JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 34, No. 1, January–February 2011

Unmanned Aerial Vehicle Optimal Cooperative Obstacle Avoidance in a Stochastic Dynamic Environment

Carole G. Prévost* and André Desbiens†
Université Laval, Québec, Québec G1V 0A6, Canada
Eric Gagnon‡
Defence Research and Development Canada, Québec, Québec G3J 1X5, Canada
and
Daniel Hodouin§
Université Laval, Québec, Québec G1V 0A6, Canada

DOI: 10.2514/1.50800

Effective target-tracking and obstacle avoidance strategies are essential to the success of unmanned aerial vehicle missions. This paper presents an extended Kalman-filter-based algorithm that predicts the optimal three-dimensional trajectory and position prediction error of a dynamic object (obstacle or target) detected by an unmanned aerial vehicle. This trajectory prediction scheme is thereafter tested in a three-dimensional path planner for multiple unmanned aerial vehicles, which relies on decentralized model-based predictive control to calculate the optimal unmanned aerial vehicle setpoints that will lead each unmanned aerial vehicle to the interception of a single dynamic ellipsoidal target while avoiding dynamic ellipsoidal obstacles detected en route. A novel model-based predictive control collision avoidance algorithm is also presented in this paper. The method first computes the unmanned aerial vehicle collision probability with an obstacle by convolving the statistical distribution of the obstacle center of mass position with the obstacle shape. The method then seeks to minimize the unmanned aerial vehicle collision probability with all known obstacles on a future horizon, all while ensuring that the collision probability with any given obstacle at each prediction step does not surpass a preset threshold. Simulations are presented to demonstrate the effectiveness of the proposed approach.

Nomenclature

$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$	=	state-space matrices
$dg\{\bullet\}$	=	diagonal of matrix $\bullet$
$f_i(\bullet)$	=	generic nonlinear functions of $\bullet$
$G(z), \boldsymbol{G}(z)$	=	discrete-time transfer function and matrix
h_c, h_p	=	control and prediction horizons
$\boldsymbol{I}, \boldsymbol{0}$	=	identity and zero matrices
j	=	objective/cost function
k, τ	=	primary and secondary temporal indices
$\boldsymbol{n}$	=	position vector (center of mass)
$\boldsymbol{P}$	=	covariance of the extended Kalman filter estimation error
$\boldsymbol{q}, \boldsymbol{Q}$	=	ellipsoid semiaxes vector and equation matrix
$\boldsymbol{r}, \boldsymbol{R}$	=	Euler angles vector and rotation matrix
$\mathsf{s}, \mathcal{S}, \theta$	=	flat speed, speed, and flat headings
$\boldsymbol{T}$	=	translation matrix
t, t_s, t_m	=	time, sampling period, and total mission time
$\boldsymbol{u}$	=	input vector (autopilot setpoint vector)
$\boldsymbol{v}, \boldsymbol{V}$	=	measurement noise vector and covariance matrix
$\boldsymbol{w}, \boldsymbol{W}$	=	process noise vector and covariance matrix
$\mathsf{x}, \mathsf{y}, \mathsf{z}$	=	x axis, y axis, and z axis coordinates
$\boldsymbol{x}, \boldsymbol{y}$	=	state vector and output vector
$\mathcal{Z}\{\bullet\}$	=	Z transform of $\bullet$
δ, Δ	=	Dirac delta function and increment operator
$\boldsymbol{\lambda}, \boldsymbol{\Lambda}$	=	weight vector and weight matrix
$\boldsymbol{\sigma}^2$	=	position prediction error covariance matrix
$\hat{\bullet}$	=	optimal estimation of $\bullet$
$\check{\bullet}$	=	operating point of $\bullet$

Subscripts

cm	=	communication range
ob	=	obstacle
sn	=	sensor range
tg	=	target
uv	=	unmanned aerial vehicle

Superscripts

l	=	linear
m	=	measurement
nl	=	nonlinear

Received 28 May 2010; revision received 3 October 2010; accepted for publication 4 October 2010.

*Ph.D. Candidate, Laboratoire d'Observation et d'Optimisation des Procédés, Department of Electrical and Computer Engineering, 1065 av. de la Médecine. Student Member AIAA.

†Professor, Laboratoire d'Observation et d'Optimisation des Procédés, Department of Electrical and Computer Engineering, 1065 av. de la Médecine; Andre.Desbiens@ulaval.ca.

‡Defense Scientist, DRDC-Valcartier, 2459 Pie-XI Boulevard North.

§Professor, Laboratoire d'Observation et d'Optimisation des Procédés, Department of Mining, Metallurgical and Materials Engineering, 1065 av. de la Médecine.

I. Introduction

DURING World War I, the U.S. Navy constructed the first pilotless aircraft [1]. Today, unmanned aerial vehicles (UAVs) play a pivotal role in military operations [2] and have the potential to serve in countless civil applications [3]. One research area requiring further development is path planning for UAVs in stochastic dynamic environments.

Many advanced UAV guidance strategies are now based on model predictive control (MPC) algorithms. For UAVs operating in a stochastic dynamic environment, the development of effective obstacle and target trajectory prediction algorithms as well as the development of methods for integrating this information into the MPC performance criterion is thereby integral to mission success. These are the two main aspects studied in this paper.

When moving objects are subject to model and measurement uncertainties, Kalman filters are often used to estimate their current state before predicting their future trajectories [4–8]. In [5], a Kalman filter estimates the coefficients of the polynomials modeling the target velocity and heading dynamics. These polynomials together with transformation equations are then used to deterministically predict the next target position (throughout this text, the term position will always refer to the center of mass position). In [6], an artificial neural network coupled with a Kalman filter is used to simultaneously estimate the dynamical model and current state of a target. The estimated target dynamical model, initialized at the current estimated target state, is then used to deterministically predict the future target trajectory. In [7], Kalman filters are each used to calculate the current position Gaussian probability density function (PDF) of a detected target and its decoys. Heuristics are then used to bound the temporal evolution of these PDFs on a future horizon.

In this paper, a Kalman filter is also used to first estimate the state of a moving object detected by a UAV given model and measurement uncertainties. However, unlike in [5–7] but as in [8,9], the future uncertain object trajectory is evaluated by propagating the Kalman filter uncertain state estimates along the prediction horizon. The method consists of calculating the object center of mass trajectory that minimizes the position prediction error variance on the future horizon. This optimal object trajectory together with its associated position prediction error variance defines the temporal evolution of the future uncertain object position, statistically represented by three-dimensional (3-D) Gaussian PDFs.

UAV tracking and avoidance strategies are then developed as a function of the predicted target and obstacle trajectories. Artificial potential fields are a popular approach to UAV path planning [10–13]. Flat-sided functions [10], spherically symmetric functions [11], and superquadric functions [12] were proposed to construct potential field navigation functions, but the majority proved to be empirical, heuristic, and specific. Although some potential field navigation functions, such as the modified harmonic functions proposed by [13], are shown to be provably correct [14,15] (i.e., stable, local-minima free, and guaranteed convergence to target), they cannot account for vehicle dynamic constraints. On the other hand, MPC-based UAV path-planning strategies can account for vehicle constraints [16–22]. Unfortunately, their obstacle avoidance criterion is often very simplistic, modeled by a series of stop–reverse–forward commands [17,18], various inverse distance functions that do not account for obstacle shapes [19,20], or no-fly zones that can highly restrict the configuration space [21,22].

In this paper, a novel MPC obstacle avoidance criterion that is more sophisticated than those in [16–22] is presented. By exploiting the predicted uncertain obstacle trajectory derived from the proposed Kalman-filter-based trajectory prediction algorithm, obstacle avoidance is achieved by explicitly minimizing the UAV probability of collision with all detected obstacles on the prediction horizon while ensuring that, at each prediction step, it does not surpass a preset threshold. The probability of collision between a UAV and an obstacle of known shape at a specific prediction step is calculated by convolving the PDF of the obstacle uncertain center of mass location with the obstacle shape [15]. This approach generates obstacle avoidance navigation functions automatically and not heuristically, as is the case with most potential field methods.

As a secondary objective, the proposed object trajectory prediction and obstacle avoidance algorithms are thereafter tested in a path planner, developed in MATLAB®, which relies on receding horizon MPC [23] to guide a fleet of UAVs in an unknown 3-D stochastic dynamic environment. The mission objective is to ensure that each UAV intercepts the same moving ellipsoidal target while avoiding several moving ellipsoidal obstacles detected en route. For this purpose, each UAV is equipped with its own trajectory control unit (TCU) that generates its UAV optimal future trajectory by processing the measured UAV position supplied by the autopilot, the obstacle information gathered by the UAV sensors and from communicating UAVs (i.e., measured ellipsoidal obstacles positions, sizes, and orientations), and the target information gathered by the UAV sensors and from communicating UAVs or sent via a mission manager (i.e., measured ellipsoidal target position, size, and orientation).

In Sec. II, the Kalman-filter-based trajectory prediction algorithm for moving objects detected by a UAV is presented. In Sec. III, the proposed collision probability calculation is integrated into the control criterion of the UAV MPC path planner used to test the algorithm of Sec. II. In Sec. IV, five test cases are presented: the first three demonstrate the sole performance of the Kalman-filter-based trajectory prediction algorithm, whereas the last two illustrate its performance in the path planner. Appendices B and C provide some additional details on the ellipsoidal shape formulation and the convolution of shapes with PDFs. Finally, in Appendix A, a notation used throughout the remainder of this paper is established. It is recommended to read this Appendix first.

II. Trajectory Prediction Algorithm (Obstacle or Target)

A UAV detects and classifies a controlled moving object (e.g., an obstacle, such as a civilian aircraft to be avoided, or a target, such as a suspicious naval ship to be tracked) and measures its position. The object size and orientation are also measured by the UAV; however, these measurements are not integral to this algorithm. The UAV then identifies the object from which stems the model $\boldsymbol{G}(z)$, depicted in Fig. 1, that best describes the object dynamic behavior. Presentation of the mechanics involved in the classification and identification of an object is beyond the scope of this work. It is assumed that the object has been correctly classified and identified such that the model $\boldsymbol{G}(z)$ in Fig. 1 is fairly accurate.

In Fig. 1, $\boldsymbol{G}(z)$ represents the modeled closed-loop dynamics of the detected object. The outputs of $\boldsymbol{G}(z)$ (i.e., flat speed $\mathsf{s}(k)$, flat heading $\theta(k)$, and altitude $\mathsf{z}(k)$) thus follow the inputs of $\boldsymbol{G}(z)$ (i.e., flat speed setpoint $u_{\mathsf{s}}(k)$, flat heading setpoint $u_{\theta}(k)$, and altitude setpoint $u_{\mathsf{z}}(k)$) as specified by $\boldsymbol{G}(z)$. The object position (i.e., x-axis position $\mathsf{x}(k)$, y-axis position $\mathsf{y}(k)$, and z-axis position $\mathsf{z}(k)$) is then calculated from the outputs of $\boldsymbol{G}(z)$. While the object altitude directly corresponds to its z-axis position, coordinate transformation equations convert the object flat speed and flat heading into x-axis and y-axis positions.

The goal is to then predict the trajectory (i.e., optimal center of mass trajectory and position prediction error variance) of the detected object, knowing that $\boldsymbol{G}(z)$ is an estimation of the true object dynamics and knowing that the UAV sensors are noisy. For this purpose, the states of the object with motion model of Fig. 1 are first estimated from its measured position. The predicted object trajectory is thereafter calculated from this motion model initialized at the object state estimates.

A. State Estimation

An extended Kalman filter (EKF) is used to estimate the states of the object (with the motion model of Fig. 1) from its measured position. The resulting Kalman filter equations are

$$\boldsymbol{x}(k+1) = f_1[\boldsymbol{x}(k)] + \boldsymbol{w}(k) \tag{1}$$

$$\boldsymbol{y}(k) = f_2[\boldsymbol{x}(k)] + \boldsymbol{v}(k) \tag{2}$$

where the state vector $\boldsymbol{x}(k)$ and measurement vector $\boldsymbol{y}(k)$ are detailed next. As for the random vectors $\boldsymbol{w}(k)$ and $\boldsymbol{v}(k)$ (Gaussian, zero mean, white noise, and statistically independent), they represent the

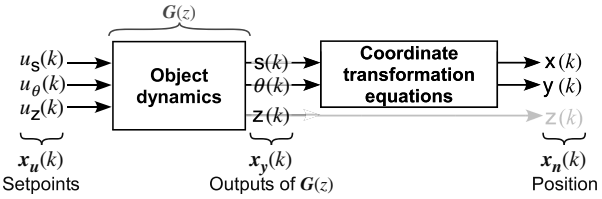


Fig. 1 Motion model of object detected by a UAV (obstacle, target).

uncertainty and perturbations on the model [Eq (1)] and the uncertainty on the measurements [Eq. (2)], respectively.

The EKF presented herein estimates the optimal object states at the next sample time (i.e., $\hat{\boldsymbol{x}}(k+1|k)$). This particular filter implementation is called a one-step prediction estimator [24]. The Bierman–Thornton upper diagonal filtering algorithm [25], which provides an alternate solution for the Kalman filter matrix Riccati difference equation, was also implemented in order to improve numerical robustness.

The states to be estimated by the EKF are those of the motion model of Fig. 1. They are the object setpoints $\boldsymbol{x}_u(k)$, the outputs of $\boldsymbol{G}(z)$ $\boldsymbol{x}_y(k)$, and the xy-planar position $\boldsymbol{x}_n(k)$. To capture the dynamic relation between states $\boldsymbol{x}_u(k)$ and $\boldsymbol{x}_y(k)$, the states internal of $\boldsymbol{G}(z)$, denoted as $\boldsymbol{x}_G(k)$, must also be estimated. The state vector in Eqs. (1) and (2) is thus

$$\boldsymbol{x}(k) = [\,\boldsymbol{x}_u(k)^T \quad \boldsymbol{x}_G(k)^T \quad \boldsymbol{x}_y(k)^T \quad \boldsymbol{x}_n(k)^T\,]^T \tag{3}$$

where

$$\boldsymbol{x}_u(k) = [\,u_{\mathsf{s}}(k) \quad u_\theta(k) \quad u_{\mathsf{z}}(k)\,]^T \tag{4}$$

$$\boldsymbol{x}_y(k) = [\,\mathsf{s}(k) \quad \theta(k) \quad \mathsf{z}(k)\,]^T \tag{5}$$

$$\boldsymbol{x}_n(k) = [\,\mathsf{x}(k) \quad \mathsf{y}(k)\,]^T \tag{6}$$

and where $\boldsymbol{x}_G(k)$ are the states of $\boldsymbol{G}(z)$.

Upon detection of the object, the UAV measures its position. The measurement vector in Eq. (2) is thus

$$\boldsymbol{y}(k) = [\,\mathsf{x}(k) \quad \mathsf{y}(k) \quad \mathsf{z}(k)\,]^T \tag{7}$$

1. State Equation

The evolution of the state vector $\boldsymbol{x}(k)$ in Eq. (3) during one sampling period t_s defines the state equation (1) needed for Kalman filtering.

