SIMULTANEOUS TRACKING OF MULTIPLE GROUND TARGETS FROM A SINGLE MULTIROTOR UAV

A Thesis

by

NATHANIEL RUSSELL MILLER

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Jon Rogers
Co-Chair of Committee,	Raktim Bhattacharya
Committee Member,	Sivakumar Rathinam
Head of Department,	Rodney Bowersox

May 2014

Major Subject: Aerospace Engineering

ABSTRACT

An algorithm for autonomous multirotor tracking of an arbitrary number of ground targets is formulated and evaluated through simulation. The algorithm consists of a particle filter to predict target motion, trajectory generator, and model predictive controller operating over a finite time horizon. Furthermore, a target selection algorithm is included to reject targets that preclude accurate tracking of the main target set. Performance of the guidance algorithm is evaluated through simulation using real-world target data. Simulation results show that targets can be kept within the camera's field of view when using either a gimbaled or non-gimbaled camera, but performance may be substantially degraded in the non-gimbaled case when target dynamics occur on time scales similar to tracking vehicle dynamics.

TABLE OF CONTENTS

Page

ABSTRACT ii

TABLE OF CONTENTS iii

LIST OF FIGURES iv

LIST OF TABLES v

NOMENCLATURE vi

1. INTRODUCTION 1

2. TRACKING ALGORITHM 4

2.1. Ground Target Model and Particle Filter 4
2.2 Trajectory Generation 6
2.3 Multirotor Model 8
2.4 Linear Model Predictive Controller 11
2.5 Target Rejection 13

3. SIMULATION 15

3.1 Target Data 15
3.2 System Identification 16
3.3 Example Trajectories 18
3.4 Trade Studies 20
3.4.1 Camera Frame Buffer Zone 21
3.4.2 Target Rejection 22
3.4.3 Camera Field of View 26

4. CONCLUSIONS 27

REFERENCES 29

LIST OF FIGURES

Page

Figure 2.1: Target coordinate system and notation 4

Figure 2.2: Determining multirotor position.. 8

Figure 2.3: Control system block diagram. 14

Figure 3.1: Example of randomized target trajectories. 16

Figure 3.2: Application of doublets and system response. 18

Figure 3.3: Example trajectory from gimbaled tracking session. 19

Figure 3.4: Example trajectory from non-gimbaled tracking session 20

Figure 3.5: Results from camera frame buffer trade study. 23

Figure 3.6: Example of target trajectories with one dissimilar target 24

Figure 3.7: Results from field of view trade study. 25

LIST OF TABLES

Page

Table 3.1: Experimentally determined system parameters. 17

Table 3.2: Target rejection results for both gimbaled and non-gimbaled case. 24

NOMENCLATURE

v	Ground speed.
ψ	Course over ground.
a_{lon}	Longitudinal acceleration.
a_{lat}	Lateral acceleration.
x	Distance from origin in easterly direction.
x_{cam}	Center of camera frame x location.
y	Distance from origin in northerly direction.
y_{cam}	Center of camera frame y location.
z	Altitude above ground.
$z_{nearest}$	Nearest altitude range (for gain scheduling).
ϕ	Roll angle.
θ	Pitch angle.
T	Non-dimensionalized thrust.
m	Mass.
ϕ_c	Commanded roll angle.
θ_c	Commanded pitch angle.
T_c	Non-dimensionalized commanded thrust.
τ	Time constant.
c_d	Coefficient of drag.
X	Measurement of random variable.

$\overline{\mathrm{X}}$	Expected value of random variable X.
σ_{X}	Standard deviation of random variable X.
n	Number of samples currently held in accumulator.
N	Maximum number of samples to be held in accumulator.
M	Intermediate variable used to facilitate accumulator calculations.
$\overline{t}$	Time vector.
Δt	Time horizon timestep.
H	Horizon length.
k	Index representing current point in time.
K	Number of discrete points in time horizon vector.
$\overline{x}$	State vector.
$\overline{u}$	Control vector.
$\overline{Y}$	Output vector.
$\left[\overline{Y}\right]$	Estimated output vector over time horizon.
$\tilde{Y}$	Desired output vector at an instant in time.
$\left[\tilde{Y}\right]$	Desired output vector over time horizon.
$[\overline{u}]$	Optimal control vector over time horizon.
$\tilde{x}$	Desired x position.
$\tilde{y}$	Desired y position.
$\tilde{z}$	Desired z position.
A	Continuous state influence matrix.

B	Continuous control influence matrix.
C	Continuous output matrix.
A_d	Discrete state influence matrix.
B_d	Discrete control influence matrix.
C_d	Discrete output matrix.
J	Cost function.
k_{ca}	Matrix used to facilitate notation.
k_{cab}	Matrix used to facilitate notation.
$[k]$	Gain matrix.
$[k_{\min}]$	Minimal gain matrix producing $\bar{u}_{k+1}$ only.
Q	State weight matrix (positive, semi-definite, diagonal).
R	Control weight matrix (positive, semi-definite, diagonal). .
r_j	Distance from target j to centroid of other targets.

1. INTRODUCTION

There is increasing interest in the ability to task low-cost airborne assets to autonomously track ground targets, specifically though video feed. Such autonomous tracking capabilities may allow air vehicles to perform monitoring and surveillance activities without persistent oversight by an operator, which may be a prohibitively burdensome task in cases involving large numbers of targets and/or tracking vehicles. In scenarios where the number of tracking vehicles is limited and the number of targets is large, one air vehicle may be responsible for simultaneously tracking multiple targets. In this case, the tracking problem primarily becomes one of vehicle guidance and path planning since the vehicle flight path must use only on-board sensors to optimally track multiple ground targets that are likely uncooperative. As ground targets move farther apart, the air vehicle must increase altitude to ensure the targets remain within sensor fields-of-view. Furthermore, the air vehicle must continuously search for the optimal position that minimizes the probability of targets being obscured. Thus, the vehicle path planning problem and the tracking solution are coupled and create a complex guidance problem that must be solved in real-time. No solution currently exists for this type of problem and therefore tactical UAV's 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 multi-target case. Similarly, the problem of tracking and engaging multiple targets from multiple aerial platforms has

been studied somewhat extensively during the past decade (see, for instance, [3-6]). In many ways these are simple generalizations of single-target-single-tracker algorithms, in which the number of trackers needed is the same as the number of targets. For high-altitude missions tasked with tracking targets in a confined area, such as the Predator scenario discussed in [7], the multi-target tracking problem from a single vehicle can be easily solved by flying standard surveillance patterns and determining optimal pointing solutions for on-board sensors. The problem becomes fundamentally different and far more difficult, however, when tracking is done with tactical low-altitude UAV's with on-board, 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.

This paper presents an algorithm for tracking an arbitrary number of ground targets from a single multirotor. For the purposes of algorithm development it is assumed that 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. The problem is framed as a receding-horizon optimal control problem, and a linear model predictive control algorithm is implemented assuming vehicle dynamics similar to a multirotor UAV. A particle filter algorithm predicts target motion over the receding control horizon. Furthermore, a target rejection algorithm ensures that the tracking vehicle does not exceed an altitude ceiling, ensuring that tracking resolution of the total target set is not sacrificed in order to track a single, rogue target. The paper proceeds as follows. First, the guidance algorithm is described including the trajectory generator, model

predictive controller, and target motion particle filter. Simulation results are provided for an example small-scale hexcopter aircraft, and trade studies are performed examining the effects of camera field of view, target rejection, and gimbaled vs non-gimbaled performance. Finally, results demonstrate that the algorithm enables simultaneous multi-target tracking performance for a broad range of ground target types and is robust to uncooperative target motion.

