\) results in the optimal control over the planning horizon given by

\[
U = k (Y_k - k_{\text{cam}} x_k)
\]

where

\[
k = (k_{\text{cab}}^T Q k_{\text{cab}} + R)^{-1} k_{\text{cab}}^T Q
\]

Because the MPC is recalculated at each time step, only the first \(u_1\) term in \(U\) is needed. Extracting only the top row \(k_{\text{min}}\) from Eq. (40) allows a more computationally efficient calculation of the optimal \(u_k\) given by
\( u_k = k_{\text{min}} (y_R - k_{\text{sc}} x_k) \)  

(41)

For the gimbaled case, \( k_{\text{min}} \) and \( k_{\text{sc}} \) are constants and can be precomputed before flight. For the fixed-camera case, recall from Eq. (19) that the output matrix is a function of the discretized altitude range. Thus, in implementation of Eq. (41), a set of 10 different \( k_{\text{min}} \) and \( k_{\text{sc}} \) matrices are computed, corresponding to each discretized altitude range, as described in Sec. II.B. When running the MPC algorithm, the appropriate feedback matrix is selected based on the current multirotor altitude, using the implicit assumption that altitude does not change significantly over the planning horizon. This scheme is equivalent to a gain-scheduling approach based on altitude. Finally, note that the \( Q \) and \( R \) matrices can be adjusted to achieve desired performance of the MPC controller for a given vehicle. For instance, if it is desired that thrust variations be relatively slow at the expense of altitude tracking performance, the altitude weight in the \( Q \) matrix can be adjusted appropriately.

F. Gimbal Angle Computation

For the gimbaled camera case, define the gimbal frame \( G \) such that the center of the camera field of view is aligned with \( K_G \). The \( G \) frame is obtained from the \( B \) frame by first rotating about the \( J_B \) axis by gimbal angle \( \theta \), and then about the \( I_G \) axis by gimbal angle \( \phi \). The location \((x_{\text{cam}}, y_{\text{cam}})\) of the center of the camera field of view is therefore given by

\[
x_{\text{cam}} = x_{CG} - z_{CG} \frac{-s_\phi s_y s_\theta + c_\theta s_\phi c_y + c_\phi c_\theta s_\phi c_y}{-s_\phi s_y s_\theta + s_\phi c_\theta c_y + c_\phi c_\theta s_\phi c_y}
\]

(42)

\[
y_{\text{cam}} = y_{CG} + z_{CG} \frac{s_\phi c_\theta + c_\phi s_\theta}{-s_\phi s_y s_\theta + s_\phi c_\theta c_y + c_\phi c_\theta s_\phi c_y}
\]

(43)

Let \( \Delta x_{\text{cam}} \) and \( \Delta y_{\text{cam}} \) represent the difference between the desired \((x_{\text{cam}}, y_{\text{cam}})\) location and that achieved from zero gimbal angles, as shown in Eqs. (6) and (7). Then, under a small angle approximation for both the gimbal angles and vehicle pitch and roll angles, Eqs. (42) and (43) may be manipulated to arrive at the desired gimbal angles given by

\[
\theta = \frac{\Delta x_{\text{cam}}}{z_{CG}}
\]

(44)

\[
\phi = \frac{\Delta y_{\text{cam}}}{z_{CG}}
\]

(45)

G. Target Rejection Algorithm

In many practical instances, a maximum altitude ceiling may be defined that limits the multirotor’s ability to track a target set that exhibits completely arbitrary, uncorrelated motion. In such instances, an automated target rejection algorithm is needed to remove targets from the tracking set that are forcing the multirotor to exceed its altitude ceiling. The target rejection algorithm proposed here includes three components. First, for each target \( i \), the distance from the target centroid is computed according to

\[
r_i^2 = \left( x_i - \frac{1}{N} \sum_{j=1}^{N} x_j \right)^2 + \left( y_i - \frac{1}{N} \sum_{j=1}^{N} y_j \right)^2
\]

(46)

In addition, the course over ground correlation of target \( i \) with the average course over ground is computed according to

\[
\psi_{icor}^i = \psi_i \frac{1}{N} \sum_{j=1}^{N} \psi_j
\]

(47)