The temporal evolution of the object setpoints $\boldsymbol{x}_u(k)$ is unknown. They are thereby assumed to behave like independent random walks [26]. Setpoint decoupling is a plausible assumption, since the detected object is considered a controlled system, producing

$$\boldsymbol{x}_u(k+1) = \boldsymbol{x}_u(k) + \boldsymbol{w}_u(k) \tag{8}$$

where $\boldsymbol{w}_u(k) = [\,w_{u_{\mathsf{s}}}(k) \quad w_{u_\theta}(k) \quad w_{u_{\mathsf{z}}}(k)\,]^T$ is a random vector comprising three independent random variables. The variances on each random variable in $\boldsymbol{w}_u(k)$ are set to model variations in the setpoints between sample times.

The temporal evolution of the states $\boldsymbol{x}_G(k)$ and $\boldsymbol{x}_y(k)$ is described by the object dynamical model $\boldsymbol{G}(z)$. The state-space representation of $\boldsymbol{G}(z)$ is thus

$$\boldsymbol{x}_G(k+1) = \boldsymbol{A}_G\boldsymbol{x}_G(k) + \boldsymbol{B}_G\boldsymbol{x}_u(k) + \boldsymbol{w}_G(k) \tag{9}$$

$$\boldsymbol{x}_y(k) = \boldsymbol{C}_G\boldsymbol{x}_G(k) \tag{10}$$

where the random vector $\boldsymbol{w}_G(k)$ models the uncertainty on $\boldsymbol{G}(z)$ (parametric and stochastic) and/or the perturbations on the states $\boldsymbol{x}_G(k)$.

The temporal evolution of the xy-planar object position $\boldsymbol{x}_n(k)$ is governed by the nonlinear coordinate transformation equations,

$$\begin{aligned}\mathsf{x}(k+1) &= \mathsf{x}(k) + t_s\mathsf{s}(k)\cos\theta(k)\\ \mathsf{y}(k+1) &= \mathsf{y}(k) + t_s\mathsf{s}(k)\sin\theta(k)\end{aligned} \tag{11}$$

or simply

$$\boldsymbol{x}_n(k+1) = \boldsymbol{x}_n(k) + f_3[\boldsymbol{x}_y(k)] \tag{12}$$

The EKF algorithm requires Eq. (12) to be linearized at each sample time. Thus, applying a Taylor approximation of the first order about the operating points $\check{\boldsymbol{x}}_y(k) = [\,\check{\mathsf{s}}(k) \quad \check{\theta}(k) \quad \check{\mathsf{z}}(k)\,]^T$ (vector $\check{\boldsymbol{x}}_y(k)$ is the EKF state estimates at the previous sample time) to Eq. (12) yields

$$\boldsymbol{x}_n(k+1) = \boldsymbol{x}_n(k) + \boldsymbol{B}_n(k)\boldsymbol{x}_y(k) + \boldsymbol{w}_n(k) \tag{13}$$

where

$$\boldsymbol{B}_n(k) = \begin{bmatrix} t_s\cos\check{\theta}(k) & -t_s\check{\mathsf{s}}(k)\sin\check{\theta}(k) & 0\\ t_s\sin\check{\theta}(k) & t_s\check{\mathsf{s}}(k)\cos\check{\theta}(k) & 0\end{bmatrix} \tag{14}$$

and where the random vector $\boldsymbol{w}_n(k) = [\,w_{\mathsf{x}}(k) \quad w_{\mathsf{y}}(k)\,]^T$ models the errors generated by the linear approximation.

Combining Eqs. (8–10) and (12) yields the complete Kalman filter nonlinear state equation:

$$\boldsymbol{x}(k+1) = f_1[\boldsymbol{x}(k)] + \boldsymbol{w}^{\mathrm{nl}}(k) \tag{15}$$

where

$$\boldsymbol{w}^{\mathrm{nl}}(k) = [\,\boldsymbol{w}_u(k)^T \quad \boldsymbol{w}_G(k)^T \quad \boldsymbol{w}_G(k)^T\boldsymbol{C}_G^T \quad \boldsymbol{0}^T\,]^T$$

with covariance matrix $\boldsymbol{W}^{\mathrm{nl}}$. As for the linear counterpart of Eq. (15), it is obtained by replacing Eq. (12) by Eq. (13), producing

$$\boldsymbol{x}(k+1) = \boldsymbol{A}(k)\boldsymbol{x}(k) + \boldsymbol{w}^l(k) \tag{16}$$

where

$$\boldsymbol{A}(k) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{B}_G & \boldsymbol{A}_G & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{C}_G\boldsymbol{B}_G & \boldsymbol{C}_G\boldsymbol{A}_G & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_n(k) & \boldsymbol{I}\end{bmatrix} \tag{17}$$

and where

$$\boldsymbol{w}^l(k) = [\,\boldsymbol{w}_u(k)^T \quad \boldsymbol{w}_G(k)^T \quad \boldsymbol{w}_G(k)^T\boldsymbol{C}_G(k)^T \quad \boldsymbol{w}_n(k)^T\,]^T$$

with covariance matrix $\boldsymbol{W}^l$ simulates the uncertainty on all system states.

2. Measurement Equation

The UAV measures the xyz-axis position of the detected object. The measurement equation is

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{v}(k) \tag{18}$$

where $\boldsymbol{y}(k)$ is Eq. (7), where $\boldsymbol{C}$ references the states in $\boldsymbol{x}(k)$ describing the object position, and where the random vector $\boldsymbol{v}(k) = [\,v_{\mathsf{x}}(k) \quad v_{\mathsf{y}}(k) \quad v_{\mathsf{z}}(k)\,]^T$ with covariance matrix $\boldsymbol{V}$ models sensor noises.

Let us consider a UAV fleet that is partitioned at each sample time into subfleets comprising members in communication range (e.g., a subfleet located far from all other subfleets would have limited or no communication with the collective). If the jth UAV part subfleet i detects the object, then its measurement equation is

$$\boldsymbol{y}_{ij}(k) = \boldsymbol{C}_{ij}\boldsymbol{x}(k) + \boldsymbol{v}_{ij}(k) \tag{19}$$

where $\boldsymbol{y}_{ij}(k)$, $\boldsymbol{C}_{ij}$, and $\boldsymbol{v}_{ij}(k)$ are, respectively, the object position measurement vector, observation matrix, and sensor noise vector of UAV_j in $\mathrm{subfleet}_i$. The corresponding covariance matrix $\boldsymbol{V}_{ij}$ associated with $\boldsymbol{v}_{ij}(k)$ may differ in nature and precision from any other UAV measurement noise covariance matrix and may even account for sensor failures.

If n communicating UAVs (member of the ith subfleet) detect the same object,[¶] Eq. (19) expands to

[¶]Two UAVs are assumed to have detected the same object if the difference in their object position measurements lies below a preset threshold.

$$\begin{bmatrix} \boldsymbol{y}_{i1}(k) \\ \vdots \\ \boldsymbol{y}_{in}(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_{i1} \\ \vdots \\ \boldsymbol{C}_{in} \end{bmatrix} \boldsymbol{x}(k) + \begin{bmatrix} \boldsymbol{v}_{i1}(k) \\ \vdots \\ \boldsymbol{v}_{in}(k) \end{bmatrix}$$

$$\boldsymbol{y}_i(k) = \boldsymbol{C}_i \boldsymbol{x}(k) + \boldsymbol{v}_i(k) \tag{20}$$

for which the corresponding measurement noise covariance matrix $\boldsymbol{V}_i$ of dimension $3n \times 3n$ is often chosen as block diagonal to represent independent UAV sensors, but this is not a necessary condition.

Furthermore, it is possible, using a maximum likelihood measurement fusion method, to combine all n object position measurements into a single pseudomeasurement, thus reducing Eq. (20) to

$$\boldsymbol{y}_{i_{n\to 1}}(k) = \boldsymbol{C}_{i_{n\to 1}} \boldsymbol{x}(k) + \boldsymbol{v}_{i_{n\to 1}}(k) \tag{21}$$

For this purpose, the detected object position $\boldsymbol{n}(k)$ must be directly observable from the object position measurements $\boldsymbol{n}_i^m(k)$. Equation (20) can thus be rewritten as

$$\boldsymbol{y}_i(k) = \boldsymbol{C}_i' \boldsymbol{n}(k) + \boldsymbol{v}_i(k) \tag{22}$$

yielding the object position pseudomeasurement

$$\begin{aligned} \boldsymbol{y}_{i_{n\to 1}}^m(k) &= \operatorname{argmin}_{\boldsymbol{n}(k)} [\boldsymbol{y}_i^m(k) - \boldsymbol{C}_i' \boldsymbol{n}(k)]^T \boldsymbol{V}_i^{-1} [\boldsymbol{y}_i^m(k) - \boldsymbol{C}_i' \boldsymbol{n}(k)] \\ &= (\boldsymbol{C}_i' \boldsymbol{V}_i^{-1} \boldsymbol{C}_i')^{-1} \boldsymbol{C}_i'^T \boldsymbol{V}_i^{-1} \boldsymbol{y}_i^m(k) \end{aligned} \tag{23}$$

with the measurement noise covariance matrix

$$\boldsymbol{V}_{i_{n\to 1}} = (\boldsymbol{C}_i'^T \boldsymbol{V}_i^{-1} \boldsymbol{C}_i')^{-1} \tag{24}$$

B. Trajectory Prediction

The future optimal center of mass trajectory of the object detected by the UAV is predicted from the states $\hat{\boldsymbol{x}}(k+1|k)$ estimated by the EKF of Sec. II.A.

$$\boldsymbol{n}(k+\tau) = [\mathsf{x}(k+\tau) \quad \mathsf{y}(k+\tau) \quad \mathsf{z}(k+\tau)]^T$$

and $\boldsymbol{C}$ is the same as in Eq. (18). Selecting $\boldsymbol{n}(k+\tau|k)$ that minimizes Eq. (25) for each $\tau \in \{1, \ldots, h_p\}$ produces the predicted optimal object trajectory,

$$\hat{\boldsymbol{n}}^{\mathrm{nl}}(k+1\colon k+h_p|k) = \begin{bmatrix} \boldsymbol{C}\hat{\boldsymbol{x}}(k+1|k) \\ \boldsymbol{C}f_1[\hat{\boldsymbol{x}}(k+1|k)] \\ \boldsymbol{C}f_1\{f_1[\hat{\boldsymbol{x}}(k+1|k)]\} \\ \vdots \\ \boldsymbol{C}f_1\{\cdots f_1[\hat{\boldsymbol{x}}(k+1|k)]\cdots\} \end{bmatrix} \tag{27}$$

where $\hat{\boldsymbol{x}}(k+1|k)$ are the EKF state estimates.

2. Position Prediction Error Variance

The quality of the predicted optimal object trajectory in Eq. (27) at each prediction step τ on h_p is the position prediction error variance,

$$\boldsymbol{\sigma}^2(k+\tau|k) = \mathrm{VAR}[\boldsymbol{n}(k+\tau) - \hat{\boldsymbol{n}}^{\mathrm{nl}}(k+\tau|k)] \tag{28}$$

where $\boldsymbol{n}(k+\tau)$ is Eq. (26) and where $\hat{\boldsymbol{n}}^{\mathrm{nl}}(k+\tau|k)$ is Eq. (27). Unfortunately, Eq. (28) is not easily evaluated and is therefore calculated from the approximation,

$$\boldsymbol{\sigma}^2(k+\tau|k) = \mathrm{VAR}[\boldsymbol{n}(k+\tau) - \hat{\boldsymbol{n}}^{l}(k+\tau|k)] \tag{29}$$

where the object position $\boldsymbol{n}(k+\tau)$ and predicted optimal object position $\hat{\boldsymbol{n}}^l(k+\tau|k)$ are now calculated from the linear dynamical model

$$\begin{aligned} \boldsymbol{x}(k+\tau+1) &= \boldsymbol{A}(k+\tau)\boldsymbol{x}(k+\tau) + \boldsymbol{w}^{\mathrm{nl}}(k+\tau) \\ \boldsymbol{n}(k+\tau) &= \boldsymbol{C}\boldsymbol{x}(k+\tau), \qquad \tau \geq 1 \end{aligned} \tag{30}$$

obtained by linearizing Eq. (26) about the operating point $\check{\boldsymbol{x}}^{\mathrm{nl}}(k+\tau)$. Errors $\boldsymbol{w}_n(k)$, due to linearizing the coordinate transformation equations, are not considered in this dynamical model [i.e., $\boldsymbol{w}^{\mathrm{nl}}(k+\tau)$ rather than $\boldsymbol{w}^l(k+\tau)$ in Eq. (30)], since the predicted optimal object center of mass trajectory is calculated via Eq. (26), which does not contain such errors.

Equation (29), evaluated at each $\tau \in \{1, \ldots, h_p\}$, thereby yields

$$\boldsymbol{\sigma}^2(k+1\colon k+h_p|k) = \begin{bmatrix} \boldsymbol{C}\boldsymbol{P}(k+1|k)\boldsymbol{C}^T \\ \boldsymbol{C}\boldsymbol{A}(k+1)\boldsymbol{P}(k+1|k)\boldsymbol{A}(k+1)^T\boldsymbol{C}^T + \boldsymbol{C}\boldsymbol{W}^{\mathrm{nl}}\boldsymbol{C}^T \\ \boldsymbol{C}\boldsymbol{M}_{(0,2)}\boldsymbol{P}(k+1|k)\boldsymbol{M}_{(0,2)}^T\boldsymbol{C}^T + \sum_{i=1}^{2} \boldsymbol{C}\boldsymbol{M}_{(i,2)}\boldsymbol{W}^{\mathrm{nl}}\boldsymbol{M}_{(i,2)}^T\boldsymbol{C}^T \\ \vdots \\ \boldsymbol{C}\boldsymbol{M}_{(0,h_p-1)}\boldsymbol{P}(k+1|k)\boldsymbol{M}_{(0,h_p-1)}^T\boldsymbol{C}^T + \sum_{i=1}^{h_p-1} \boldsymbol{C}\boldsymbol{M}_{(i,h_p-1)}\boldsymbol{W}^{\mathrm{nl}}\boldsymbol{M}_{(i,h_p-1)}^T\boldsymbol{C}^T \end{bmatrix} \tag{31}$$

1. Predicted Optimal Object Center of Mass Trajectory

The predicted optimal object trajectory minimizes, at each prediction step τ, the position prediction error variance

$$\boldsymbol{\sigma}^2(k+\tau|k) = \mathrm{VAR}[\boldsymbol{n}(k+\tau) - \boldsymbol{n}(k+\tau|k)] \tag{25}$$

where $\boldsymbol{n}(k+\tau)$ and $\boldsymbol{n}(k+\tau|k)$ are, respectively, the object position and predicted object position. The object position $\boldsymbol{n}(k+\tau)$ is computed from the nonlinear dynamical model,

$$\begin{aligned} \boldsymbol{x}(k+\tau+1) &= f_1[\boldsymbol{x}(k+\tau)] + \boldsymbol{w}^{\mathrm{nl}}(k+\tau) \\ \boldsymbol{n}(k+\tau) &= \boldsymbol{C}\boldsymbol{x}(k+\tau), \qquad \tau \geq 1 \end{aligned} \tag{26}$$

where the nonlinear state equation is Eq. (15),

where

$$\boldsymbol{M}_{(i,j)}^T = \prod_{m=i+1}^{j} \boldsymbol{A}(k+m)^T \tag{32}$$

III. Unmanned Aerial Vehicle Path Planner

The trajectory prediction algorithm of Sec. II is tested in a path planner that relies on optimal predictive control to guide a fleet of UAVs in an unknown 3-D environment. The mission objective is to ensure that each UAV intercepts a moving ellipsoidal target while avoiding several moving ellipsoidal obstacles detected en route. UAVs within communication range cooperate to improve their performances. A decentralized cooperative control structure regulates the flow of information between UAVs.