2. TRACKING ALGORITHM

The tracking algorithm operates in four layers. First, the trajectory predictions of each target are estimated using a simple target model and particle filter. A trajectory generator then takes the estimated path of all targets and constructs a desired trajectory through 3D space for the multirotor to follow. Linear model predictive control (MPC) with a finite time horizon is used to track this trajectory. Finally, a target rejection algorithm works to determine if any targets are driving the tracking vehicle to prohibitively high altitudes and reducing tracking performance for the majority of targets. Such "rogue" targets are removed from consideration through a mathematical rejection criteria.

2.1 Ground Target Model and Particle Filter

Targets are assumed fixed to a flat earth and are modeled as points in two-dimensional space (Figure 2.1).

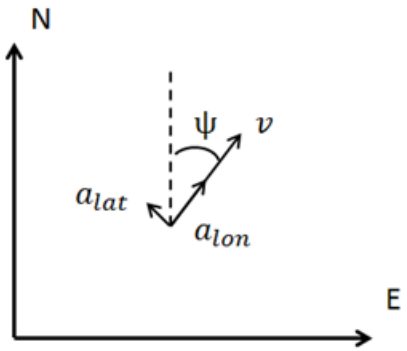

Figure 2.1: Target coordinate system and notation.

It is assumed that targets hold constant longitudinal and lateral accelerations over the planning horizon, and that lateral acceleration is reasonably small such that it does not change the ground speed of the target. The target equations of motion are below.

$$\dot{x}(t) = (a_{lon}t + v_0)\sin\left[\psi(t)\right] \tag{2.1}$$

$$\dot{y}(t) = (a_{lon}t + v_0)\cos\left[\psi(t)\right] \tag{2.2}$$

$$\psi = \begin{cases} \psi_0 - \dfrac{a_{lat}\ln\left[v_0\right]}{a_{lon}} + \dfrac{a_{lat}\ln\left(v_0 + a_{lon}t\right)}{a_{lon}} & a_{lon} \neq 0 \\ \psi_0 + \dfrac{a_{lat}t}{v_0} & a_{lon} \approx 0 \end{cases} \tag{2.3}$$

Accumulators are used to estimate the mean and variance of each target's course over ground (COG), ground speed, and longitudinal and lateral acceleration. Distributions of these variables are assumed Gaussian. Position, velocity, and COG measurements are taken from a GPS receiver and assumed deterministic due to the GPS's relatively high precision in taking these measurements. Longitudinal and lateral accelerations are determined by first order finite differencing applied to the groundspeed and COG. Online accumulation algorithms, as shown in (2.4)- (2.6), are taken from [9] and modified to include a forgetting factor.

$$\bar{\mathrm{X}}_{k+1} = \begin{cases} \dfrac{(n-1)\bar{\mathrm{X}}_k + \mathrm{X}_{k+1}}{n} & n \leq N \\ \dfrac{(N-1)\bar{\mathrm{X}}_k + \mathrm{X}_{k+1}}{N} & n > N \end{cases} \tag{2.4}$$

$$M_{k+1} = \begin{cases} M_k + \left(\mathrm{X}_{k+1} - \bar{\mathrm{X}}_{k+1}\right)^2 & n \le N \\ \left[1 - \dfrac{1}{N+1}\right]\left[M_k + \left(\mathrm{X}_{k+1} - \bar{\mathrm{X}}_{k+1}\right)^2\right] & n > N \end{cases} \tag{2.5}$$

$$\sigma_{\mathrm{X}_{k+1}} = \begin{cases} \sqrt{\dfrac{M_{k+1}}{n-1}} & n \le N \\ \sqrt{\dfrac{M_{k+1}}{N-1}} & n > N \end{cases} \tag{2.6}$$

Each target has a particle filter to estimate future target motion. Particles are sampled assuming a Gaussian distribution of the 4 unknowns on the right hand side of (2.1)-(2.3), given by a_{lat}, a_{lon}, v_0, and ψ_0. These equations are integrated using a cumulative trapezoidal method yielding particle trajectories over a finite time horizon. Particles are regenerated, not resampled, with each new GPS measurement.

2.2 Trajectory Generation

A time horizon is chosen and discretized as shown in (2.7). The desired rotorcraft trajectory is generated on this time horizon to keep all targets in the camera's field of view at the lowest altitude possible. Figure 2.2 presents a graphical example of how this is done. The desired altitude over the planning horizon is a constant determined by the minimum altitude necessary to keep all the targets' particles within the camera's field of view at the end of the planning horizon ($t = H$). For this work, H was chosen to be 5 seconds. Optionally, a buffer zone can be designated near the edge of the camera frame to keep targets from approaching the edge of the field of view, guarding against modeling errors and target prediction uncertainty. Use of this buffer zone has the effect

of increasing the multirotor's altitude and increasing the probability that targets will be within the camera's field of view. At the same time, this altitude increase means that the targets are viewed from a longer distance.

$$\bar{t} = \begin{bmatrix} 0 & \Delta t & 2\Delta t & \dots & H \end{bmatrix} \tag{2.7}$$

The generated trajectory is stacked into a single vector as shown in (2.8). The reason for this unusual stacking will be made clear in Section 2.4. The zeroes in (2.8) represent the desired roll and pitch angles. Setting them to zero allows penalizing of roll and pitch angles within the MPC. The second instance of the x and y positions in the non-gimbaled output equation is specifying the location of x_{cam} and y_{cam}, the center of the camera frame.

$$\left[\tilde{Y}\right] = \begin{cases} \begin{bmatrix} \tilde{x}_k & \tilde{y}_k & \tilde{z} & 0 & 0 & \dots & \tilde{x}_{k+K} & \tilde{y}_{k+K} & \tilde{z} & 0 & 0 \end{bmatrix}^t & gimbaled \\ \begin{bmatrix} \tilde{x}_k & \tilde{y}_k & \tilde{z} & 0 & 0 & \tilde{x}_k & \tilde{y}_k & \dots & \tilde{x}_{k+K} & \tilde{y}_{k+K} & \tilde{z} & 0 & 0 & \tilde{x}_{k+K} & \tilde{y}_{k+K} \end{bmatrix}^t & non-gimbaled \end{cases} \tag{2.8}$$

Due to the inherent symmetry of a multirotor, it is assumed that yaw angle has no effect on the vehicle's motion. For simplicity the vehicle yaw angle is assumed to be zero at all times and is excluded from both the trajectory generation and system model.