Finally, the third component \( P_i \) counts the number of times target \( i \) has been responsible for limiting the bounding rectangle, as shown in Fig. 5; i.e., the number of times the target is at the edge of the camera field of view. Throughout the tracking session, rejection scores for each target are computed as a weighted sum according to

\[
R_i = w_1 \frac{r_i^2}{\max r_j^2} + w_2 \frac{\max |\psi_{icor}^i|}{\max_i |\psi_{icor}^i|} + w_3 \frac{P_i}{\max_j P_j}
\]

(48)

where \( w_1, w_2, \) and \( w_3 \) are weighting values. Note that all terms in Eq. (48) are normalized by the greatest value over the target set, since units are different between each term, making weight selection easier. If the multirotor has been within 10% of its user-specified altitude ceiling consecutively for 2 s, the target with the greatest score is rejected (i.e., removed from the tracking set).

Note that this rejection algorithm exhibits some important characteristics in the context of the multitarget tracking framework in order to maintain general compatibility with an asset assignment algorithm. First, if motion of the target set is highly uncorrelated, then as the multirotor reaches its altitude ceiling, additional targets will be rejected, potentially leaving only a single target within the tracking set. This is valid behavior since, as long as the altitude ceiling is enforced, it is impossible to track this disjoint target set from a single vehicle. Likewise, the algorithm has no mechanism to add targets back to the tracking set once they reappear in the field of view. Both of these considerations have implications for the asset assignment algorithm, which is responsible for assigning targets to the tracking set. In the uncorrelated motion case, this would signify that
the asset assignment algorithm has done a poor job in target selection for this tracking vehicle, and continual pruning of the set could be an indication to the assignment algorithm that the tracking set should be reconstituted. Likewise, once a target has been rejected, it would presumably be assigned to another vehicle, and thus it is not the responsibility of the low-level tracking algorithm to reintroduce it to the target set.

Computational burden for the overall algorithm described in this section is judged to be reasonable for low-cost embedded real-time systems. Each component of the algorithm [namely, target prediction in Eqs. (26–28), MPC in Eq. (41), trajectory generation, and target rejection in Eq. (48)] either scales linearly with or is independent of the number of targets. The exception is trajectory generation in complex [three-dimensional (3-D)] terrain that requires solution of the constrained nonlinear optimization problem shown in Sec. II.D. In the case of complex terrain where targets may be obscured, a high-throughput coprocessor may potentially be used to efficiently solve for a reference trajectory at a reasonable update rate.

III. Data Collection and System Identification

Simulation experiments for this work are based on target data obtained through experimental measurements. Furthermore, flight data were obtained using a 3-D Robotics Hexacopter B multirotor, and system identification was performed to enhance simulation fidelity.

A. Target Data Collection

Target data for simulation experiments were collected by placing a GPS receiver on three different ground vehicles: a pedestrian on foot, a bicyclist, and an automobile. Several resulting trajectories were recorded. These data were postprocessed to determine time histories of $a_{\text{in}}, a_{\text{iw}}, v_0$, and $\psi(0)$. The data were then interpolated onto a uniform time grid using a time step of 0.2 s.

---

![Example randomized target trajectories.](image)

Fig. 6 Example randomized target trajectories.

![Simulation and observed responses to roll and pitch doublets for multirotor system identification.](image)

Fig. 7 Simulation and observed responses to roll and pitch doublets for multirotor system identification.
Table 1  Identified multirotor model parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_\phi, \tau_\theta )</td>
<td>5.0</td>
</tr>
<tr>
<td>( \tau_T )</td>
<td>2.0</td>
</tr>
<tr>
<td>( \tilde{e}<em>{\phi}, \tilde{e}</em>{\theta} ) (s(^{-1}))</td>
<td>0.045</td>
</tr>
<tr>
<td>( \tilde{\tau} )</td>
<td>[0.2, 2.0]</td>
</tr>
</tbody>
</table>

Fig. 8  \( x\)-\( y \) location for example simulation with walking targets. Desired and actual multirotor paths are overlaid.

Fig. 9  Altitude vs time for example simulation with walking targets (gimbaled and nongimbaled cases overlaid).