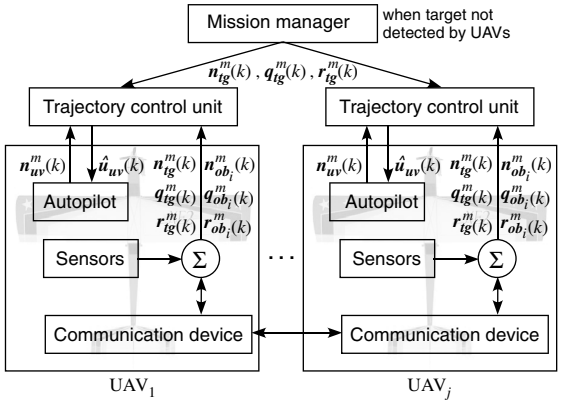


Fig. 2 High-level flowchart of 3-D path planner.

A. General Guidance Strategy

Each UAV is assumed equipped with sensors (e.g., video, electro-optic, and infrared devices), a communication device (e.g., radio-frequency transceiver), and an autopilot (Fig. 2). As the UAVs navigate in the unknown environment, they detect objects (obstacles and target) residing within their respective sensor range. UAVs also exchange object information with other UAVs residing within communication range.

At each sample time, UAVs transmit all acquired obstacle information [position $\boldsymbol{n}^m_{ob_i}(k)$, size $\boldsymbol{q}^m_{ob_i}(k)$, and orientation $\boldsymbol{r}^m_{ob_i}(k)$; see Sec. III.B.1] and target information (position $\boldsymbol{n}^m_{tg}(k)$, size $\boldsymbol{q}^m_{tg}(k)$, and orientation $\boldsymbol{r}^m_{tg}(k)$; see Sec. III.B.2) from sensors and/or communicating UAVs to their own TCU. If no target measurement is available, a mission manager supplies the TCU with virtual target measurements, as shown in Fig. 2. TCUs also receive, from their respective autopilots, the measured position of the aircraft $\boldsymbol{n}^m_{uv}(k)$. Each TCU then determines the optimal UAV autopilot setpoints $\hat{\boldsymbol{u}}_{uv}(k)$ that will guide the particular aircraft toward the moving target while avoiding moving obstacles. Finally, each autopilot translates these setpoints into lower-level commands supplied to the UAV actuators, all while maintaining the aircraft stability.

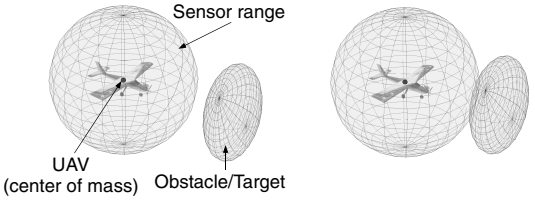
The ellipsoid was chosen to model various elements in the simulated environment since its basic shape allows for fewer and faster computations. These ellipsoids can represent true object shapes or may model safety regions surrounding the objects. In this path planner, the obstacles and the target are modeled as ellipsoids of various sizes and orientations. Ellipsoids are also employed to model each UAV sensor range and communication range. These ellipsoids are centered upon the UAV center of mass and mimic the aircraft motion.

1. *Object Detection Scheme*

Sensors onboard the UAV allow the aircraft to characterize its surrounding environment. An object (obstacle or target) is detected by the UAV when the ellipsoidal UAV sensor range intersects the ellipsoidal object [27] (Fig. 3, Table C1 in Appendix C).

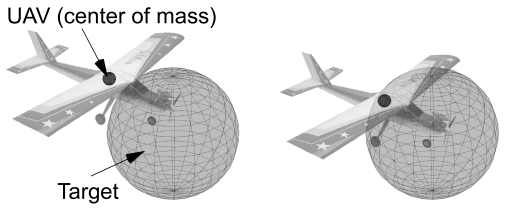
2. *Target Interception Scheme*

The UAV intercepts the target when its center of mass is contained within the ellipsoidal target [27] (Fig. 4, Table C1 in Appendix C).



a) b)

Fig. 3 Object detection scheme: a) no detection and b) detection.



a) b)

Fig. 4 Target interception scheme: a) no interception and b) interception.

The mission is deemed complete when all UAVs have intercepted the target.

3. *Unmanned Aerial Vehicle Communication Scheme*

Information on objects (obstacles and target) detected by fellow UAVs that are in communication range is also relayed to the UAV. Two UAVs can communicate when the ellipsoids modeling their respective communication range encompass the other UAV center of mass [28] (Fig. 5, Table C1 in Appendix C). A multihopping communication scheme [29] is used when more than two UAVs cooperate. Furthermore, an obstacle is no longer tracked; thus, its trajectory is no longer predicted if the obstacle exits all UAV sensor ranges. The effects of neglecting obstacle information outside all UAV sensor ranges will be studied in future work.

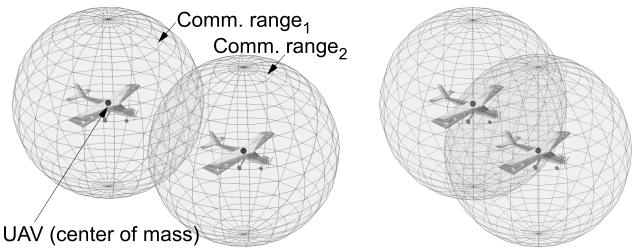
B. Trajectory Control Unit

The TCU for a single UAV is depicted in Fig. 6. This TCU is a receding horizon model-based predictive controller [23] that calculates the optimal UAV setpoints $\hat{\boldsymbol{u}}_{uv}(k)$ as a function of predicted obstacle, target, and UAV trajectories. The UAV setpoints are chosen to obey the mission objectives quantified by the objective function and constraints, defined in Sec. III.B.4.

1. *Obstacle Trajectory Prediction*

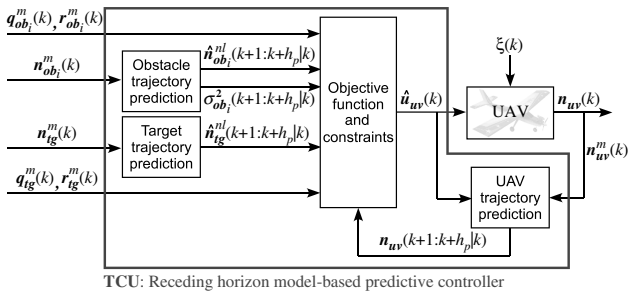
At each sample time, the UAV transmits the measured location (i.e., position obtained from a video camera images),

$$\boldsymbol{n}^m_{ob_i}(k) = [\,\mathrm{x}^m_{ob_i}(k) \quad \mathrm{y}^m_{ob_i}(k) \quad \mathrm{z}^m_{ob_i}(k)\,]^T \tag{33}$$



a) b)

Fig. 5 UAV communication scheme: a) no communication and b) communication.



TCU: Receding horizon model-based predictive controller

Fig. 6 TCU: model-based predictive controller.

along with the measured xyz semiaxes (i.e., size deduced from video camera images),

$$\boldsymbol{q}^{\,m}_{\boldsymbol{ob}_i}(k) = [\, a^m_{ob_i}(k) \quad b^m_{ob_i}(k) \quad c^m_{ob_i}(k)\,]^T \tag{34}$$

and the measured roll–pitch–yaw Euler rotation angles (i.e., orientation deduced from video camera images),

$$\boldsymbol{r}^{\,m}_{\boldsymbol{ob}_i}(k) = [\, \phi^m_{ob_i}(k) \quad \vartheta^m_{ob_i}(k) \quad \psi^m_{ob_i}(k)\,]^T \tag{35}$$

of its i known ellipsoidal obstacles to its TCU (Figs. 2 and 6). Using $\boldsymbol{n}^m_{\boldsymbol{ob}_i}(k)$, the TCU employs the trajectory prediction algorithm of Sec. II to estimate the states and to ultimately predict the obstacle trajectories. Assuming that the ith obstacle motion is governed by Eq. (26), its predicted optimal center of mass trajectory on h_p is calculated from Eq. (27) and yields $\hat{\boldsymbol{n}}^{\mathrm{nl}}_{\boldsymbol{ob}_i}(k+1\colon k+h_p|k)$. The quality of this trajectory is calculated from Eq. (31), producing $\boldsymbol{\sigma}^2_{\boldsymbol{ob}_i}(k+1\colon k+h_p|k)$. These quantities, along with the measurements $\boldsymbol{q}^m_{\boldsymbol{ob}_i}(k)$ and $\boldsymbol{r}^m_{\boldsymbol{ob}_i}(k)$, are then used to define the obstacle avoidance criterion detailed in Sec. III.B.4.

2. *Target Trajectory Prediction*

At each sample time, the UAV transmits the measured position $\boldsymbol{n}^m_{\boldsymbol{tg}}(k)$, the measured size $\boldsymbol{q}^m_{\boldsymbol{tg}}(k)$, and the orientation $\boldsymbol{r}^m_{\boldsymbol{tg}}(k)$ of the dynamic ellipsoidal target to its TCU (Figs. 2 and 6). Using $\boldsymbol{n}^m_{\boldsymbol{tg}}(k)$, the TCU then employs the trajectory prediction algorithm of Sec. II to estimate the states and to ultimately predict the target trajectory. Assuming that the target motion is governed by Eq. (26), the predicted optimal target center of mass trajectory on h_p is calculated from Eq. (27) and yields $\hat{\boldsymbol{n}}^{\mathrm{nl}}_{\boldsymbol{tg}}(k+1\colon k+h_p|k)$. This vector is integral to the target-tracking criterion, defined in Sec. III.B.4. As for the measurements $\boldsymbol{q}^m_{\boldsymbol{tg}}(k)$ and $\boldsymbol{r}^m_{\boldsymbol{tg}}(k)$, they are used to determine if the UAV has intercepted the target (see Sec. III.A.2).

3. *Unmanned Aerial Vehicle Trajectory Prediction*

An augmented EKF (AEKF) [23], as detailed in [30], is first used to estimate the optimal UAV states $\hat{\boldsymbol{x}}_{\boldsymbol{uv}}(k+1|k)$ from the measured UAV position $\boldsymbol{n}^m_{\boldsymbol{uv}}(k)$ and setpoints $\hat{\boldsymbol{u}}_{\boldsymbol{uv}}(k)$. The UAV motion model used for Kalman filtering is the same as in Fig. 1, where $\boldsymbol{G}(z)$ now models the closed-loop dynamics of the UAV (i.e., combined aircraft and autopilot dynamics). To compensate for unmeasured disturbances [$\boldsymbol{\xi}(k)$ in Fig. 6] due, for instance, to a constant wind flow acting upon the UAV, the UAV motion model $\boldsymbol{G}^d_{\boldsymbol{uv}}(z)$ is augmented by a disturbance model $\boldsymbol{G}^s_{\boldsymbol{uv}}(z)$ comprising a collection of integrators driven by white noises, as shown in [30]. The future UAV trajectory on h_p is then computed from the augmented UAV motion model and yields

$$\boldsymbol{n}_{\boldsymbol{uv}}(k+1\colon k+h_p|k) = f_4[\hat{\boldsymbol{x}}_{\boldsymbol{uv}}(k+1|k), \boldsymbol{u}_{\boldsymbol{uv}}(k+1\colon k+h_c|k)] \tag{36}$$

4. *Objective Function and Constraints*

At each sample time, the TCU calculates the optimal predicted UAV setpoint trajectory on the control horizon $\hat{\boldsymbol{u}}_{\boldsymbol{uv}}(k+1\colon k+h_c|k)$ that generates the optimal predicted UAV trajectory $\hat{\boldsymbol{n}}_{\boldsymbol{uv}}(k+1\colon k+h_p|k)$. This setpoint trajectory is obtained by minimizing an objective function while respecting a set of linear and nonlinear constraints. The objective function is constructed such that the UAV attains the mission objectives (i.e., the interception of a single moving ellipsoidal target while avoiding several moving ellipsoidal obstacles). The constraints ensure that the selected setpoints are confined within the UAV limits of operation.

a. Objective Function. The objective function

$$j(k) = j_u(k) + j_{ob}(k) + j_{tg}(k) \tag{37}$$

for a single UAV is a summation of a setpoint weighting criterion, an obstacle avoidance criterion, and a target-tracking criterion, respectively.

The setpoint weighting criterion

$$j_u(k) = \Delta\boldsymbol{u}_{\boldsymbol{uv}}(k+1\colon k+h_c|k)^T \boldsymbol{\Lambda}\, \Delta\boldsymbol{u}_{\boldsymbol{uv}}(k+1\colon k+h_c|k) \tag{38}$$

seeks to minimize UAV autopilot setpoint increments on h_c. In Eq. (38), the vector of setpoint increments are weighted according to $\boldsymbol{\Lambda}$, a $3h_c \times 3h_c$ diagonal matrix. Setpoints are thus assumed uncorrelated, meaning that simultaneous variations of all setpoints are possible (see Sec. II.A.1).

The obstacle avoidance criterion seeks to distance the UAV from predicted obstacle trajectories: UAVs are considered small (i.e., zero-dimensional) in comparison with obstacles. This is accomplished by first evaluating, at each prediction time τ on h_p, the probability of collision of the UAV with a known location [$\boldsymbol{n}_{uv}(k+\tau|k)$ of Eq. (36)] with the i ellipsoidal obstacle of known size and orientation [$\boldsymbol{q}^m_{\boldsymbol{ob}_i}(k)$ of Eq. (34) and $\boldsymbol{r}^m_{\boldsymbol{ob}_i}(k)$ of Eq. (35)] but with an uncertain center of mass location $\boldsymbol{n}_{\boldsymbol{ob}_i}(k+\tau)$. What is known is that the location of the i obstacle center of mass is normally distributed [31,32] with mean $\hat{\boldsymbol{n}}^{\mathrm{nl}}_{\boldsymbol{ob}_i}(k+\tau|k)$ and variance $\boldsymbol{\sigma}^2_{\boldsymbol{ob}_i}(k+\tau|k)$, yielding the PDF

$$\begin{aligned} f_{ob_{i,\tau}}(\boldsymbol{n}) = {} & \frac{1}{\sqrt{(2\pi)^3 \mid \boldsymbol{\sigma}^2_{\boldsymbol{ob}_i}(k+\tau|k) \mid}} \\ & \times \exp\left\{-\frac{1}{2}[\boldsymbol{n} - \hat{\boldsymbol{n}}^{\mathrm{nl}}_{\boldsymbol{ob}_i}(k+\tau|k)]^T \boldsymbol{\sigma}^2_{\boldsymbol{ob}_i}(k+\tau|k)^{-1} \right. \\ & \left. \times [\boldsymbol{n} - \hat{\boldsymbol{n}}^{\mathrm{nl}}_{\boldsymbol{ob}_i}(k+\tau|k)]\right\} \end{aligned} \tag{39}$$

where $\boldsymbol{n} = [\,\mathsf{x} \quad \mathsf{y} \quad \mathsf{z}\,]^T$ is a random position in space.