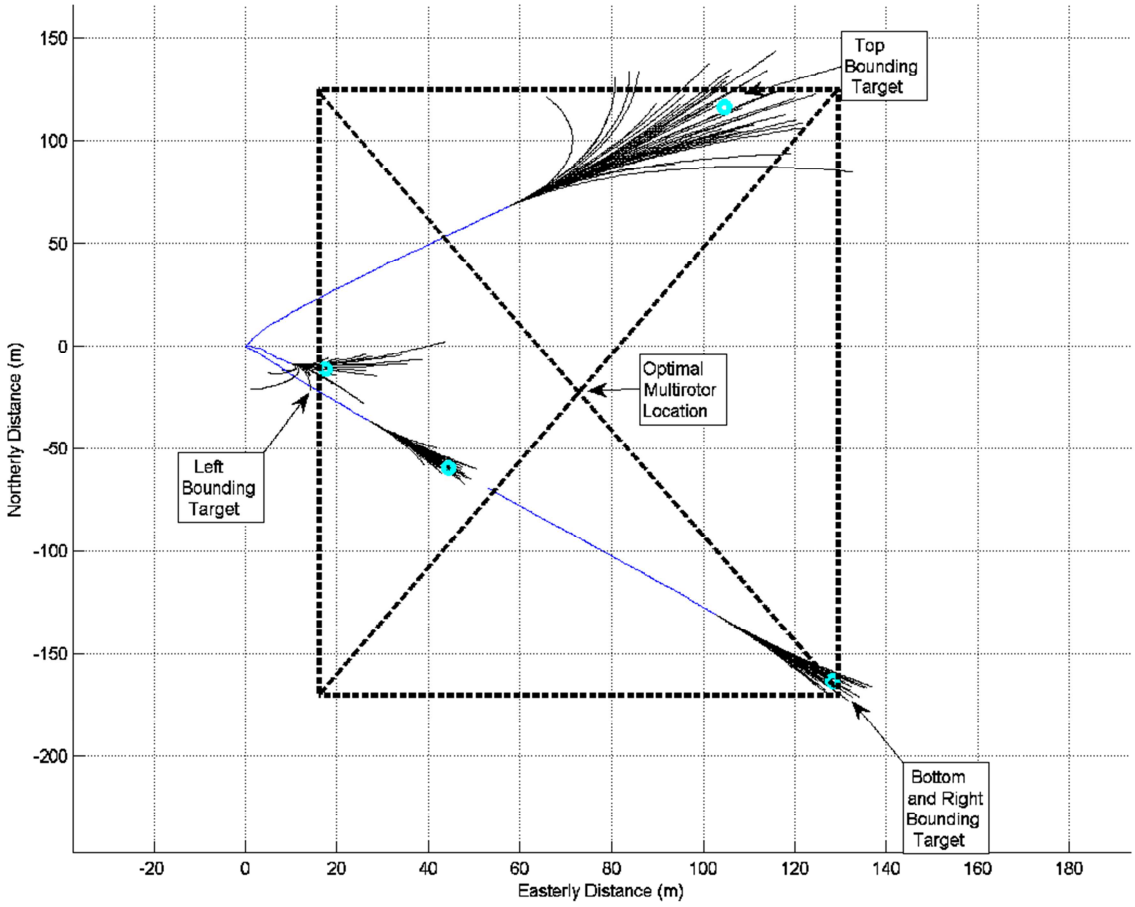

Figure 2.2: Determining multirotor position. The dark blue lines are observed paths of the targets. The many black lines emanating from the blue lines are possible target paths predicted by the particle filter. Light blue circles represent the expected location of each target at a future time. A bounding rectangle is drawn which contains the expected locations of all targets at the future time. The desired multirotor location is the center of this rectangle.

2.3 Multirotor Model

A simple linear model of a multirotor is constructed assuming the rotorcraft contains a stabilization system capable of tracking commanded Euler angles with first order lag. Translational acceleration is accomplished by vectoring the thrust vector using the roll and pitch degrees of freedom. Linearized drag is added to the three translational

velocities. The controls are taken to be commanded roll rate, commanded pitch rate, and commanded magnitude of thrust. Two controller states, commanded roll and commanded pitch, are also included in the state vector. In this way both control and control rate can be penalized within the model predictive controller.

Thrust is non-dimensionalized by vehicle mass, resulting in equations of motion independent of vehicle mass. Thrust, roll, pitch, roll rate, and pitch rate are constrained to reasonable values, discussed further in Section 3.2. The effect of gravity is not included in the system model. Rather, lack of gravity is accounted for by decreasing the non-dimensionalized thrust by 1.

$$\bar{x} = \begin{bmatrix} \phi & \theta & T & x & y & z & \dot{x} & \dot{y} & \dot{z} & \phi_c & \theta_c \end{bmatrix}^t \tag{2.9}$$

$$\bar{u} = \begin{bmatrix} \dot{\phi}_c & \dot{\theta}_c & T_d \end{bmatrix}^t \tag{2.10}$$

$$A = \begin{bmatrix} -\tau_\phi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tau_\phi & 0 \\ 0 & -\tau_\theta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tau_\theta \\ 0 & 0 & -\tau_T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & c_{d_x} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & c_{d_y} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & c_{d_z} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.11}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \tau_t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.12}$$

$$\bar{Y} = \begin{cases} \begin{bmatrix} x & y & z & \phi & \theta \end{bmatrix}^t & gimbaled \\ \begin{bmatrix} x & y & z & \phi & \theta & x_{cam} & y_{cam} \end{bmatrix}^t & non-gimbaled \end{cases} \tag{2.13}$$

$$x_{cam} = x + z\tan(-\varphi) \simeq x - z\varphi \tag{2.14}$$

$$y_{cam} = y + z\tan(-\theta) \simeq y - z\theta \tag{2.15}$$

The x_{cam} and y_{cam} terms in (2.13) are a nonlinear product of altitude and orientation angle. Gain scheduling is implemented to retain linearity in the equations of motion. The user defined permissible altitude range is discretized into ten intervals on a logarithmic spacing and (2.16) is used to generate output matrix C assuming altitude z is a constant on the interval. In this way, the standard linear system equations (2.17) and (2.18) can be used.

$$C = \begin{cases} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} gimbaled \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -z_{nearest} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -z_{nearest} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} non-gimbaled \end{cases} \tag{2.16}$$

$$\dot{\bar{x}} = A\bar{x} + B\bar{u} \tag{2.17}$$

$$\bar{Y} = C\bar{x} \tag{2.18}$$

The continuous system equations of motion (2.17) and (2.18) are integrated using the continuous to discrete transform to arrive at the discrete equations of motion (2.19) and (2.20).

$$\bar{x}_{k+1} = A_d \bar{x}_k + B_d \bar{u}_k \tag{2.19}$$

$$\bar{Y}_k = C_d \bar{x}_k \tag{2.20}$$

2.4 Linear Model Predictive Controller

A linear model predictive controller is constructed based on [8]. The control vector is stacked as shown in (2.21). A standard quadratic cost function is constructed to penalize both the states and controls (2.22).The estimated output over the planning

horizon $\left[\tilde{Y}\right]$ for control sequence $\left[\bar{u}\right]$ is calculated using the discrete system model. Generating the block matrices k_{ca} and k_{cab} as shown in (2.24) and (2.25), respectively, facilitates calculations.

$$\left[\bar{u}\right] = \begin{bmatrix} \bar{u}_{k+1}^t & \bar{u}_{k+2}^t & \ldots & \bar{u}_{k+K}^t \end{bmatrix}^t \tag{2.21}$$

$$J = \left(\left[\bar{Y}\right] - \left[\tilde{Y}\right]\right)^t Q\left(\left[\bar{Y}\right] - \left[\tilde{Y}\right]\right) + \left[\bar{u}\right]^t R\left[\bar{u}\right] \tag{2.22}$$

$$\left[\bar{Y}\right] = k_{ca}\bar{x}_k + k_{cab}\left[\bar{u}\right] \tag{2.23}$$

$$k_{ca} = \begin{bmatrix} C_d A_d \\ C_d A_d^2 \\ \vdots \\ C_d A_2^H \end{bmatrix} \tag{2.24}$$

$$k_{cad} = \begin{bmatrix} C_d A_d^0 B_d & 0 & 0 & \ldots & 0 \\ C_d A_d^1 B_d & C_d A_d^0 B_d & 0 & \ldots & 0 \\ C_d A_d^2 B_d & C_d A_d^1 B_d & C_d A_d^0 B_d & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ C_d A_d^H B_d & C_d A_d^{H-1} B_d & \ldots & C_d A_d^1 B_d & C_d A_d^0 B_d \end{bmatrix} \tag{2.25}$$

Setting the derivative of (2.22) with respect to $\left[\bar{u}\right]$ equal to zero and solving yields the optimal control vector over the planning horizon (2.26). Because the MPC is

re-run at each Δt the only part of $[\bar{u}]$ needed is $\bar{u}_{k+1}$. Extracting just the top block-row of $[k]$ allows the more computationally efficient calculation in (2.27)

$$\begin{aligned}[k] &= \left(k_{cab}^{t} Q k_{cab} + R\right)^{-1} k_{cab}^{t} Q \\ [\bar{u}] &= [k]\left(\left[\tilde{Y}\right] - k_{ca}\bar{x}_k\right)\end{aligned} \tag{2.26}$$

$$\bar{u}_{k+1} = [k_{\min}]\left(\left[\tilde{Y}\right] - k_{ca}\bar{x}_k\right) \tag{2.27}$$

Recall from (2.16) the output matrix for the non-gimbaled case depends on altitude. Thus, each discretized altitude range has a $[k_{\min}]$ associated with it. When running the MPC the appropriate feedback matrix is selected based on current multirotor altitude. A block diagram of the control system which includes the gain scheduling is shown in Figure 2.3.