Fig. 10  Horizontal (left) and vertical (right) velocity vs time for example simulation with walking targets (gimbaled and nongimbaled cases overlaid).
Walking data were collected on sidewalks in a small residential area. The pace under motion was about 1–2 m/s. Biking took place in a low-traffic district along a somewhat randomized route. Due to strong winds, riding speed varied from 5 m/s when traveling against the wind to 9 m/s with the wind. Driving was performed in a large residential area. Roads were selected that had clear views of the sky and speed limits under 18 m/s. This was done to ensure the car did not drive faster than the multicopter was capable of flying. Other than selecting roads with appropriate speed limits, no effort was made to drive out of the ordinary while collecting data.

For a given simulation experiment, a dataset is selected at random. Small perturbations are applied in heading, velocity, start time, and start location. The tracking algorithm is then tasked with tracking five sets of randomly perturbed data. A sample dataset generated from experimentally

---

**Fig. 11** Trajectory data for example simulation with walking targets: gimbaled case.

**Fig. 12** Trajectory data for example simulation with walking targets: nongimbaled case.
obtained walking trajectories is shown in Fig. 6, where targets begin near the origin and travel in a northeasterly direction. Note that open circles in Fig. 6 indicate synchronized time points spaced 60 s apart.

IV. System Identification

System identification was performed using the example multirotor with an active inner-loop tracking controller. The purpose behind these system identification experiments was to determine the first-order lag parameters $\tau_\phi$, $\tau_\theta$, and $\tau_T$, as well as drag coefficient values $\tilde{c}_d$ and $\tilde{c}_\phi$, in Eq. (14). To determine the angular rate response, roll and pitch doublets were applied, as shown in Fig. 7. Model parameters $\tau_\phi$ and $\tau_\theta$ were fit such that the simulated time response approximately overlaid the observed response. Likewise, axial flight maneuvers were performed to identify the $\tau_T$ thrust.

Fig. 13 $x$-$y$ location for example simulation with biking targets. Desired and actual multirotor paths are overlaid.

Fig. 14 Altitude vs time for example simulation with biking targets (gimbaled and nongimbaled cases overlaid).

Fig. 15 Horizontal (left) and vertical (right) velocity vs time for example simulation with biking targets (gimbaled and nongimbaled cases overlaid).
response parameter. Drag coefficients were determined by establishing steady-state forward flight at a constant pitch and roll angle, and equating the drag force with the horizontal components of thrust. Finally, axial flight sequences were used to determine maximum and minimum thrust values, which were used to limit desired thrust \( \tilde{T}_r \), output from the model predictive controller. The final identified model parameters are shown in Table 1.

V. Simulation Results

Example simulations are performed to demonstrate system performance in nominal conditions for walking, biking, and driving targets. Furthermore, the effects of various system parameters, such as camera field of view and the camera buffer region, are analyzed through trade studies. Finally, Monte Carlo simulations involving a rogue target (for which the motion is significantly different from other targets) are provided to demonstrate the difference in performance with and without the target rejection algorithm activated.
A. Example Simulations

Three example simulations are provided for the hexacopter vehicle described in the previous section. In all example cases, the camera field of view is 70 deg, zero buffer area is used around the edges of the camera frame, zero winds are assumed, and the hexacopter starts from rest. Figures 8–12 show an example simulation involving walking targets for both gimbaled and nongimbaled vehicles. In Fig. 8 (where the targets begin on the right of the figure), it is clear that the hexacopter tracks the reference trajectory fairly well in both the gimbaled and nongimbaled cases. The altitude time history in Fig. 9 shows that the hexacopter climbs almost steadily throughout the trajectory as the southernmost target moves away from the others. Figure 10 shows velocity time histories, where it is evident that the hexacopter maintains a translational velocity equal to a walking pace (1–2 m/s) over most of the flight. The reasonably high vertical velocities exhibited in Fig. 10 are due to the relatively high-altitude tracking gains used in the MPC for these scenarios. These gains may be relaxed to reduce the overall magnitudes of vertical velocity. Figure 11 demonstrates that, except for an initial transient as the hexacopter accelerates from rest, gimbal angles are small and tracking performance is excellent, with all targets captured throughout the majority of flight. In comparison, Fig. 12 shows that the nongimbaled case has a slightly higher tracking error compared to the gimbaled case. This highlights the inherent control tradeoff faced by the nongimbaled multirotor: tracking error can be improved through increased pitch and roll control action, but pitch and roll deviations tend to perturb the camera field of view significantly so that targets cannot be captured. In this case, target motion is relatively slow compared to the multirotor dynamics, so tracking performance is largely unaffected by this tradeoff and positioning error is small. Finally, note that pitch and roll angles of the multirotor are small for both the gimbaled and nongimbaled cases, which is a trend that is evident throughout the remainder of this section.