Thus, knowing that the probability for a point $\boldsymbol{n}$ to reside within the ith obstacle, centered at an arbitrary position $\boldsymbol{n}_{\boldsymbol{ob}_i}(k+\tau)$, is given by the indicator function

$$\begin{aligned} & g_{ob_{i,\tau}}[\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{ob}_i}(k+\tau)] \\ & = 1\begin{cases} 1 & [\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{ob}_i}(k+\tau)]^T \boldsymbol{R}_{\boldsymbol{ob}_i} \boldsymbol{Q}_{\boldsymbol{ob}_i} \boldsymbol{R}^T_{\boldsymbol{ob}_i} [\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{ob}_i}(k+\tau)] \le 1 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{40}$$

Then, the probability for a point $\boldsymbol{n}$ to reside within this obstacle, knowing that the obstacle center of mass is statistically centered at $\hat{\boldsymbol{n}}^{\mathrm{nl}}_{\boldsymbol{ob}_i}(k+\tau|k)$ and distributed, as shown in Eq. (39), is

$$\begin{aligned} p_{i,\tau}(\boldsymbol{n}) &= \int_{\boldsymbol{n}_{\boldsymbol{ob}_{i,\tau}}} f_{ob_{i,\tau}}(\boldsymbol{n}_{\boldsymbol{ob}_{i,\tau}}) g_{ob_{i,\tau}}(\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{ob}_{i,\tau}}) d\boldsymbol{n}_{\boldsymbol{ob}_{i,\tau}} \\ &= f_{ob_{i,\tau}}(\boldsymbol{n}) \cdot g_{ob_{i,\tau}}(\boldsymbol{n}) \end{aligned} \tag{41}$$

where $\boldsymbol{n}_{\boldsymbol{ob}_{i,\tau}}$ is the short form for $\boldsymbol{n}_{\boldsymbol{ob}_i}(k+\tau)$ and where $*$ denotes the linear convolution operator.

The complete obstacle avoidance criterion for a single UAV is then the probability of collision between the UAV and all known ellipsoidal obstacles on h_p. Using probability theory, one obtains

$$j_{ob}(k) = \kappa_{ob} \sum_{\tau=1}^{h_p} p(e_\tau) \tag{42}$$

where e_τ denotes the event that a collision occurs at prediction time τ, such that

$$p(e_\tau) = \begin{cases} m_\tau & \tau = 1 \\ m_\tau \prod_{n=1}^{\tau-1} [1 - p(e_n)] & \text{otherwise} \end{cases} \tag{43}$$

where

$$m_\tau = \sum_{i \in \mathbb{A}} \left\{ p_{i,\tau}(\boldsymbol{n}) \prod_{j \in \mathbb{B}} [1 - p_{j,\tau}(\boldsymbol{n})] \right\} \Bigg|_{\boldsymbol{n} = \boldsymbol{n}_{uv}(k+\tau|k)} \tag{44}$$

In Eq. (44), $\mathbb{A}$ is the set of obstacles known to the UAV, whereas $\mathbb{B}$ is a subset of $\mathbb{A}$ containing all elements in $\mathbb{A}$ minus the ith obstacle. Finally, the constant κ_{ob} in Eq. (42) is chosen such that the obstacle

avoidance criterion takes precedence over the setpoint weighting criterion and the target-tracking criterion.

An example of a two-dimensional probability of collision function for an elliptical obstacle calculated from the statistical distribution of its center of mass position is illustrated in Appendix C. Details on how the convolution integral of Eq. (41) is evaluated in the simulations of Sec. IV.B are also provided in Appendix C.

There exists a wide range of target-tracking strategies, each tailored to the mission objective and the specific target behavior. Examples of target-tracking methods include strategies seeking to minimize the UAV energy upon interception of the target, strategies intended to minimize the UAV-to-target arrival time, and strategies seeking to minimize the distance traveled by the UAV while tracking the target [33,34]. The target-tracking criterion adopted herein

$$j_{tg}(k) = \sum_{\tau=1}^{h_p} \kappa_{tg}(\tau) \| \boldsymbol{n}_{uv}(k+\tau|k) - \hat{\boldsymbol{n}}_{tg}^{nl}(k+\tau|k) \|_2 \tag{45}$$

seeks to minimize the two-norm distance $\| \bullet \|_2$ between the predicted UAV position and predicted optimal target position along h_p. This criterion attempts to match the UAV heading to the target heading at the time of interception. In Eq. (45), the constant $\kappa_{tg}(\tau)$ weights the UAV-to-target distance along the trajectory.

b. Constraints. Numerical minimization of $j(k)$ is subject to constraints. First, the UAV setpoint increments on h_c and UAV setpoints on h_c must be limited to ensure the safe operation of the UAV. Thus,

$$\Delta \boldsymbol{u}_{uv}^{\min} \le \Delta \boldsymbol{u}_{uv}(k+1{:}\ k+h_c|k) \le \Delta \boldsymbol{u}_{uv}^{\max} \tag{46}$$

$$\boldsymbol{u}_{uv}^{\min} \le \boldsymbol{u}_{uv}(k+1{:}\ k+h_c|k) \le \boldsymbol{u}_{uv}^{\max} \tag{47}$$

Second, the predicted UAV trajectory $\boldsymbol{n}_{uv}(k+1{:}\ k+h_p|k)$ should be contained within the span of the aircraft sensor range [28]:

$$\boldsymbol{n}_{uv}(k+\tau|k) \boldsymbol{R}_{sn} \boldsymbol{Q}_{sn} \boldsymbol{R}_{sn}^T \boldsymbol{n}_{uv}(k+\tau|k)^T \le 0 \tag{48}$$

where $\tau \in \{1, \ldots, h_p\}$, thus ensuring that a predicted UAV trajectory does not intersect an undetected obstacle. Third, the UAV speed $\mathcal{S}_{uv}(k)$, where the flat speed $\mathsf{s}_{uv}(k)$ is the projection of $\mathcal{S}_{uv}(k)$ onto the xy plane, must also be constrained along h_p. The minimum UAV speed is chosen to obey requirements for a safe flight, whereas the maximum UAV speed must respect the physical limitations of the aircraft. Thus,

$$\mathcal{S}_{uv}^{\min} \le \mathcal{S}_{uv}(k+1{:}\ k+h_p|k) \le \mathcal{S}_{uv}^{\max} \tag{49}$$

where at prediction time $k+\tau$,

$$\mathcal{S}_{uv}(k+\tau|k) = \sqrt{[\mathsf{s}_{uv}(k+\tau|k)]^2 + \left(\frac{\Delta \mathsf{z}_{uv}(k+\tau|k)}{t_s}\right)^2} \tag{50}$$

In Eq. (50), the UAV speed is the hypotenuse of the right-angled triangle formed by the UAV flat speed and altitude variation during the sampling period t_s. Finally, an additional constraint can be added to ensure that the probability of collision between the UAV and an obstacle is less than a preset threshold at each prediction time τ on h_p:

$$p_{i,\tau}[\boldsymbol{n}_{uv}(k+\tau|k)] \le p_{i,\tau}^{\max} \tag{51}$$

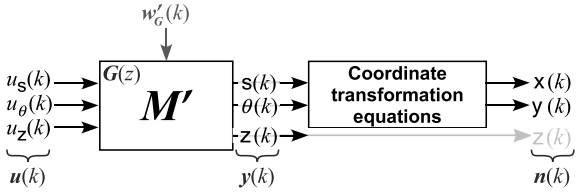
for $0 \le p_{i,\tau}^{\max} \le 1$ and $\tau \in \{1, \ldots, h_p\}$.

IV. Simulations

A. Trajectory Prediction Test Cases (Obstacle or Target)

Three test cases involving the trajectory prediction algorithm of Sec. II are now shown. In test case 1, the EKF of Sec. II.A is used to estimate the states of a controlled dynamic object detected by a UAV. In test case 2, the algorithm of Sec. II.B is used to predict two object trajectories from the EKF state estimates of test case 1. In test case 3, the effect of redundant object position measurements on a predicted object trajectory, occurring when two communicating UAVs detect the same object, is demonstrated.

• Simulated object dynamics:



➢ $\boldsymbol{u}(k)$ is the random walk sequence:

$$\boldsymbol{u}(k) = \boldsymbol{u}(k-1) + \boldsymbol{w}'_{\boldsymbol{u}}(k)$$

which undergoes large step changes at specific sample times.

$\boldsymbol{W}'_{\boldsymbol{u}}$ is a diagonal matrix:

$dg\{\boldsymbol{W}'_{\boldsymbol{u}}\} = [0.4\ \text{km}^2/\text{h}^2 \ \ 0.33^{\circ 2} \ \ 2.7\ \text{m}^2]$

➢ $\boldsymbol{w}'_G(k) \sim N(\boldsymbol{0}, \boldsymbol{W}'_G)$

$\boldsymbol{W}'_G$ is a diagonal matrix:

$dg\{\boldsymbol{W}'_G\} = [\boldsymbol{W}'_{G_{ss}} \ \boldsymbol{W}'_{G_{z\theta}} \ \boldsymbol{W}'_{G_{\theta\theta}} \ \boldsymbol{W}'_{G_{sz}} \ \boldsymbol{W}'_{G_{\theta z}} \ \boldsymbol{W}'_{G_{zz}}]$

$\boldsymbol{W}'_G$ diagonal elements are 1 unit^2 except for $dg\{\boldsymbol{W}'_{G_{\theta\theta}}\} = [3.3^{\circ 2} \ \ 3.3^{\circ 2}]$

• Noisy object position measurement:

$$\boldsymbol{n}^m(k) = \boldsymbol{n}(k) + \boldsymbol{v}'(k)$$

$\boldsymbol{v}'(k) = N(\boldsymbol{0}, \boldsymbol{V}')$ where $\boldsymbol{V}'$ is a diagonal matrix:

$dg\{\boldsymbol{V}'\} = [30\ \text{m}^2 \ \ 30\ \text{m}^2 \ \ 30\ \text{m}^2]$

Fig. 7 Simulated object dynamics.

1. Test Case 1: State Estimation

A UAV detects and classifies a stabilized moving object, for which the simulated dynamics are shown in Fig. 7. The UAV measures its position $\boldsymbol{n}^m(k)$** then identifies the dynamical model M, shown in Fig. 8, that best describes the object dynamic behavior. It is assumed that the UAV correctly identifies the object such that $M' = M = \boldsymbol{G}(z)$, where $\boldsymbol{G}(z)$ is the closed-loop discrete-time transfer function:

$$\boldsymbol{G}(z) = \begin{bmatrix} G_{ss}(z) & 0 & G_{sz}(z) \\ 0 & G_{\theta\theta}(z) & 0 \\ G_{zs}(z) & -G_{z\theta}(z) & G_{zz}(z) \end{bmatrix} \tag{52}$$

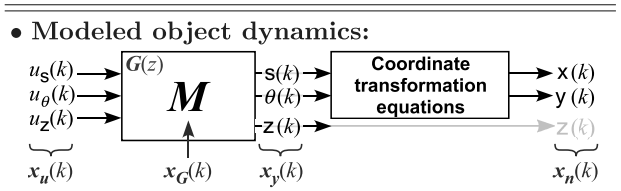
with

$$G_{ii}(z) = \mathcal{Z}\left\{\frac{1}{(1+4s)^2}\right\}\Bigg|_{ii \in \{ss, \theta\theta, zz\}}$$
$$G_{ij}(z) = \mathcal{Z}\left\{\frac{-3s}{(1+8s)^2}\right\}\Bigg|_{ij \in \{sz, z\theta, zs\}} \tag{53}$$

for which the continuous-time transfer functions (s is the complex argument of the Laplace transform) are discretized using a zero-order holder with sampling period t_s. Differences between the simulated and modeled object dynamics stem from the stochastic perturbations $\boldsymbol{w}'_G(k)$ on the simulated states of M' (e.g., wind affecting object trajectory).

The goal is then to employ the EKF of Sec. II.A to estimate the simulated object states from the modeled object dynamics in Fig. 8 (also in Fig. 1) and from the noisy object position measurement

**The UAV also measures the object size $\boldsymbol{q}^m(k)$ and orientation $\boldsymbol{r}^m(k)$; however, these measurements are not needed in this test case.



- **EKF process noise covariance matrix:**

$$W^l = \begin{bmatrix} W_u & 0 & 0 & 0 \\ 0 & W_G & 0 & 0 \\ 0 & 0 & C_G W_G C_G^T & 0 \\ 0 & 0 & 0 & W_n \end{bmatrix}$$

W_u, W_G, W_n are diagonal matrices:

$dg\{W_u\} = dg\{W'_u\}$

$dg\{W_G\} = dg\{W'_G\}$

$dg\{W_n\} = [0.1\ \mathrm{m}^2\ \ 0.1\ \mathrm{m}^2]$

- **EKF measurement noise covariance matrix:**

V is a diagonal matrix: $dg\{V\} = dg\{V'\}$

Fig. 8 Estimated object dynamics.

$\boldsymbol{n}^m(k)$ in Fig. 7. The states estimated by the EKF [$\boldsymbol{x}(k)$ in Eq. (3)] are the object setpoints $\boldsymbol{x_u}(k)$, the internal states $\boldsymbol{x_G}(k)$ of $\boldsymbol{G}(z)$, the outputs $\boldsymbol{x_y}(k)$ of $\boldsymbol{G}(z)$, and the xy-planar position $\boldsymbol{x_n}(k)$. The noise covariance matrices $\boldsymbol{W}^l$ and $\boldsymbol{V}$ used to tune the EKF in this simulation are listed in Fig. 8.