2.5 Target Rejection

Targets which are prohibitively difficult to keep in the field of view are automatically removed from the target set. The target rejection routine is initiated once the multirotor has been within 10 percent of its user defined maximum altitude for two seconds. Three metrics are used to determine which target is rejected; distance from other targets (2.28), course over ground correlation with the other targets (2.29), and number of times a given target has been at the edge of the camera's field of view. The three metrics are weighted by experimentally determine weights, summed, and the target with the greatest score is rejected.

$$r_j^2 = \left(x_j - \frac{1}{N}\sum_{n=1}^{N} x_n \right)^2 + \left(y_j - \frac{1}{N}\sum_{n=1}^{N} y_n \right)^2 \tag{2.28}$$

$$\psi_j^{corelation} = \psi_j - \frac{1}{N}\sum_{n=1}^{N} \psi_n \tag{2.29}$$

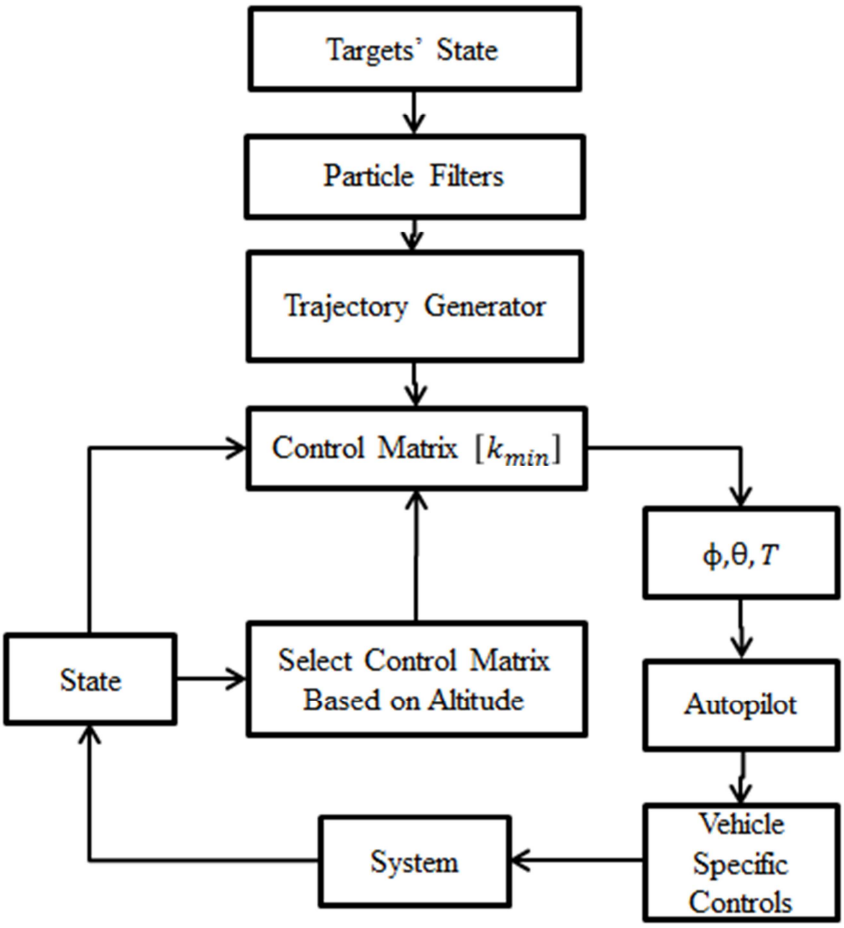

Figure 2.3: Control system block diagram.

3. SIMULATION

3.1 Target Data

Data is collected by placing a GPS receiver on three different ground vehicles; a pedestrian on foot, a bicyclist, and an automobile. The resulting trajectories are recorded. This data is post-processed to determine the four unknowns on the right had side of (2.1) -(2.3) then interpolated onto a uniform time grid.

Walking data is collected on sidewalks in a small residential area. Every effort is made to ensure the GPS receiver has a clear view of the sky during the trajectory recording sessions. No further special accommodations are made. The pace under motion is about 2 m/s.

Biking takes place in a low traffic research district. Again, the route is intentionally planned to maintain a clear view of the sky. Due to strong winds, riding speed varies from 5 m/s when traveling against the wind to 9 m/s with the wind. All traffic laws are followed including yielding to other pedestrians and motor vehicles.

Driving is performed in a large residential area. Roads are selected which have clear view of the sky and speed limits under 18 m/s. This is done to ensure the car does not drive faster than the hexcopter is capable of flying. Normal traffic laws are followed. Other than selecting roads with appropriate speed limits no effort is made to drive out of the ordinary while collecting data.

For a given simulation a data set is selected at random. Small perturbations are applied in heading, velocity, start time, velocity, and start location. The tracking algorithm is then tasked with tracking 5 sets of randomly perturbed data. A sample data

set generated from walking trajectories is shown in Figure 3.1.

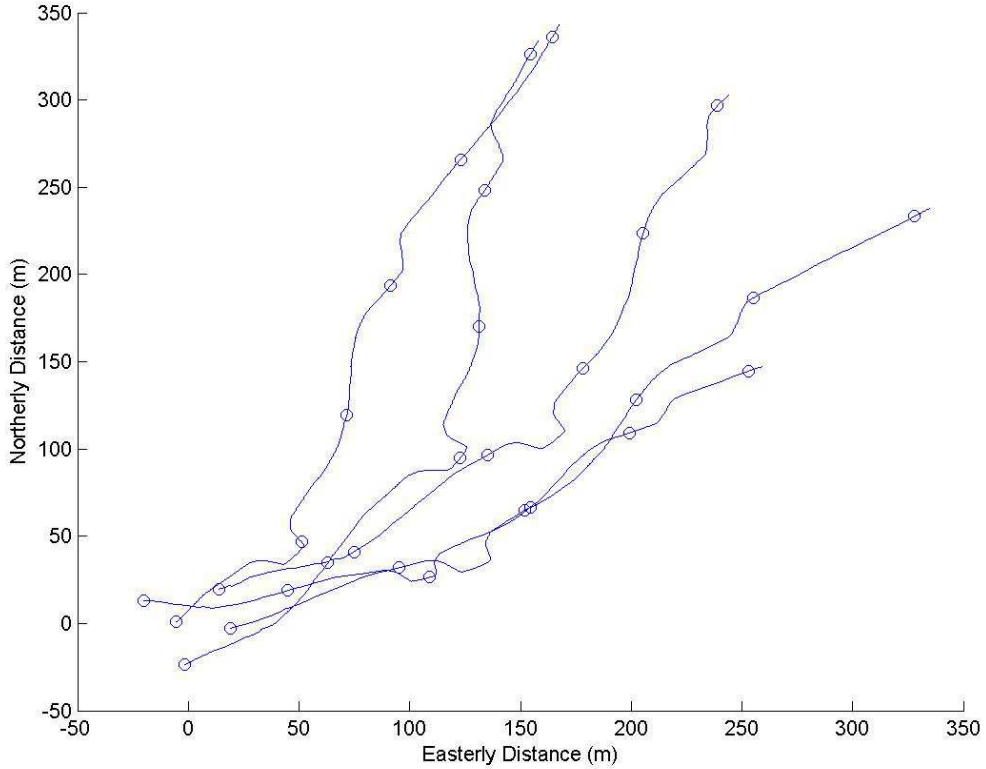

Figure 3.1: Example of randomized target trajectories. Targets begin near the origin and move in a north easterly direction.