An additional example demonstrates performance for biking targets, as shown in Figs. 13–17. Figures 13 and 14 show x-y locations of the vehicles and the multirotor altitude time history for the biking example, respectively. Note that, in this scenario, the ground tracks of each target are reasonably correlated, allowing the multirotor to maintain a relatively constant altitude between 150 and 250 m. Figure 15 shows horizontal and vertical velocity time histories, which demonstrate similar trends as in the previous example. Additional trajectory plots for the biking case, shown in Figs. 16 and 17, again demonstrate that pitch and roll angles for the vehicle are fairly small, except for initial transients as the multirotor accelerates from rest. Again, tracking performance is excellent, with all targets remaining within the field of view for almost the entire flight.

Fig. 18 \textit{x-y} location for example simulation with driving targets. Desired and actual multirotor paths are overlaid.

Fig. 19 Altitude vs Time for example simulation with driving targets (gimbaled and nongimbaled cases overlaid.)
A final example demonstrates performance using driving targets, as shown in Figs. 18–22. Figure 18 shows $x$-$y$ locations of the targets and the multirotor, Fig. 19 shows an altitude time history, and Fig. 20 shows velocity time histories. In this case, the targets start relatively close together (in the lower-right corner of Fig. 18) and slowly spread apart, leading to the slowly increasing altitude profile in Fig. 19. Figures 21 and 22 show that the multirotor pitch and roll angles are noticeably higher for these driving cases, in which target dynamics are faster and the hexacopter requires more aggressive maneuvers. As noted in the walking and biking cases, the tracking error is small overall but generally larger for the nongimbaled case, which is penalized for maneuvers that place targets outside the field of view. Even so, although all targets are visible, the majority of the time, one target leaves the camera field of view (FOV) intermittently in the nongimbaled case, as shown in Fig. 19. This case, as well as the trade studies outlined in the next section, highlights the fact that, as target dynamic rates approach those of the multirotor, tracking performance generally suffers. This is especially true of the nongimbaled configuration, where a tradeoff exists between large-attitude angles needed for accurate trajectory tracking and low-attitude angles needed for target tracking.

B. Trade Studies

Trade studies are performed to demonstrate the effects of several controller parameters and features. The first trade study examines the effect of a camera frame buffer on tracking performance. The camera frame buffer is used to determine the desired multirotor altitude, given the desired
Fig. 22 Trajectory data for example simulation with driving targets: nongimbaled case.

Fig. 23 Camera frame buffer trade study results.
location. For instance, if the bounding box is drawn to encompass all targets, as shown in Fig. 5, a 10% buffer zone means the footprint of the camera field of view must overlay the bounding box on all sides by 10%. If the buffer zone is zero, the desired camera field of view footprint is the same as the bounding box. The purpose of this buffer zone is to account for uncertainty in target location prediction and vehicle tracking performance by providing the guidance algorithm with a margin of error.

A Monte Carlo simulation of 100 runs each is performed for various buffer zone values from 0 to 75%. In each run, five targets are used, with random biases applied to each in heading, velocity, start time, and start location. For each tracking session, three metrics are recorded. The first is the target out-of-frame percentage, which measures the percentage of time that the target set is outside the camera frame. For instance, if one of the five targets is out of frame for half the tracking session, the out-of-frame percentage would be 10%. If all targets are out of frame for the entire simulation, the out-of-frame percentage is 100%. The second metric used is normalized flight time, which is a measure of the energy required to execute a given tracking session divided by a reference energy value. For instance, for a vehicle with 10 min endurance while hovering, if a tracking session were conducted that exhausts battery life in 8 min, this would result in a normalized flight time of 80%. In calculation of this metric, it is assumed that power consumption is directly proportional to thrust. The final metric is mean altitude for a given tracking session.