The results are illustrated in Fig. 9, where simulated object states [i.e., $\boldsymbol{u}(k)$, $\boldsymbol{y}(k)$, and $\boldsymbol{n}(k)$ of the simulated object dynamics shown in Fig. 7] are depicted as thick solid lines, and estimated object states [i.e., $\boldsymbol{x_u}(k)$, $\boldsymbol{x_y}(k)$, and $\boldsymbol{x_n}(k)$ of the modeled object dynamics shown in Fig. 8] are represented by medium solid lines. The simulated and estimated internal states of $\boldsymbol{G}(z)$ are not shown. The noisy object position measurements $\boldsymbol{n}^m(k)$ are additionally shown as fine solid lines in Fig. 9c. The EKF manages to filter the UAV sensor noises (the estimated state curves are smooth in comparison with the measurement curves) and estimates adequate object positions. In spite of measurement and model uncertainties, the EKF also succeeds in estimating accurate outputs of $\boldsymbol{G}(z)$ (Fig. 9b). Finally, when the simulated setpoints behave as modeled, the setpoint estimates closely track the simulated object setpoints. However, when the simulated setpoints undergo large step changes, the setpoint estimates converge slowly to the new simulated setpoint sequence. For faster convergence to the new simulated setpoint sequence, one would choose $\boldsymbol{W_u} > \boldsymbol{W'_u}$.

2. *Test Case 2: Trajectory Prediction*

Consider the object detected by a UAV of test case 1, for which the simulated dynamics and modeled dynamics are depicted in Figs. 7 and 8, respectively, and for which $\boldsymbol{G}(z)$ is Eqs. (52) and (53). The goal is to employ the algorithm of Sec. II.B to predict object trajectories (i.e., optimal center of mass trajectory and position prediction error variance) from the EKF state estimates of test case 1. Figure 10 illustrates two such predicted trajectories calculated from the EKF state estimates at times $t = 122$ s and $t = 50$ s, respectively. Depicted in Fig. 10 are the predicted optimal object trajectories (fine solid lines with circular markers) computed from Eq. (27), the position prediction error variances (dotted–dashed lines defining the $\pm 3\sigma$ uncertainty boundaries) corresponding to the diagonal elements of Eq. (31), and the simulated object trajectories (thick solid lines) calculated from the simulated object dynamics shown in Fig. 8.

Between times $t = 122$–140 s (Fig. 10a), the simulated trajectory lies within the $\pm 3\sigma$ uncertainty model. During this time interval, the simulated object setpoints behave like the random walk setpoint model used for estimation and prediction. However, between times $t = 50$–68 s (Fig. 10b), the object exhibits heading and altitude setpoint variations of magnitudes much larger than assumed by the variances of the setpoint random walk generator. Consequently, the simulated trajectory evolves outside the $\pm 3\sigma$ uncertainty boundaries, particularly toward the end of the predicted trajectory. This is an obvious limitation of the predictor performance (i.e., when the object behavior differs greatly from its modeled behavior). However, the trajectory prediction scheme will be executed at subsequent sample times, thus adapting to the changing behavior of the observed object.

3. *Test Case 3: Cooperation*

The EKF of Sec. II.A can exploit redundant object position measurements through Eq. (20) [or equivalently, through Eq. (21)] to

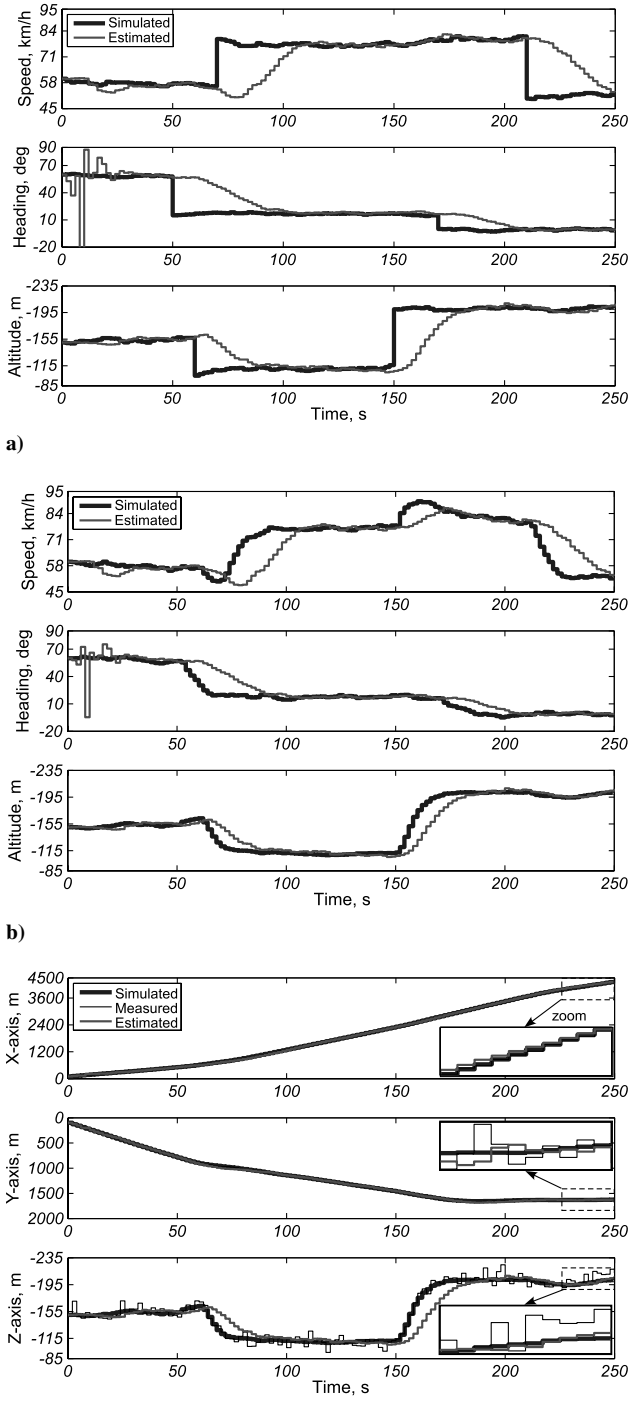
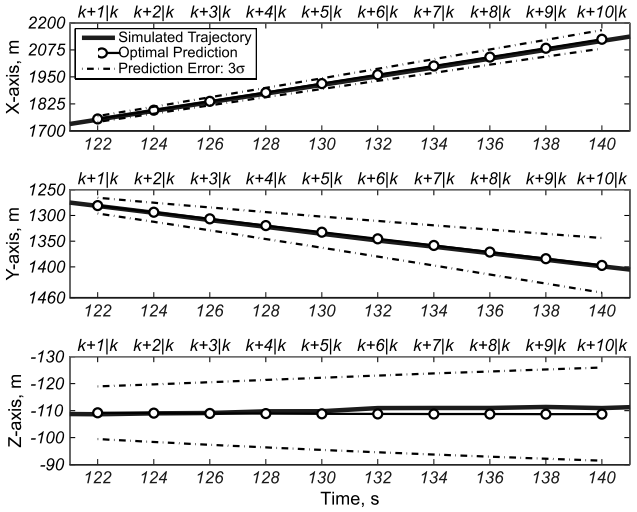
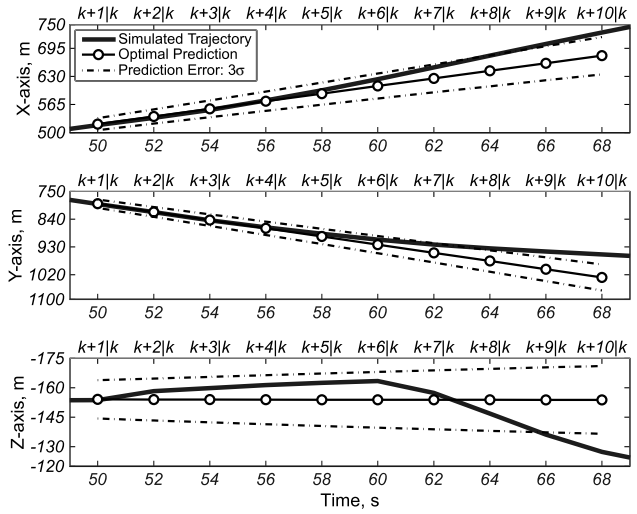


Fig. 9 Object state estimation: a) setpoints, b) outputs of $G(z)$, and c) positions.



a)



b)

Fig. 10 Object trajectory prediction: a) good prediction and b) degraded prediction.

reduce the variance on the object position measurements [as shown in Eq. (24)] and, therefore, on its estimated states [i.e., through $\boldsymbol{P}(k+1|k)$]. A reduction in $\boldsymbol{P}(k+1|k)$ necessarily diminishes $\boldsymbol{\sigma}^2(k+1\colon k+h_p|k)$ of Eq. (31), since $\boldsymbol{P}(k+1|k)$ is the starting point for all prediction error covariance calculations. In light of this observation, let us consider once again the object of test case 1, where it is now assumed that its position is measured by two communicating UAVs (forming the *i*th subfleet within the collective) equipped with fully functional independent identical sensors.

Figure 11 presents a comparison of the position prediction error (standard deviation ratio) at prediction times $t = 122$–140 s for one object position measurement (results of Fig. 10a) versus two object position measurements. The position prediction error calculated at the first prediction time ($t = 122$ s) is entirely dependent on $\boldsymbol{P}(k+1|k)$ [see Eq. (31)]. Thus, should the EKF consider uncertainty on the object measurements only [$\boldsymbol{w}(k) = \boldsymbol{0}$ in Eq. (1)], one would expect a ratio of $1/\sqrt{2}$ for all axes at $t = 122$ s, since Eq. (24) reduces to

$$\boldsymbol{V}_{i_{n\to 1}} = \boldsymbol{V}_{ij}/2 \tag{54}$$

for two independent identically distributed measurements. However, since the EKF considers both the dynamical model and measurements of the object when computing its optimal states, the standard deviation ratio for all axes at prediction time $t = 122$ s is greater than $1/\sqrt{2}$ (see Fig. 11). Moreover, as $\tau \to \infty$, the standard

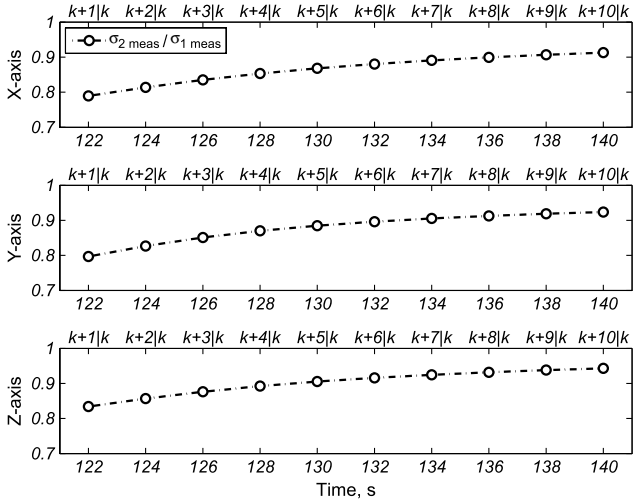


Fig. 11 Position prediction error ratio (one vs two measurements).

deviation ratio at prediction time $k+\tau|k$ tends toward unity. The advantage of having an extra object measurement therefore decreases as one predicts further into the future. This is to be expected, since the effect of $\boldsymbol{P}(k+1|k)$ on the position prediction error is attenuated over time [see Eq. (31)].

B. Path Planner Test Cases

Two test cases involving the path planner described in Sec. III are now shown. These test cases demonstrate the performance of the path planner when UAVs accomplish their mission, with and without cooperation. The mission scenario for both test cases involves two UAVs, two ellipsoidal obstacles, and one spherical target. Only five objects are considered in the following test cases in order to facilitate their understanding. There are, however, no limitations to the number of objects that can be considered in the path planner. In both test cases, the sampling period, prediction horizon, and control horizon are, respectively,

$$t_s = 2 \text{ s}; \qquad h_p = 10; \qquad h_c = 3 \tag{55}$$

The simulated UAV dynamics and modeled UAV dynamics are

$$\boldsymbol{y}_{\boldsymbol{uv}_i}(z) = \boldsymbol{G}_{\boldsymbol{uv}_i}(z)\boldsymbol{u}_{\boldsymbol{uv}_i}(z)|_{i\in\{1,2\}} \tag{56}$$

where

$$\boldsymbol{G}_{\boldsymbol{uv}_i}(z) = \begin{bmatrix} G_{uv_{ss}}(z) & 0 & G_{uv_{sz}}(z) \\ 0 & G_{uv_{\theta\theta}}(z) & 0 \\ G_{uv_{zs}}(z) & G_{uv_{z\theta}}(z) & G_{uv_{zz}}(z) \end{bmatrix} \tag{57}$$

with

$$\begin{aligned}
G_{uv_{ss}}(z) &= \mathcal{Z}\left\{\frac{1}{6s+1}\right\} \\
G_{uv_{sz}}(z) &= \mathcal{Z}\left\{\frac{-4.75s}{0.45s^2+4.62s+1}\right\} \\
G_{uv_{\theta\theta}}(z) &= \mathcal{Z}\left\{\frac{1}{3.78s+1}\right\} \\
G_{uv_{zs}}(z) &= \mathcal{Z}\left\{\frac{-0.37s}{37.56s^2+2.85s+1}\right\} \\
G_{uv_{z\theta}}(z) &= \mathcal{Z}\left\{\frac{-0.09s}{13.68s^2+2.02s+1}\right\} \\
G_{uv_{zz}}(z) &= \mathcal{Z}\left\{\frac{13.77s+1}{14.29s^2+11.67s+1}\right\}
\end{aligned} \tag{58}$$

This model was obtained through identification of a Hangar^{TM}Xtra Easy2 remote-control plane equipped with an MP2028^g autopilot

fabricated by MicroPilot®. The initial UAV positions in North–East–Down (NED) coordinates are

$$\begin{aligned}\boldsymbol{n}_{\boldsymbol{uv}_1}(0) &= [\,900\text{ m}\quad 600\text{ m}\quad -400\text{ m}\,]^T\\ \boldsymbol{n}_{\boldsymbol{uv}_2}(0) &= [\,-400\text{ m}\quad -500\text{ m}\quad -400\text{ m}\,]^T\end{aligned}\tag{59}$$

whereas the initial UAV setpoints and outputs are

$$\begin{aligned}\boldsymbol{u}_{\boldsymbol{uv}_1}(0) = \boldsymbol{y}_{\boldsymbol{uv}_1}(0) &= [\,90\text{ km/h}\quad -35^\circ\quad -400\text{ m}\,]^T\\ \boldsymbol{u}_{\boldsymbol{uv}_2}(0) = \boldsymbol{y}_{\boldsymbol{uv}_2}(0) &= [\,96\text{ km/h}\quad 0^\circ\quad -400\text{ m}\,]^T\end{aligned}\tag{60}$$

UAVs are also surrounded by spherical sensor ranges and communication ranges (test case 2 only) with xyz semiaxes

$$\begin{aligned}\boldsymbol{q}_{\boldsymbol{sr}_i} &= [\,600\text{ m}\quad 600\text{ m}\quad 600\text{ m}\,]^T \boldsymbol{q}_{\boldsymbol{cm}_i}\\ &= [\,3000\text{ m}\quad 3000\text{ m}\quad 3000\text{ m}\,]^T\Big|_{i\in\{1,2\}}\end{aligned}\tag{61}$$

, respectively. Moreover, the diagonal of the weight matrix $\boldsymbol{\Lambda}$ in Eq. (38) is the vector

$$\lambda_i = [\,1\times 10^{-2}\quad 2\times 10^{-3}\quad 1\times 10^{-4}\,]^T|_{i\in\{1,2\}}\tag{62}$$

repeated on the control horizon, and the UAV trajectories are subject to the constraints

$$\begin{aligned}\Delta\boldsymbol{u}_{\boldsymbol{uv}}^{\min} &= [\,-40\text{ km/h}\quad -10^\circ\quad -6\text{ m}\,]^T\\ \Delta\boldsymbol{u}_{\boldsymbol{uv}}^{\max} &= [\,40\text{ km/h}\quad 10^\circ\quad 12\text{ m}\,]^T\end{aligned}\tag{63}$$

$$\begin{aligned}\boldsymbol{u}_{\boldsymbol{uv}}^{\min} &= [\,76\text{ km/h}\quad -180^\circ\quad -9970\text{ m}\,]^T\\ \boldsymbol{u}_{\boldsymbol{uv}}^{\max} &= [\,140\text{ km/h}\quad 180^\circ\quad -70\text{ m}\,]^T\end{aligned}\tag{64}$$

$$\mathcal{S}_{uv}^{\min} = 79\text{ km/h};\qquad \mathcal{S}_{uv}^{\max} = 142\text{ km/h}\tag{65}$$

Finally, the AEKF tuning parameters employed to estimate the UAV states are listed in Fig. 12.