3.2 System Identification

A hexcopter is instrumented with an autopilot tasked with tracking human pilot commanded roll and pitch angles. Roll and pitch doubles are applied to determine appropriate rate limits and time constants (Figure 3.2). Performance testing is conducted to determine maximum speed and thrust. Parameters used in the simulation are in Table 3.1. Linearized coefficients of drag are determined from maximum speed by Newton's Second Law applied at equilibrium at the maximum allowed roll and pitch angle (designated by the subscript max). It is assumed the only lateral and longitudinal forces

present are the horizontal component of thrust and aerodynamic drag. The equation to determine coefficients of drag from maximum speed and bank angle is developed in (3.1)- (3.2). The same equations are used to find c_{d_y} with the necessary substitutions for $|\dot{y}_{\max}|$ and $\theta_{\max}$ in place of $|\dot{x}_{\max}|$ and $\phi_{\max}$, respectively.

$$\sum F_x / m = \ddot{x} = c_{d_x} \dot{x} - \tan\phi = 0 \tag{3.1}$$

$$c_{d_x} = \frac{\tan(\phi_{\max})}{|\dot{x}|_{\max}} \tag{3.2}$$

Parameter	Value
τ_ϕ, τ_θ	5
τ_T	2
$\|\dot{x}\|_{\max}, \|\dot{y}\|_{\max}$	22 m/s
c_{d_x}, c_{d_y}	0.045
c_{d_z}	0.06
T	[-0.8 1][1]
$\|\phi\|_{\max}, \|\theta\|_{\max}$	45 deg
$\|\dot{\phi}\|_{\max}, \|\dot{\theta}\|_{\max}$	90 deg/sec

Table 3.1: Experimentally determined system parameters. These parameters are used in the system model presented in (2.11) and (2.12).

[1] Recall gravity is accounted for by decreasing the non-dimensionalized thrust by 1, so a non-dimensionalized thrust value of -0.8 is not physically impossible.

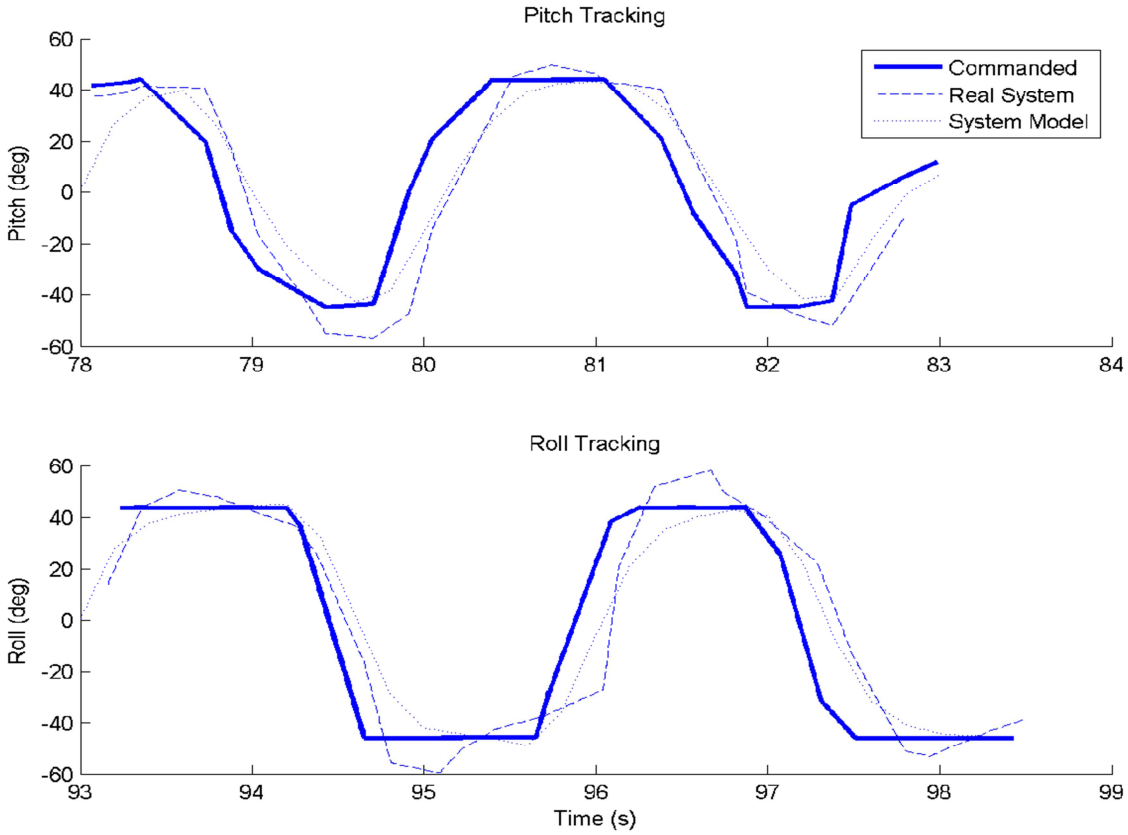

Figure 3.2: Application of doublets and system response.

3.3 Example Trajectories

An example trajectory generated by tracking walking targets with a gimbaled camera is shown in Figure 3.3. The hexcopter maneuvers very aggressively to keep on the desired trajectory while the gimbal keeps the camera point at the targets by mostly pointing straight down.

An example trajectory for the same case except with a non-gimbaled camera is shown in Figure 3.4. The hexcopter maneuvers aggressively for only a few tens of seconds to approach the targets, then keeps the roll and pitch angles small so the targets are kept within the camera's field of view.