For varying values of camera frame buffer, Monte Carlo simulations are performed, with the mean value of each of these three metrics shown in Fig. 23. Target rejection is turned off, and camera field of view is 70 deg. Several trends are evident in these data. First, for walking and biking targets, tracking performance is generally very favorable, even with camera buffer zones of 10% or less. For the nongimbaled case with driving targets, the system generally requires a buffer zone of 15% or greater to account for significant uncertainty in the target dynamics and restricted angular motion of the tracking vehicle due to the lack of a gimbal. Normalized flight time generally decreases as target dynamics become faster and as the camera buffer increases in size. These effects are due to the larger thrust that is generally required to track faster targets with a larger camera buffer (which reduces the usable field of view). As expected, the mean altitude increases as the frame buffer increases, again due to the reduction in the usable area of the camera field of view. Note that performance is universally better in the gimbaled cases compared to the nongimbaled cases, as expected. The exception is normalized flight time, where nongimbaled cases outperform gimbaled cases due to less aggressive maneuvering as the vehicle trades positioning error for minimal angular motion.

A second trade study is performed, examining the effects of camera field of view on tracking performance. In this case, walking targets are used and the camera buffer zone is set to zero. Figure 24 shows the results of this study. Note that favorable tracking performance is achieved at very small fields of view; however, the vehicle is forced to fly at a high altitude, which may compromise the ability to resolve targets effectively. As field of view is increased, the mean altitude decreases approximately exponentially, and normalized flight time increases slightly. In general, nongimbaled performance is observed to be only slightly worse than gimbaled performance.

A final trade study characterizes the effect of the target rejection portion of the algorithm. For these studies, six targets are used, with one target for which the motion is substantially different from the others. An example of walking target trajectories with one "rogue" target is shown in Fig. 25. Note that this rogue target is generated by adding rather large heading, velocity, start time, and start location perturbations to one of the recorded trajectory sequences. For this study, the hexacopter's maximum altitude is set to 140 m for walking cases, 380 m for biking cases, and 400 m for driving cases. The camera field of view is set to 70 deg and a zero camera frame buffer is used. During each simulation, the target
rejection feature is activated with all weights in Eq. (48) set to one. Weights were determined by varying each weight independently and observing the out-of-frame percentage for a set of randomized simulations. It was found that unity weights for each term produced the minimum out-of-frame percentage on average and that small changes in weights did not affect performance significantly. Note that, in the following simulations, once a target is rejected, it is counted as outside the camera frame for the rest of the tracking session.

Monte Carlo simulations of 100 runs each were performed for walking, biking, and driving targets, and the out-of-frame percentage and mean altitude were averaged over each run. The results are shown in Table 2. Several trends are apparent: most notably the fact that use of the target rejection algorithm universally improves tracking performance for the entire target set. Without allowing the algorithm to reject targets, the out-of-frame percentage is over 70% for all configurations. By allowing the algorithm to discard the rogue target from the set, performance improves by approximately 40%. Note that, if the rogue target was eliminated immediately and all other targets were perfectly tracked, the theoretically best out-of-frame percentage is 16.7%. However, since identification of the rogue target cannot be achieved immediately, tracking performance at the beginning of the session is generally poor, leading to out-of-frame percentages around 30–40%. Further, note that rejection of the rogue target allows the target set to be tracked from a much lower altitude, which is an additional benefit. Overall, for both the gimbaled and nongimbaled configurations, target rejection is shown to be an important element of the multitarget tracking algorithm, in that overall tracking of the target set can be accomplished by autonomously pruning outlier targets.

### VI. Conclusions

A guidance algorithm for simultaneous tracking of multiple targets by a rotary-winged vehicle is presented. The algorithm includes a generalized reference trajectory generator based on target motion predictions over a finite horizon capable of handling terrain obstacles and other obstructions (although results in this work are limited to 2-D terrain). A linear model predictive control scheme, based on a linearized multrotor dynamic model, is used to track reference trajectory commands. In addition, a target rejection scheme is proposed that allows the algorithm to improve overall tracking performance by eliminating specific targets that drive the tracking vehicle away from the core target set. Example simulation results with experimentally derived walking, biking, and driving data show that the algorithm exhibits favorable tracking performance. Vertical velocities in example simulations are shown to be high for a typical multrotor, but these can be reduced by adjustment of the model predictive control gains to allow for slower altitude tracking performance. Trade studies quantify the effects of the camera field of view and buffer zone size during trajectory generation. Additional Monte Carlo studies show that target rejection almost universally improves overall tracking performance in the case of highly uncorrelated target motion. Overall, the multitarget tracking scheme is shown to be effective across a wide range of targets displaying varying maneuver capabilities.

### References


J. How
Associate Editor