The obstacles are modeled as ellipsoids of sizes and orientations

$$\begin{aligned}\boldsymbol{q}_{\boldsymbol{ob}_1} &= [\,300\text{ m}\quad 225\text{ m}\quad 300\text{ m}\,]^T\\ \boldsymbol{q}_{\boldsymbol{ob}_2} &= [\,260\text{ m}\quad 200\text{ m}\quad 260\text{ m}\,]^T\end{aligned}\tag{66}$$

- **AEKF process noise covariance matrix:**

$$\boldsymbol{W}^l = \begin{bmatrix}\boldsymbol{W}_{\boldsymbol{G}^d} & 0 & 0\\ 0 & \boldsymbol{W}_{\boldsymbol{n}} & 0\\ 0 & 0 & \boldsymbol{W}_{\boldsymbol{G}^s}\end{bmatrix}$$

  $\boldsymbol{W}_{\boldsymbol{G}^d}$, $\boldsymbol{W}_{\boldsymbol{n}}$, $\boldsymbol{W}_{\boldsymbol{G}^s}$ are diagonal matrices:

  $dg\{\boldsymbol{W}_{\boldsymbol{G}^d}\} = [\boldsymbol{W}_{G_{\mathrm{ss}}}\ \boldsymbol{W}_{G_{\mathrm{z}\theta}}\ \boldsymbol{W}_{G_{\theta\theta}}\ \boldsymbol{W}_{G_{\mathrm{sz}}}\ \boldsymbol{W}_{G_{\theta\mathrm{z}}}\ \boldsymbol{W}_{G_{\mathrm{zz}}}]$

  $\boldsymbol{W}_{\boldsymbol{G}^d}$ diagonal elements are 0.01 unit^2

  except for $\boldsymbol{W}_{G_{\theta\theta}} = 0.03^{\circ 2}$

  $dg\{\boldsymbol{W}_{\boldsymbol{n}}\} = [0.001\text{ m}^2\ \ 0.001\text{ m}^2]$

  $dg\{\boldsymbol{W}_{\boldsymbol{G}^s}\} = [\boldsymbol{W}_{G_{\mathrm{x}\xi}}\ \ \boldsymbol{W}_{G_{\mathrm{y}\xi}}\ \ \boldsymbol{W}_{G_{\mathrm{z}\xi}}]$

  $\boldsymbol{W}_{\boldsymbol{G}^s}$ diagonal elements are 1×10^{-6} unit^2

- **AEKF measurement noise covariance matrix:**

  $\boldsymbol{V}$ is a diagonal matrix:

  $dg\{\boldsymbol{V}\} = [1\text{ m}^2\ \ 1\text{ m}^2\ \ 1\text{ m}^2]$

Fig. 12 AEKF tuning parameters. The AEKF tuning set is chosen equal to the simulated UAV dynamics.

$$\boldsymbol{r}_{\boldsymbol{ob}_1} = [\,-45^\circ\quad 0^\circ\quad -45^\circ\,]^T;\qquad \boldsymbol{r}_{\boldsymbol{ob}_2} = [\,0^\circ\quad -45^\circ\quad 0^\circ\,]^T\tag{67}$$

Their simulated dynamics and dynamic model are given by

$$\boldsymbol{y}_{\boldsymbol{ob}_i}(z) = \boldsymbol{G}_{\boldsymbol{ob}_i}(z)\boldsymbol{u}_{\boldsymbol{ob}_i}(z)|_{i\in\{1,2\}}\tag{68}$$

where

$$\boldsymbol{G}_{\boldsymbol{ob}_i}(z) = \begin{bmatrix} G_{ob_{\mathrm{ss}}}(z) & 0 & G_{ob_{\mathrm{sz}}}(z)\\ 0 & G_{ob_{\theta\theta}}(z) & 0\\ G_{ob_{\mathrm{zs}}}(z) & -G_{ob_{\mathrm{z}\theta}}(z) & G_{ob_{\mathrm{zz}}}(z)\end{bmatrix}\tag{69}$$

with

$$\begin{aligned}G_{ob_{ii}}(z) &= \mathcal{Z}\left\{\frac{1}{(4s+1)^2}\right\}\Big|_{ii\in\{\mathrm{ss},\theta\theta,\mathrm{zz}\}}\\ G_{ob_{ij}}(z) &= \mathcal{Z}\left\{\frac{-3s}{(8s+1)^2}\right\}\Big|_{ij\in\{\mathrm{sz},\mathrm{zs},\mathrm{z}\theta\}}\end{aligned}\tag{70}$$

The initial obstacle positions in NED coordinates are

$$\begin{aligned}\boldsymbol{n}_{\boldsymbol{ob}_1}(0) &= [\,2700\text{ m}\quad 700\text{ m}\quad -400\text{ m}\,]^T\\ \boldsymbol{n}_{\boldsymbol{ob}_2}(0) &= [\,6000\text{ m}\quad 100\text{ m}\quad -400\text{ m}\,]^T\end{aligned}\tag{71}$$

The obstacle setpoints are initially set to

$$\begin{aligned}\boldsymbol{u}_{\boldsymbol{ob}_1} &= [\,65\text{ km/h}\quad -110^\circ\quad -400\text{ m}\,]^T\\ \boldsymbol{u}_{\boldsymbol{ob}_2} &= [\,40\text{ km/h}\quad -140^\circ\quad -400\text{ m}\,]^T\end{aligned}\tag{72}$$

and modified at sample time $k = 20$ to

$$\begin{aligned}\boldsymbol{u}_{\boldsymbol{ob}_1} &= [\,70\text{ km/h}\quad -100^\circ\quad -400\text{ m}\,]^T\\ \boldsymbol{u}_{\boldsymbol{ob}_2} &= [\,40\text{ km/h}\quad -140^\circ\quad -400\text{ m}\,]^T\end{aligned}\tag{73}$$

Moreover, UAV measurements of the obstacle sizes and rotations are assumed exact and time invariant throughout the simulation: $\boldsymbol{q}_{\boldsymbol{ob}_i}^m(k) = \boldsymbol{q}_{\boldsymbol{ob}_i}|_{i\in\{1,2\}}$, $\boldsymbol{r}_{\boldsymbol{ob}_i}^m(k) = \boldsymbol{r}_{\boldsymbol{ob}_i}|_{i\in\{1,2\}}$. In addition, the constant $\kappa_{ob} = j_u^{\max} + j_{tg}^{\max} = 4.7\times 10^4$ in Eq. (42), and $p_{i,\tau}^{\max} = 0.1$ in Eq. (51). Finally, the EKF tuning parameters employed in obstacle trajectory predictions are those of Fig. 8.

The target is a sphere with xyz semiaxes

$$\boldsymbol{q}_{\boldsymbol{tg}} = [\,50\text{ m}\quad 50\text{ m}\quad 50\text{ m}\,]^T\tag{74}$$

for which the simulated dynamics and dynamic model are

$$\boldsymbol{y}_{\boldsymbol{tg}}(z) = \boldsymbol{G}_{\boldsymbol{tg}}(z)\boldsymbol{u}_{\boldsymbol{tg}}(z)\tag{75}$$

where

$$\boldsymbol{G}_{\boldsymbol{tg}}(z) = \begin{bmatrix} G_{tg_{\mathrm{ss}}}(z) & 0 & G_{tg_{\mathrm{sz}}}(z)\\ 0 & G_{tg_{\theta\theta}}(z) & 0\\ G_{tg_{\mathrm{zs}}}(z) & -G_{tg_{\mathrm{z}\theta}}(z) & G_{tg_{\mathrm{zz}}}(z)\end{bmatrix}\tag{76}$$

with

$$\begin{aligned}G_{tg_{ii}}(z) &= \mathcal{Z}\left\{\frac{1}{(5s+1)^2}\right\}\Big|_{ii\in\{\mathrm{ss},\theta\theta,\mathrm{zz}\}}\\ G_{tg_{ij}}(z) &= \mathcal{Z}\left\{\frac{-5s}{(8s+1)^2}\right\}\Big|_{ij\in\{\mathrm{sz},\mathrm{zs},\mathrm{z}\theta\}}\end{aligned}\tag{77}$$

The initial target position in NED coordinates is

$$\boldsymbol{n}_{\boldsymbol{tg}}(0) = [\,5500\text{ m}\quad -550\text{ m}\quad -400\text{ m}\,]^T\tag{78}$$

The target setpoints are initially set to

$$\boldsymbol{u}_{\boldsymbol{tg}} = [\,20\text{ km/h}\quad -60^\circ\quad -400\text{ m}\,]^T\tag{79}$$

and modified at sample time $k = 35$ to

$$\boldsymbol{u}_{tg} = [\,20\ \text{km/h} \quad -60^\circ \quad -410\ \text{m}\,]^T \tag{80}$$

Moreover, UAV measurements of the target size are assumed exact and time invariant throughout the simulation: $\boldsymbol{q}^m_{tg}(k) = \boldsymbol{q}_{tg}$. The constant $\kappa_{tg}(\tau)$ in Eq. (45) is also selected such that $\kappa_{tg}(\tau) = 1, \forall\ \tau$. Finally, the EKF tuning parameters employed in target trajectory predictions are those of Fig. 8.

The optimization problem is solved on a PC using the fmincon optimizer in MATLAB R2008. The average runtime needed by fmincon to solve the optimization problem is 1.4 s per UAV per sample time. A similar optimization problem has been solved using the Numerical Algorithms Group optimizer in C++® and has succeeded in reducing the runtime by a factor of 10.

1. *Test Case 1: No Cooperation Between Unmanned Aerial Vehicles*

Figure 13 illustrates the performance of the path planner when UAVs do not exchange obstacle information. The first aircraft to detect both obstacles is UAV_1. In Fig. 13a, this aircraft can be seen taking appropriate actions to avoid Obstacle_1. However, since UAVs do not cooperate, UAV_2 detects both oncoming obstacles rather late and is thus forced to execute drastic maneuvers to avoid them both (Figs. 13b and 13c). UAV_2 thus intercepts the moving target after UAV_1 (Fig. 13d). The total mission time (time required for both UAVs to intercept the target) is $t_m = 190$ s.

2. *Test Case 2: Cooperation Between Unmanned Aerial Vehicles*

Figure 14 illustrates the performance of the path planner when UAVs in communication range exchange obstacle information. Once again, UAV_1 is the first aircraft to detect both obstacles and, as such, its trajectory is very similar to the previous test case. Since UAV_1 communicates information on the detected obstacles to UAV_2, the latter now benefits from a priori information on the oncoming obstacles and thus has more time to execute an evasive course of action. As a result, UAV_2 chooses a different, much smoother trajectory than in the previous test case. Moreover, even though the target-tracking criterion of Eq. (45) does not explicitly seek to minimize the UAV-to-target arrival time, cooperation between UAVs nonetheless decreases the mission time by roughly 8.5%. In this test case, the total mission time is $t_m = 174$ s.

3. *Performance Analysis*

The objective function of Eq. (37) provides a means of analyzing the performance of the path planner throughout both mission scenarios. Figure 15 depicts the contributions $j_u(k)$, $j_{ob}(k)$, and

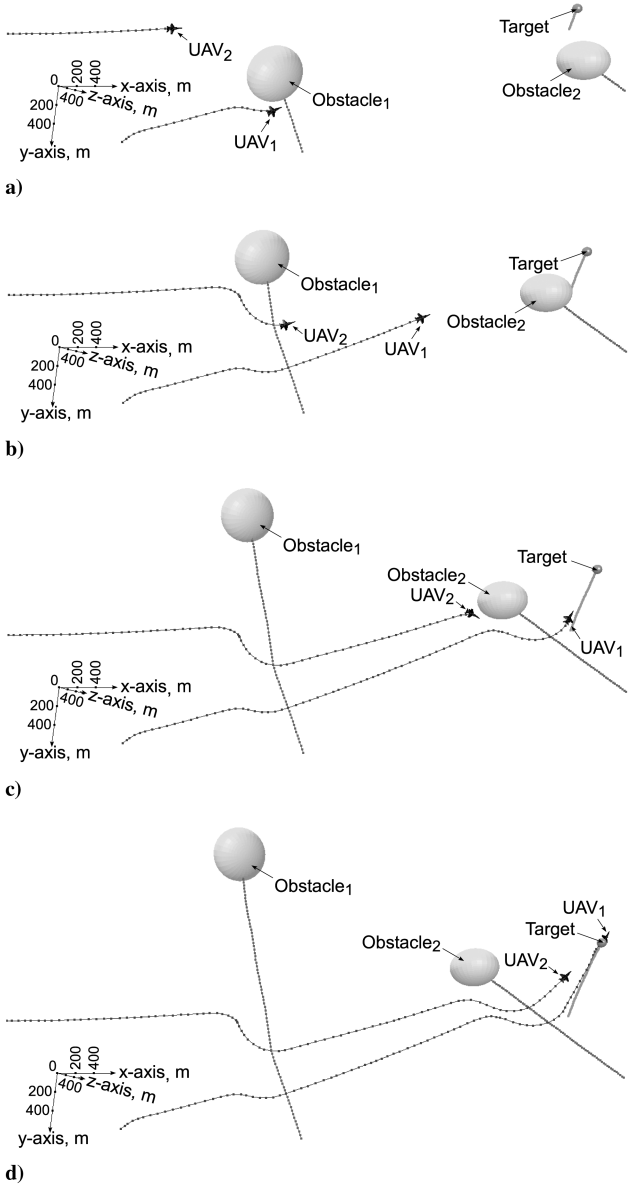


Fig. 13 Path planning without cooperation: a) $t = 46$ s, b) $t = 90$ s, c) $t = 140$ s, and d) $t = 170$ s.