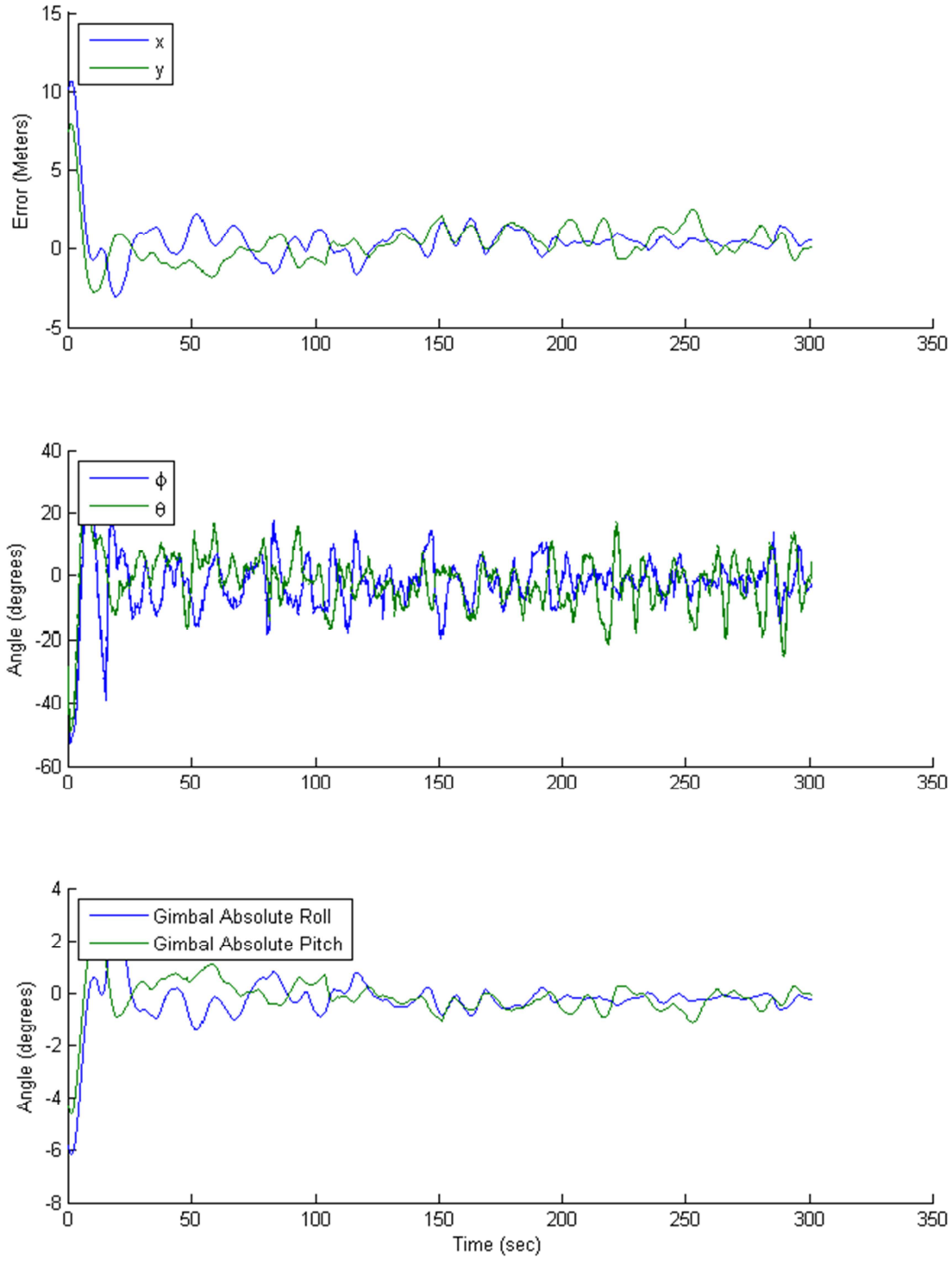

Figure 3.3: Example trajectory from gimbaled tracking session.

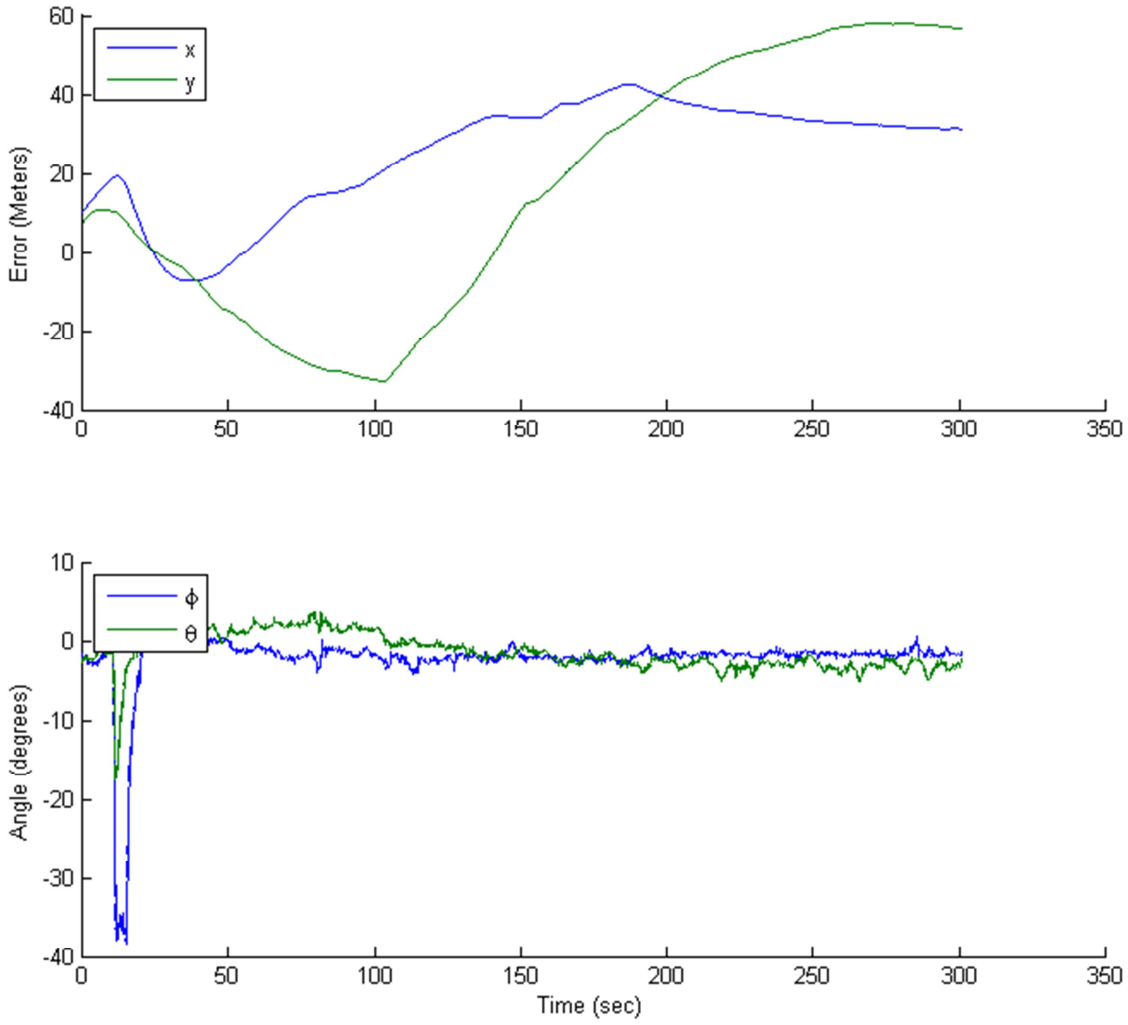

Figure 3.4: Example trajectory from non-gimbaled tracking session.

3.4 Trade Studies

A Monte-Carlo based trade study is conducted to explore sensitivity to different simulation parameters. Three metrics are used to judge the success of a tracking session: normalized flight time, percentage of targets outside the video camera frame, and mean

altitude. As a means of robustness each data point is run 35 times with different randomized data sets and the results are averaged.

Normalized flight time is a metric based on energy necessary to track targets. For example, a vehicle with 10 minute endurance while hovering conducting a tracking session with a normalized flight time of 80% would have an 8 minute endurance. In calculating normalized flight time it is assumed power consumption is directly proportional to thrust.

A tracking session with 0% targets outside of the camera frame is able to keep all targets inside the camera frame at every instant in time. One with 100% targets outside of the camera frame has every single target outside the camera frame at every instant in time.

3.4.1 Camera Frame Buffer Zone

Mean altitude is varied by adjusting the buffer zone around the edge of the camera frame. Target rejection is not allowed in this trade study. Maximum allowed hexcopter altitude is chosen to be 1000 m as to not restrict the trajectory.

Results in Figure 3.5 show the gimbaled case performs well for all vehicles independently of the camera frame buffer zone. These results also suggest that decoupling between vehicle motion and camera attitude is a key element in the algorithm's performance. Normalized flight time is not strongly dependent on camera frame buffer zone size; although a slight trend towards decreased flight time is exhibited as the buffer zone increases.

For the two quicker moving targets (bicycles and cars), the non-gimbaled case's ability to keep targets within the camera's field of view at low camera buffer zones is poor. The mean altitude must increase by nearly a factor of two before the non-gimbaled case has success on the same order of magnitude as the gimbaled case. The non-gimbaled case does, however, have considerably longer flight times. This is because the non-gimbaled case does not maneuver as aggressively, as aggressive maneuvering with a non-gimbaled camera will result in the camera pointing far from the location directly below the multirotor.

3.4.2 Target Rejection

The tracking session is modified to include one target whose motion is substantially dissimilar from the motion of the other targets (Figure 3.6). Simulation is performed with both target rejection allowed and disallowed. Once a target is rejected it is counted as outside camera frame for the rest of the tracking session. The hexcopter's maximum allowable altitude during the target rejection simulations is 200m. The camera's field of view is set to 70 degrees and the camera frame buffer zone to 0%.