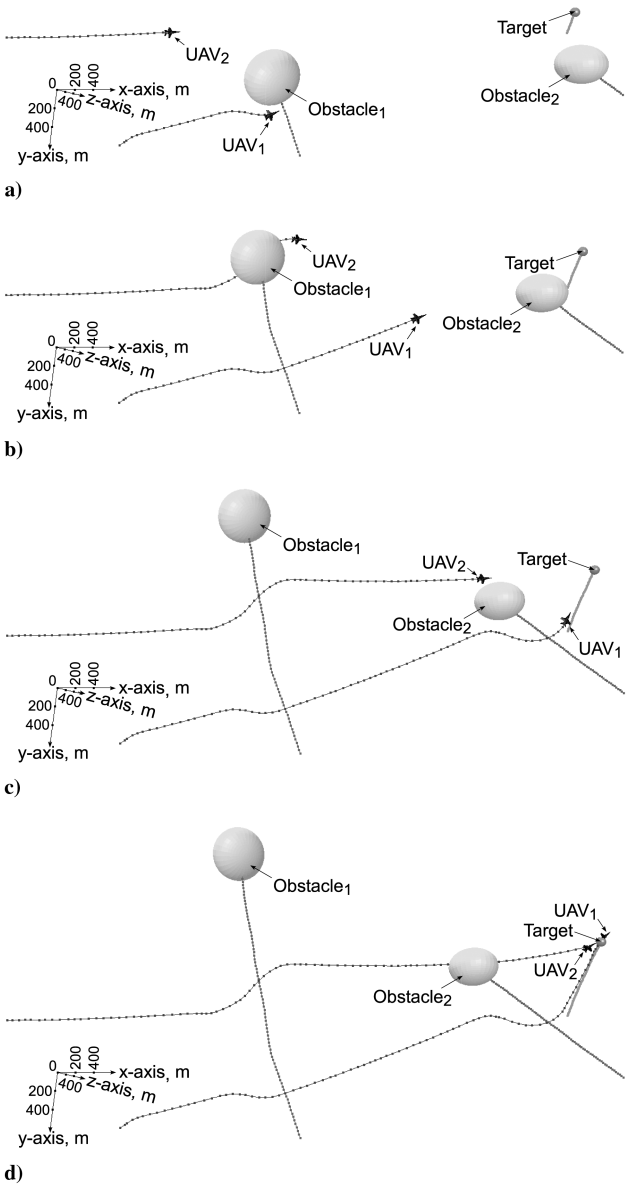
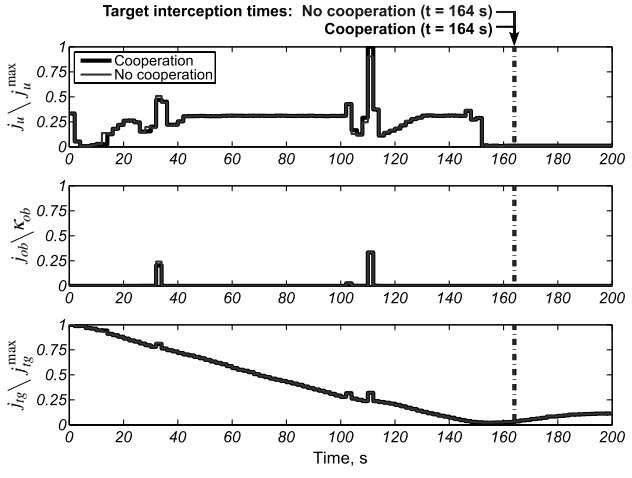
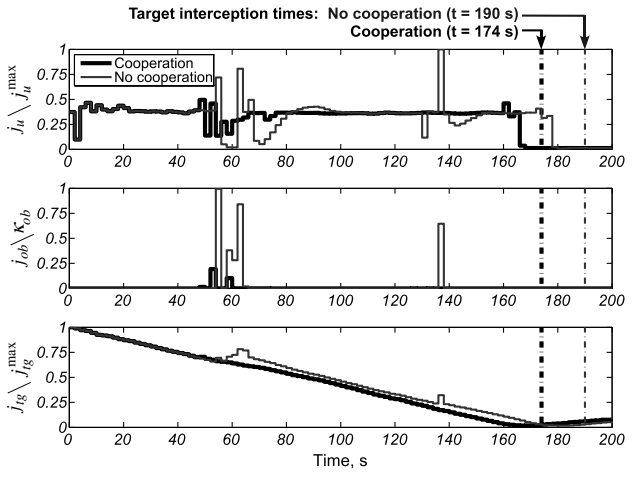


Fig. 14 Path planning with cooperation: a) $t = 46$ s, b) $t = 90$ s, c) $t = 140$ s, and d) $t = 170$ s.

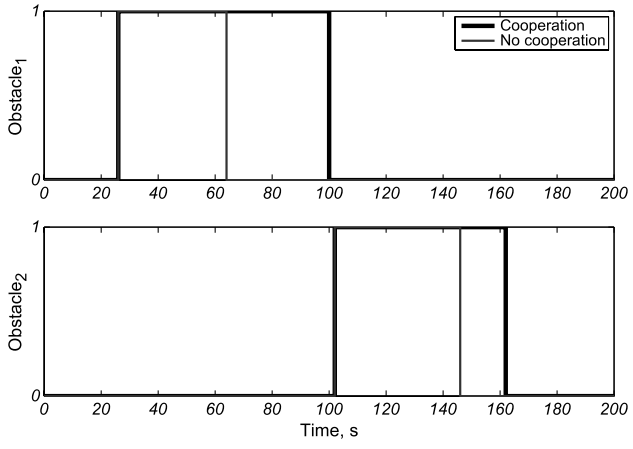


a)

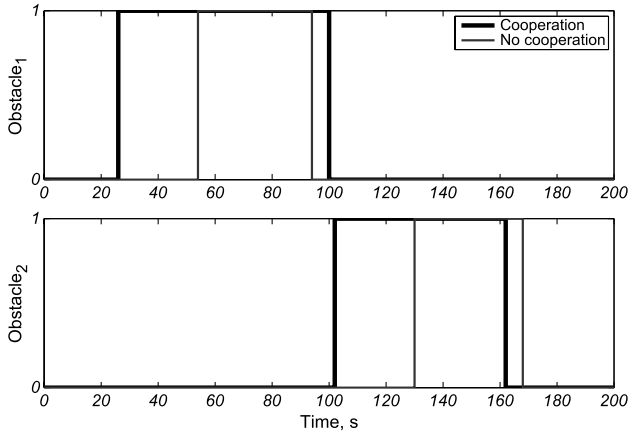


b)

Fig. 15 Objective function with and without cooperation: a) UAV_1 and b) UAV_2.



a)



b)

Fig. 16 Obstacle detection status, not detected → 0 and detected → 1: a) UAV_1 and b) UAV_2.

$j_{tg}(k)$ to the objective function $j(k)$ for each UAV throughout the simulations. Each inset in Figs. 15a and 15b depicts the results obtained when UAVs accomplish the mission, with and without cooperation. Contributions $j_u(k)$ and $j_{tg}(k)$, illustrated in Fig. 15, are normalized with respect to their maximum values for both test cases, whereas $j_{ob}(k)/\kappa_{ob}$ is the probability of collision for the specified UAV.

Figure 16 illustrates the obstacle detection status for each UAV, with and without cooperation. When UAVs cooperate, they communicate as soon as an obstacle is detected by one of the aircraft [large communication ranges in Eq. (61)].

The evolution of the objective function of UAV_1 is similar throughout both test cases (Fig. 15a). This is to be expected, since this UAV is the first to detect both obstacles (Fig. 16a) and thus does not truly benefit from cooperation. Contrarily, UAV_2 benefits from cooperation (Fig. 16b), which improves its performance throughout the mission (Fig. 15b). First, when UAVs cooperate, UAV_2 setpoint variations are generally of comparable amplitude throughout the mission. However, without cooperation, UAV_2 suffers more frequent and considerable variations in its setpoints between sample times. Second, when UAVs do not cooperate, the probability of collision for UAV_2 is high, especially upon detection of $Obstacle_1$. When UAVs cooperate, the probability of collision for UAV_2 diminishes considerably and only exceeds $p_{i,\tau}^{\max} = 0.1$ at two sample times. This allows UAV_2 to clear the obstacles at a safer distance than without cooperation. Finally, the rate of decrease of $j_{tg}(k)$ for UAV_2 is larger with cooperation, thus permitting this aircraft to reach the target faster than without cooperation.

V. Conclusions

In this paper, an EKF-based algorithm that predicts the trajectory of a moving object (obstacle or target) from its measured position was first presented. The algorithm was then tested in a decentralized model-based predictive control path planner that computes optimal UAV setpoints based on predicted obstacle and target trajectories. A novel model-based predictive control obstacle avoidance performance criterion that seeks to minimize the UAV collision probability with all known dynamic ellipsoidal obstacles on the prediction horizon, all while ensuring that it does not surpass a preset threshold at each prediction step, was also presented in this paper.

It was shown that accurate object trajectories can be predicted when the EKF model well represents the simulated object dynamics. It was also shown that communication between UAVs improves simulation results, whether it is by increasing the accuracy of predicted object trajectories or by reducing the total mission time. Finally, it was shown that the exact UAV collision probability can be computed by convolving the shape of the obstacle with the statistical distribution of its center of mass position.

Future work will involve modifying the existing EKF structure to include fault detection in order to eliminate sensor biases and disregard measurements generated by defective or damaged sensors. An adaptive EKF structure will also be developed to deal with objects for which the dynamics and uncertainties are being identified en route. The EKF will also be compared with other filters, such as the unscented Kalman filter, to determine which algorithm yields the most accurate predicted object trajectories. Finally, the proposed algorithms will be tested in a hardware-in-the-loop system, and practical problems (e.g., payload requirements) and implementation

problems (e.g., real-time data-processing requirements) will be addressed before testing the proposed algorithm in real flight.

Appendix A: Notation

Throughout the text, the temporal index k complements key variables in order to indicate the sample time at which these variables occur. For example, scalars (e.g., x), vectors (e.g., $\boldsymbol{x}$), and matrices (e.g., $\boldsymbol{X}$) occurring at sample time k are denoted $x(k)$, $\boldsymbol{x}(k)$, and $\boldsymbol{X}(k)$, respectively. If these elements do not evolve with time, the index k is omitted.

A notation for vector and matrix predictions must also be defined to support the predictive algorithm at the core of this research. The temporal index describing a prediction at τ samples in the future from sample time k is denoted $k+\tau|k$. The notations $\boldsymbol{x}(k+\tau|k)$ and $\boldsymbol{X}(k+\tau|k)$ thus represent the prediction of a vector and a matrix at sample time $k+\tau$ from sample time k. Moreover, the predictions of a vector over a prediction horizon h_p (i.e., for the future times $k+\tau$ with $\tau \in \{1, 2, \ldots, h_p\}$) is denoted $\boldsymbol{x}(k+1\colon k+h_p|k)$, where

$$\boldsymbol{x}(k+1\colon k+h_p|k) = \begin{bmatrix} \boldsymbol{x}(k+1|k) \\ \boldsymbol{x}(k+2|k) \\ \vdots \\ \boldsymbol{x}(k+h_p|k) \end{bmatrix} \tag{A1}$$

Appendix B: Ellipsoid Formulation

The reference frame adopted herein is the NED Euclidean coordinate system, where the positive x axis is forward, the positive y axis points to the right, and the positive z axis points downward. The algebraic representation of an ellipsoid centered at the origin of this reference frame is

$$\frac{\mathsf{x}^2}{a^2} + \frac{\mathsf{y}^2}{b^2} + \frac{\mathsf{z}^2}{c^2} = 1 \tag{B1}$$

and can be written in matrix form as

$$\boldsymbol{n}^T \boldsymbol{Q} \boldsymbol{n} = 1 \tag{B2}$$

where

$$\boldsymbol{n} = [\mathsf{x} \quad \mathsf{y} \quad \mathsf{z}]^T; \qquad \boldsymbol{Q} = \begin{bmatrix} a^{-2} & 0 & 0 \\ 0 & b^{-2} & 0 \\ 0 & 0 & c^{-2} \end{bmatrix} \tag{B3}$$

Rotations of this ellipsoid about any combination of axes can be accomplished through the matrix representation

$$\boldsymbol{n}^T \boldsymbol{R} \boldsymbol{Q} \boldsymbol{R}^T \boldsymbol{n} = 1 \tag{B4}$$

where $\boldsymbol{R}$ is the desired rotation matrix. In the path planner, rotations are performed relative to moving (or current) axes [35] such that, from the NED-fixed coordinate system, the first Euler rotation (roll) is of an angle ϕ about the z axis:

$$\boldsymbol{R}_{\mathsf{z}} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B5}$$

the second Euler rotation (pitch) is of an angle ϑ about the y' axis of the first intermediate coordinate systems $x'y'z'$:

$$\boldsymbol{R}_{\mathsf{y}} = \begin{bmatrix} \cos\vartheta & 0 & \sin\vartheta \\ 0 & 1 & 0 \\ -\sin\vartheta & 0 & \cos\vartheta \end{bmatrix} \tag{B6}$$

and the third Euler rotation (yaw) is of an angle ψ about the x'' axis of the second intermediate coordinate system $x''y''z''$:

$$\boldsymbol{R}_{\mathsf{x}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix} \tag{B7}$$

The roll, pitch, and yaw rotation matrices are thereafter post-multiplied to yield the desired object rotation:

$$\boldsymbol{R} = \boldsymbol{R}_{\mathsf{z}} \boldsymbol{R}_{\mathsf{y}} \boldsymbol{R}_{\mathsf{x}} \tag{B8}$$

Moreover, a translation of the rotated ellipsoid $\boldsymbol{n}^T \boldsymbol{R} \boldsymbol{Q} \boldsymbol{R}^T \boldsymbol{n} = 1$ to the point $\boldsymbol{n}_0 = [\mathsf{x}_0 \quad \mathsf{y}_0 \quad \mathsf{z}_0]^T$ relative to the NED-fixed coordinate system yields

$$(\boldsymbol{n} - \boldsymbol{n}_0)^T \boldsymbol{R} \boldsymbol{Q} \boldsymbol{R}^T (\boldsymbol{n} - \boldsymbol{n}_0) = 1 \tag{B9}$$

Appendix C: Convolution Integral

I. Implementation

In theory, the convolution integral of Eq. (46) is continuous and evaluated over an infinite range. However, in practice, the integral can only be computed at discrete points over a finite range. In the simulations of Sec. IV.B, the convolution sum is evaluated at equally spaced positions (guide points) over a cubic region centered on the