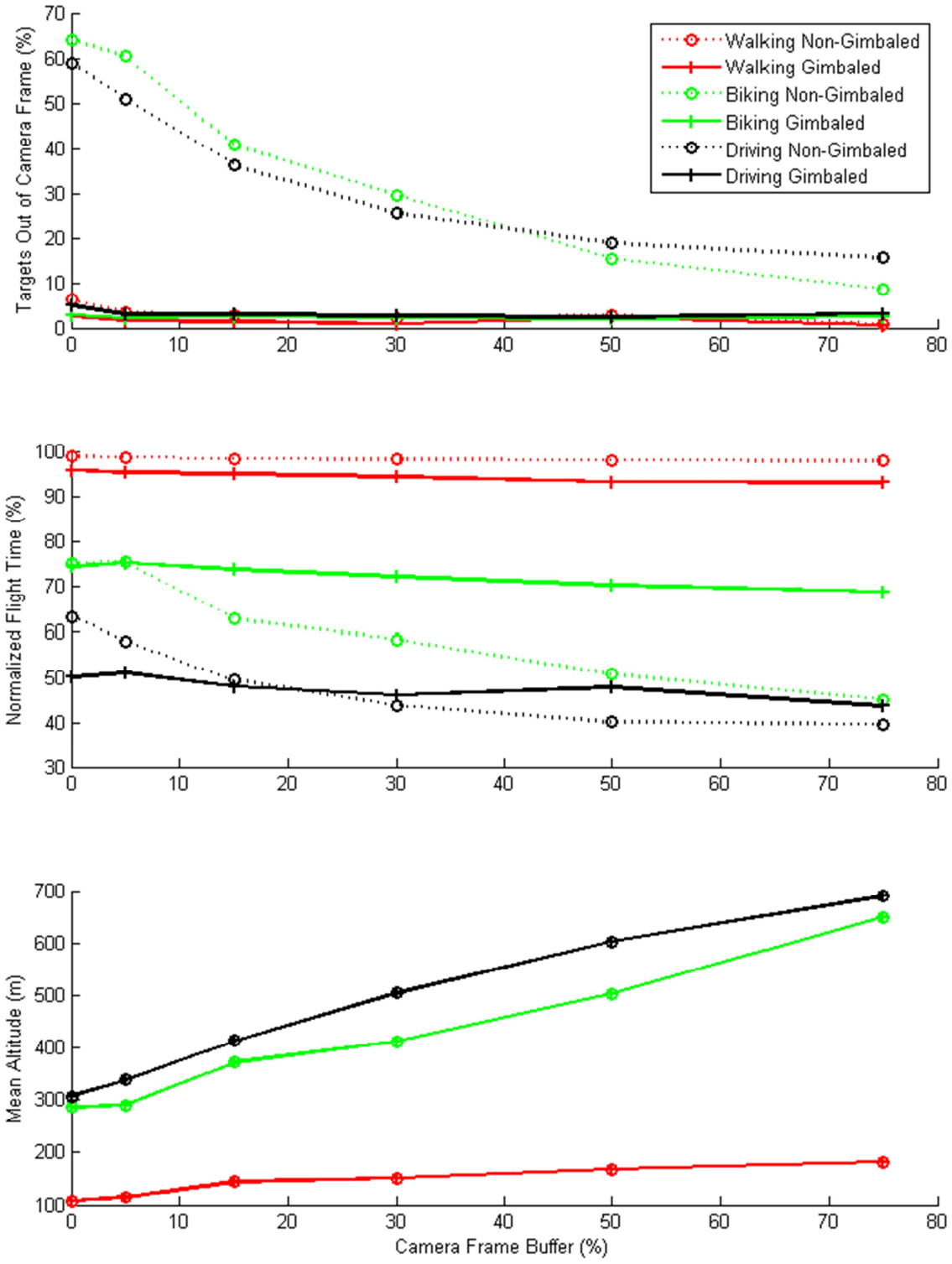

Figure 3.5: Results from camera frame buffer trade study.

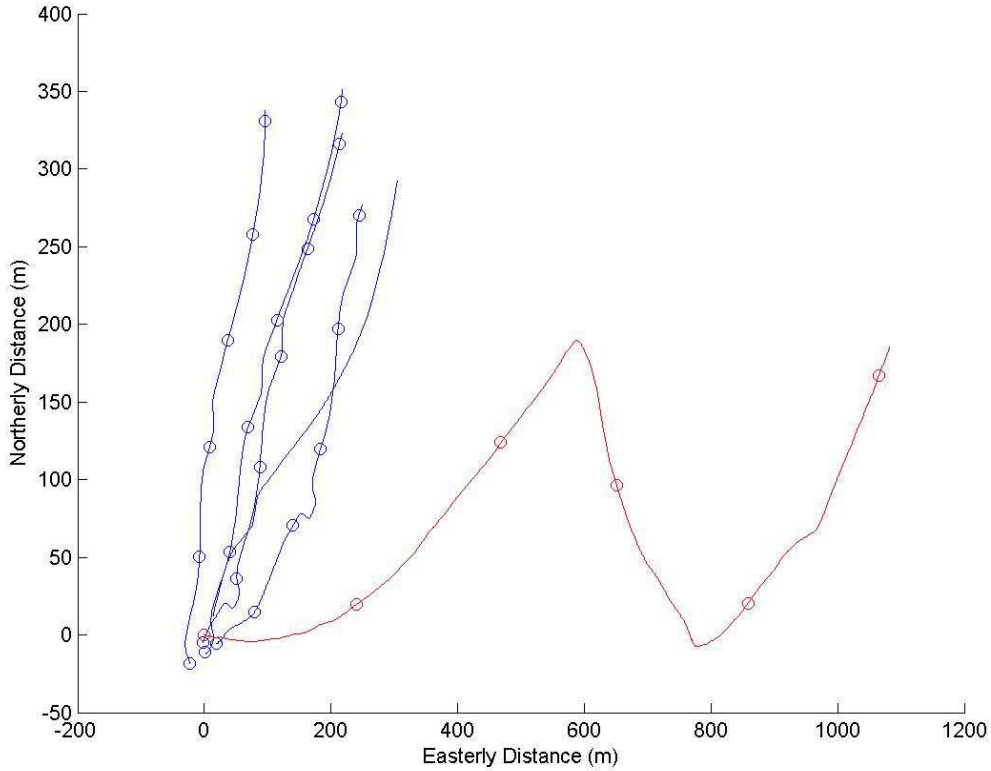

Figure 3.6: Example of target trajectories with one dissimilar target. Dissimilar target's path is shown in red. Targets begin near the origin and move in a north easterly direction.

Results from target rejection simulation are presented in Table 3.2. The target rejection algorithm is able to significantly improve tracking performance of all metrics by removing the rogue target from the target set.

	Gimbaled		Non Gimbaled	
Rejection	No	Yes	No	Yes
Out of Frame (%)	61	23	56	28
Normalized Flight Time (%)	83	91	98	98
Mean Altitude (m)	187	113	187	113

Table 3.2: Target rejection results for both gimbaled and non-gimbaled case.