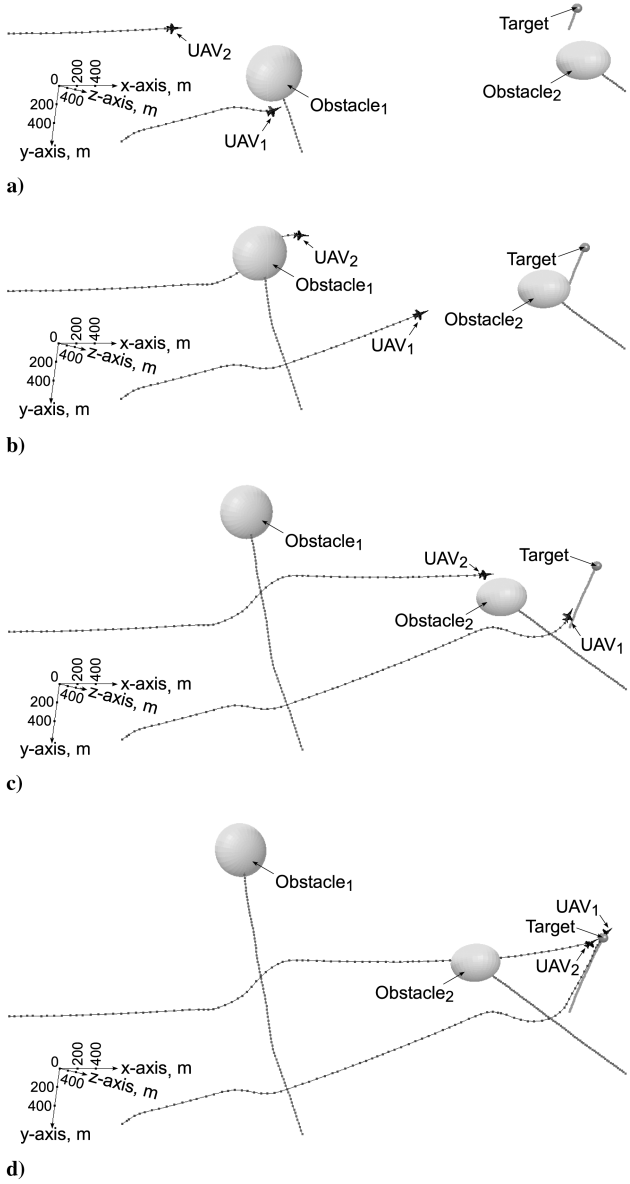


Fig. C1 Convolution example: a) object shape, b) center of mass distribution, and c) convolution result.

Table C1 Ellipsoidal objects and ranges

Object/range	Mathematical formulation
Obstacle	$[\boldsymbol{n}-\boldsymbol{n}_{ob}(k)]^T\boldsymbol{R}_{ob}\boldsymbol{Q}_{ob}\boldsymbol{R}_{ob}^T[\boldsymbol{n}-\boldsymbol{n}_{ob}(k)]=1$
Target	$[\boldsymbol{n}-\boldsymbol{n}_{tg}(k)]^T\boldsymbol{R}_{tg}\boldsymbol{Q}_{tg}\boldsymbol{R}_{tg}^T[\boldsymbol{n}-\boldsymbol{n}_{tg}(k)]=1$
Sensor range	$[\boldsymbol{n}-\boldsymbol{n}_{uv}(k)]^T\boldsymbol{R}_{\mathrm{sn}}\boldsymbol{Q}_{\mathrm{sn}}\boldsymbol{R}_{\mathrm{sn}}^T[\boldsymbol{n}-\boldsymbol{n}_{uv}(k)]=1$
Communication range	$[\boldsymbol{n}-\boldsymbol{n}_{uv}(k)]^T\boldsymbol{R}_{cm}\boldsymbol{Q}_{cm}\boldsymbol{R}_{cm}^T[\boldsymbol{n}-\boldsymbol{n}_{uv}(k)]=1$

obstacle center of mass. The length of the cube edges are chosen to equal

$$2\{\boldsymbol{q}_{ob_i}^{\max}+[3\sqrt{\boldsymbol{\sigma}_{ob_i}^2(k+\tau|k)}]^{\max}\}$$

to minimize the information lost when truncating the resulting collision probability function. Linear interpolation between guide points is then applied to render the convolution surface continuous.

II. Two-Dimensional Example

Consider the object

$$\frac{\mathsf{x}^2}{8^2}+\frac{\mathsf{y}^2}{4^2}=1 \tag{C1}$$

for which the probability of a point $\boldsymbol{n}=[\,\mathsf{x}\quad\mathsf{y}\,]^T$ to reside within the object is given by the indicator function (Fig. C1a)

$$g(\boldsymbol{n})=\begin{cases}1 & \frac{\mathsf{x}^2}{8^2}+\frac{\mathsf{y}^2}{4^2}\le 1\\ 0 & \text{otherwise}\end{cases} \tag{C2}$$

If the location of the object center of mass is uncertain and normally distributed, such that (Fig. C1b)

$$f(\boldsymbol{n})=\frac{1}{\sqrt{4(2\pi)^2}}\exp\left\{-\frac{1}{2}\left[(\mathsf{x}-5)^2+\frac{\mathsf{y}^2}{4}\right]\right\} \tag{C3}$$

then the probability of a point $\boldsymbol{n}$ to fall inside the object [computed from Eq. (41)] is the result illustrated in Fig. C1c.

Acknowledgments

This work was supported in part by La Fondation Baxter and Alma Ricard, the Natural Sciences and Engineering Research Council of Canada, and Defence Research and Development Canada.

References

[1] Newcome, L. R., "Birth of a Concept," *Unmanned Aviation: A Brief History of Unmanned Aerial Vehicles*, 1st ed., Library of Flight, AIAA, Reston, VA, 2004, pp. 11–14.

[2] Committee on Autonomous Vehicles in Support of Naval Operations and National Research Council (ed.), "Unmanned Aerial Vehicles: Capabilities and Potential," *Autonomous Vehicles in Support of Naval Operations*, 1st ed., National Academies Press, Washington, D. C., 2005, pp. 82–115.

[3] DeGarmo, M., and Nelson, G. M., "Prospective Unmanned Aerial Vehicle Operations in the Future National Airspace System," 4th Aviation Technology, Integration, and Operations Forum, Chicago, IL, AIAA Paper 2004-6243, 2004.

[4] Watanabe, Y., Calise, A. J., Johnson, E. N., and Evers, J. H., "Minimum-Effort Guidance for Vision-Based Collision Avoidance," Atmospheric Flight Mechanics Conference and Exhibit, Keystone, CO, AIAA Paper 2006-6641, 2006.

[5] Dogan, A., and Zengin, U., "Unmanned Aerial Vehicle Dynamic-Target Pursuit by Using Probabilistic Threat Exposure Map," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 4, 2006, pp. 944–954. doi:10.2514/1.18386

[6] Stubberud, S. C., and Kramer, K. A., "Predictive Guidance Intercept Using the Neural Extended Kalman Filter Tracker," *Control and Intelligent Systems*, Vol. 35, No. 3, 2007, pp. 228–235.

[7] Dionne, D., Michalska, H., and Rabbath, C. A., "Predictive Guidance for Pursuit-Evasion Engagements Involving Multiple Decoys," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1277–1286. doi:10.2514/1.25481

[8] Mora, M. C., and Tornero, J., "Path Planning and Trajectory Generation Using Multi-Rate Predictive Artificial Potential Fields," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, IEEE Publ., Piscataway, NJ, 2008, pp. 2990–2995. doi:10.1109/IROS.2008.4651091

[9] Prévost, C. G., Desbiens, A., and Gagnon, E., "Extended Kalman Filter for State Estimation and Trajectory Prediction of a Moving Object Detected by an Unmanned Aerial Vehicle," *Proceedings of the American Control Conference*, New York, IEEE Publ., Piscataway, NJ, 2007, pp. 1805–1810. doi:10.1109/ACC.2007.4282823

[10] Lee, L.-F., "Decentralized Motion Planning Within An Artificial Potential Framework (APF) for Cooperative Payload Transport by Multi-Robot Collectives," M.S. Dissertation, State Univ. of New York at Buffalo, Buffalo, NY, 2004, pp. 32–33, http://www.eng.buffalo.edu/mechatronics/research/mobilemanipulator/Publications/LengFengThesis.pdf [retrieved 30 Sept. 2010].

[11] Chou, F.-Y., Yang, C.-Y., and Yang, J.-S., "Support Vector Machine Based Artificial Potential Field for Autonomous Guided Vehicle," *Proceedings of the 4th International Symposium on Precision Mechanical Measurements*, edited by Y. Fei, K.-C. Fan, and R. Lu, Vol. 7130, SPIE, Bellingham, WA, 2008, pp. 71304J.1–71304J.6.

[12] Helble, H., and Cameron, S., "3-D Path Planning and Target Trajectory Prediction for the Oxford Aerial Tracking System," *Proceedings of the IEEE International Conference on Robotics and Automation*, Roma, Italy, IEEE Publ., Piscataway, NJ, 2007, pp. 1042–1048. doi:10.1109/ROBOT.2007.363122

[13] Massoud, A. A., "Managing the Dynamics of a Harmonic Potential Field-Guided Robot in a Cluttered Environment," *IEEE Transactions on Industrial Electronics*, Vol. 56, No. 2, 2009, pp. 488–496. doi:10.1109/TIE.2008.2002720

[14] Massoud, A. A., "Decentralized Self-Organizing Potential Field-Based Control for Individually Motivated Mobile Agents in a Cluttered Environment: A Vector-Harmonic Potential Field Approach," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 37, No. 3, 2007, pp. 372–390. doi:10.1109/TSMCA.2007.893483

[15] Massoud, A. A., "A Harmonic Potential Field Approach with a Probabilistic Space Descriptor for Planning in Non-Divisible Environments," *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, IEEE Publ., Piscataway, NJ, 2009, pp. 3774–3779. doi:10.1109/ROBOT.2009.5152176

[16] Borrelli, F., Keviczky, T., and Balas, G. J., "Collision-free UAV Formation Flight Using Decentralized Optimization and Invariant Sets," *Proceedings of the 43rd IEEE Conference on Decision and Control*, Vol. 1, Nassau, Bahamas, IEEE Publ., Kobe, Japan, 2004, pp. 1099–1104. doi:10.1109/CDC.2004.1428839

[17] Xie, F., "Model Predictive Control of Nonholonomic Mobile Robots," Ph.D. Dissertation, Oklahoma State Univ., Stillwater, OK, 2007, pp. 124–125, http://gradworks.umi.com/32/91/3291276.html [retrieved 30 Sept. 2010].

[18] Xie, F., and Fierro, R., "Stabilization of Nonholonomic Robot Formations: A First-State Contractive Model Predictive Control Approach," *Journal of Computing and Information Technology - CIT*, Vol. 17, No. 1, 2007, pp. 37–50. doi:10.2498/cit.1001188

[19] Kang, Y., and Hedrick, J. K., "Design of Nonlinear Model Predictive Controller for a Small Fixed-wing Unmanned Aerial Vehicle," Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, AIAA Paper 2006-6685, 2006.

[20] Shim, D. H., and Sastry, S., "An Evasive Maneuvering Algorithm for UAVs in See-and-Avoid Situation," *Proceedings of the American Control Conference*, New York, IEEE Publ., Piscataway, NJ, 2007, pp. 3886–3891.doi:10.1109/ACC.2007.4283147

[21] Richards, A., and How, J., "Decentralized Model Predictive Control of Cooperating UAVs," *Proceedings of the 43rd IEEE Conference on Decision and Control*, Vol. 4, Paradise Island, Bahamas, IEEE Publ., Piscataway, NJ, 2004, pp. 4286–4291. doi:10.1109/CDC.2004.1429425

[22] Boivin, E., Desbiens, A., and Gagnon, E., "Collision Avoidance Using Cooperative Predictive Control," *Proceedings of the 16th Mediterranean Conference on Control and Automation*, Ajaccio, France, IEEE Publ., Piscataway, NJ, 2008, pp. 682–688. doi:10.1109/MED.2008.4602109

[23] Maciejowski, J. M., "A Basic Formulation of Predictive Control," *Predictive Control with Constraints*, 1st ed., Prentice–Hall, Harlow,

England, 2002, pp. 36–73.

[24] Kailath, T., Sayed, A. H., and Hassibi, B., "The Kalman Filter," *Linear Estimation*, 1st ed., Prentice–Hall, Upper Saddle River, NJ, 2000, pp. 310–369.

[25] Bierman, G. J., "Inclusion of Mapping Effects and Process Noise," *Factorization Methods for Discrete Sequential Estimation*, 1st ed., Academic Press, New York, 1977, pp. 113–134.

[26] Bar-Shalom, Y., Li, X.-R., and Kirubarajan, T., "Linear Dynamic Systems with Random Inputs," *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*, 1st ed., Wiley, New York, 2001, pp. 179–198.

[27] Alfano, S., and Greer, M. L., "Determining if Two Solid Ellipsoids Intersect," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 1, 2003, pp. 106–110.
doi:10.2514/2.5020

[28] Stewart, J., "Three-Dimensional Analytic Geometry and Vectors," *Calculus: Early Transcendentals*, 2nd ed., Brooks/Cole Publ., Pacific Grove, CA, 1991, pp. 630–703.

[29] Izadi, H. A., Gordon, B. W., and Rabbath, C. A., "A Variable Communication Approach for Decentralized Receding Horizon Control of Multi-Vehicle Systems," *Proceedings of the American Control Conference*, New York, IEEE Publ., Piscataway, NJ, 2007, pp. 1781–1786.
doi:10.1109/ACC.2007.4283112

[30] Prévost, C. G., Thériault, O., Desbiens, A., Poulin, E., and Gagnon, E., "Receding Horizon Model-Based Predictive Control for Dynamic Target Tracking; a Comparative Study," Guidance, Navigation, and Control Conference, Chicago, IL, AIAA Paper 2009-6268, 2009.

[31] Goodwin, G. C., and Sin, K. S., "Optimal Filtering and Prediction," *Adaptive Filtering Prediction and Control*, 1st ed., Prentice–Hall, Englewood Cliffs, NJ, 1984, pp. 245–300.

[32] Maybeck, P. S., "Optimal Filtering with Linear System Models," *Stochastic Models, Estimation, and Control*, 1st ed., Vol. 1, Academic Press, New York, 1979, pp. 203–288.

[33] Zarchan, P., "Introduction," *Tactical and Strategic Missile Guidance*, edited by F. K. Lu, 5th ed., Vol. 219, Progress in Astronautics and Aeronautics, AIAA, Reston, VA, 2007, pp. 1–8.

[34] Ben-Asher, J. Z., and Yaesh, I., "Introduction," *Advances in Missile Guidance Theory*, edited by P. Zarchan, 1st ed., Progress in Astronautics and Aeronautics, Vol. 180, AIAA, Reston, VA, 1998, pp. 1–6.

[35] Niku, S. B., "Robot Kinematics: Position Analysis," *Introduction to Robotics, Analysis, Systems, Applications*, 1st ed., Prentice–Hall, Upper Saddle River, NJ, 2001, pp. 29–94.