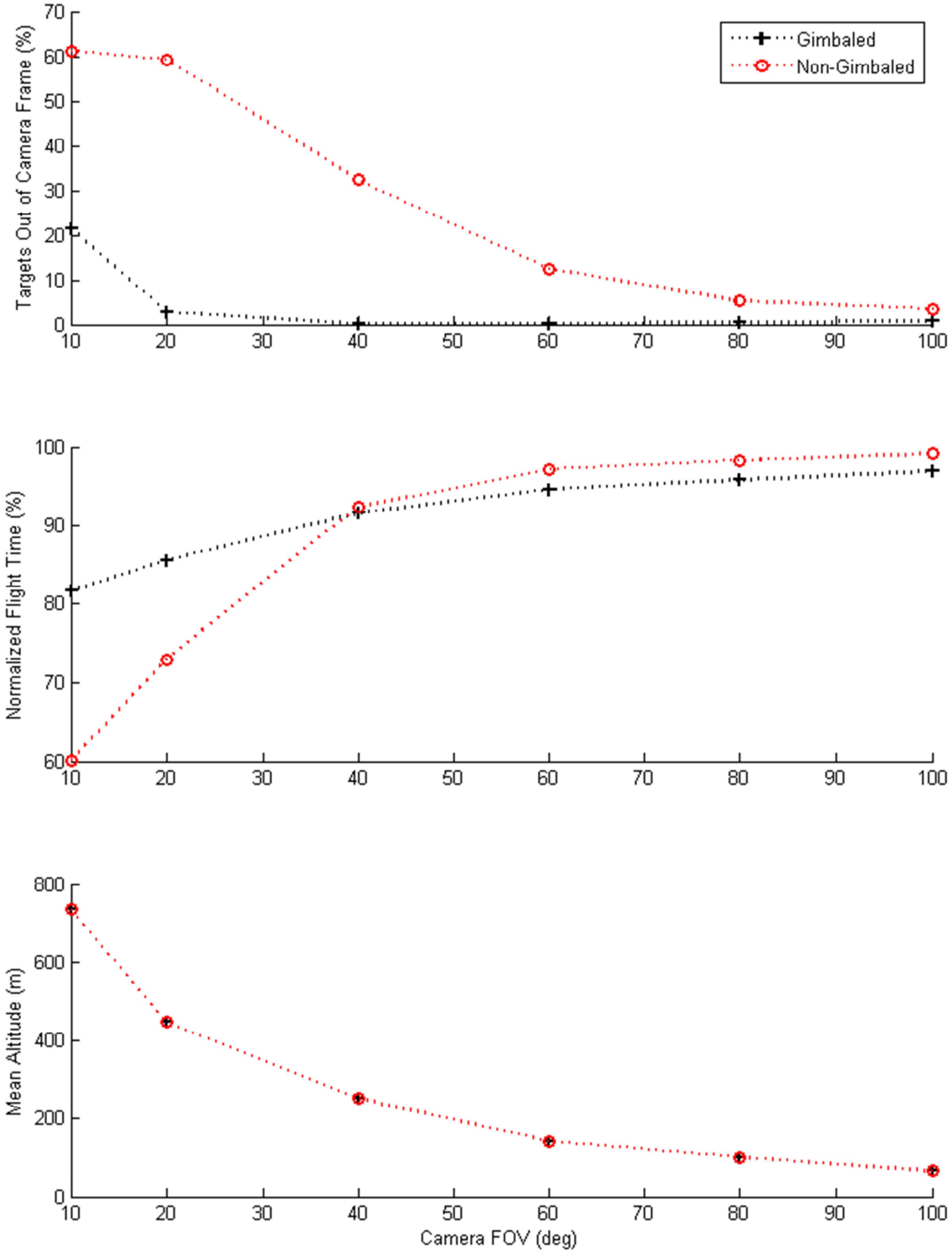

Figure 3.7: Results from field of view trade study.

3.4.3 Camera Field of View

The camera's field of view is varied and its effects on controller performance are demonstrated as shown in Figure 3.7. The simulation is conducted using walking trajectories with target rejection disallowed and a camera frame buffer zone of zero. The maximum allowable altitude is 1000m.

An increasing camera field of view increases performance as judged by all metrics. Most prominent is the ability of the non-gimbaled case to capture targets within the camera's frame for wide fields of view. The wider field of view affords the vehicle greater maneuverability while still keeping targets within the camera's frame. Not examined is the effect of field of view on the camera's ability to resolve targets.

4. CONCLUSIONS

An algorithm for tracking multiple ground targets from a multirotor is proposed and evaluated. Simulation results demonstrate the validity of the algorithm when given real world data. Target tracking is possible with both gimbaled and non-gimbaled camera, though ability to keep targets within the camera's field of view is substantially degraded when using a non-gimbaled camera at lower altitudes.

Perhaps the most interesting data from simulation is Figure 3.5. These data suggest the tracking algorithm is capable of tracking targets ranging from near stationary (walking) to approaching the maximum speed of the tracking vehicle itself (driving), though performance as measured through normalized flight time and non-gimbaled percentage of targets outside of camera frame suffers as target speed increases.

Plausibility is shown for video tracking using a non-gimbaled camera rigidly attached to the multirotor. The only sacrifice made is the need to increase the camera frame buffer, thereby increasing mean altitude. For extremely lightweight multirotors where carrying a gimbal is mass prohibitive this could be a viable option, especially for cameras with wide field of views.

Of interest to discuss are two matters not examined in this study. The first is the availability of information that comes from mounting a GPS receiver on the target vehicles. Such mounting is obviously not practical should this algorithm be fielded for military or police purposes. In this work having a target outside the camera's field of view does not decrease knowledge of the motion of the target. In actuality, there would be a complex interaction between losing information about the motion of a target and the

optimal future motion of the tracking multirotor. Perhaps it would be desirable to alternate which targets are within the camera's field of view at a given instant, e.g. targets 1-3 for a brief period of time followed by targets 4 and 5 for a brief period and repeat. This extension is left to future work.

Second, the effect of increased separation between the target and tracking multirotor on knowledge of target motion is not explored. A video processing algorithm will have more difficulty identifying and tracking a target at an altitude of 700m than it will at 100m. It may even be possible the resolution of the camera is such that the target is not distinguishable at greater altitudes.

Future work will focus on a real time implementation of the tracking algorithm. GPS information from a small number of targets will be wirelessly transmitted to a computer, which will run the particle filters and MPC and send commanded roll and pitch angles, and thrust to the hexcopter via telecommands. Video from the multirotor will be collected and will be used to determine success in keeping targets within the camera's field of view.

REFERENCES

[1]Vladamir N. Dobrokhodov, Isaac I. Kamineer, Kevin D. Jones, and Reza Ghobcheloo. "Vision-Based Tracking and Motion Estimation for Moving Targets using Small UAVs," Proceedings of the 2006 American Control Conference, Minneapolis, Minnesota, June 14-16 2006.

[2]Niki Regina and Matteo Zanzi. "Fixed-Wing UAV Guidance Law for Surface-Target Tracking and Overflight," Proceedings of Aerospace Conference. Big Sky, Mt., 2009.

[3]Arambel, P., Antone, M., Rago, C., Landau, H., Strat, T., "A Multiple-Hypothesis Tracking of Multiple Ground Targets from Aerial Video with Dynamic Sensor Control," 7th International Conference on Information Fusion, 28 June – 1 July 2004, Stockholm, Sweden.

[4]Girard, A., Howell, A., Hedrick, J. K., "Border Patrol and Surveillance Missions Using Multiple Unmanned Air Vehicles," 43rd IEEE Conference on Decision and Control, December 14-17, 2004, Atlantis, Bahamas.

[5]Kingston, D., "Decentralized Control of Multiple UAV's for Perimeter and Target Surveillance," PhD Dissertation, Brigham Young University, December 2007.

[6]Sinha, A., Kirubarajan, T., Bar-Shalom, Y., "Autonomous Ground Target Tracking by Multiple Cooperative UAV's," Proceedings of the 2005 IEEE Aerospace Conference, Big Sky, MT, March 2005.

[7]Brown, A., Sullivan, K., Miller, D., "Feature-Aided Multiple Target Tracking in the Image Plane," Proceedings of the SPIE, Vol. 6229, 62290Q-1, 2006.

[8] Camacho, E., Bordons, C., Model Predictive Control, Springer-Verlag, London, 2004.

[9] Donald E. Knuth (1998). The Art of Computer Programming, volume 2: Seminumerical Algorithms, 3rd edn., p. 232. Boston: Addison-Wesley.