Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

UAV Optimal Cooperative Target Tracking and Collision Avoidance of Moving Objects ⋆

Carole G. Prévost * **André Desbiens** * **Eric Gagnon** **
Daniel Hodouin ***

* *LOOP, Department of Electrical and Computer Engineering*
Université Laval, Québec City, Québec, G1V 0A6, Canada
** *Defence R&D Canada-Valcartier, Québec City, Québec, G3J 1X5, Canada*
*** *LOOP, Department of Mining, Metallurgical, Materials Engineering*
Université Laval, Québec City, Québec, G1V 0A6, Canada

Abstract: Effective target tracking and collision avoidance schemes are essential to the success of unmanned aerial vehicle (UAV) missions. In a dynamic environment, UAV path planning often relies on predicted obstacle and target motion. This paper presents an algorithm that predicts the trajectory of a moving object (target or obstacle) detected by a UAV. An extended Kalman filter is first employed to estimate the states of the object from its measured spatial position. The optimal object trajectory and its associated position prediction error are then calculated using the state equation defined for Kalman filtering. The proposed trajectory prediction scheme is afterward tested in a path planner which relies on decentralized cooperative predictive control to select optimal UAV trajectories as a function of the predicted target and obstacle trajectories. Simulation results are presented to demonstrate the effectiveness of the proposed approach.

1. INTRODUCTION

It is thought that Leonardo Da Vinci was the first to conceptualize the unmanned aerial vehicle (UAV)(Newcome, 2004). Today, UAVs play an important role in military operations and have the potential to serve in countless civil applications. For such reasons, the interest in further developing this technology has grown tremendously over the years. Two areas of research requiring additional development are collision avoidance and target tracking in unknown dynamic environments.

UAV path planning in a dynamic environment strongly relies on predicted target and obstacle motion (Dogan and Zengin, 2006; Penney, 2005; Elnagar, 1999). Predicting erroneous object trajectories can have serious repercussions on the UAV and may even lead to mission failure. In order to accurately predict the future trajectory of an object, it is essential to correctly estimate its present state. If the object dynamics are uncertain and if sensors collecting data about the object are noisy, estimating the true object states can become a difficult task.

Kalman filters are often employed to estimate object states, given model and measurement uncertainty (Dogan and Zengin, 2006; Prévost et al., 2007). In Dogan and Zengin (2006), a Kalman filter is used to estimate a target's speed and heading from its measured xy-plane position. Authors employ the filter to infer the coefficients of the polynomials modeling the target's speed and heading dynamics. In Prévost et al. (2007), an extended Kalman filter (EKF) is proposed to estimate the states of a moving object detected by a UAV. The object is considered as a controlled system of known dynamics and the EKF attempts to estimate its true setpoints, internal model states, model outputs and position, in spite of model uncertainty and measurement uncertainty. These estimated states together with the EKF state equation are then used to calculate the optimal trajectory of the particular object over a specified prediction horizon. The quality of the predicted trajectory is also evaluated by calculating the variance of the position prediction error.

In this paper, the algorithms in Prévost et al. (2007) are tested in a path planner which relies on cooperative decentralized model-based predictive (MP) control to guide a fleet of UAVs in an unknown three-dimensional dynamic environment. The mission objective is to ensure that each UAV intercepts several moving targets while avoiding multiple moving obstacles. For this purpose, UAVs are equipped with an autopilot, sensors and a communication device. Kalman filters as proposed in Prévost et al. (2007) are employed to predict the trajectories of the moving target and of the known moving obstacles. The path planner's trajectory control units (TCUs) (one for each UAV) will select each vehicle's optimal trajectory based on the predicted target and obstacle displacements.

In section 2, the state estimation and trajectory prediction algorithms based on Kalman filtering are presented. The reader is referred to Prévost et al. (2007) for a detailed description of the algorithms. In section 3, the UAV path planner used to test the algorithms of section 2 is described. Section 4 presents the TCU of a single UAV. In section 5, two simulations are presented: they demonstrate the performance of the path planner when UAVs do not

⋆ This work was supported by La Fondation Baxter & Alma Ricard and the Natural Sciences & Engineering Research Council of Canada (NSERC). Their financial support has made this work possible. Thanks to also accorded to Defence Research and Development Canada (DRDC) who has initiated and also supported this project.

cooperate and cooperate to fulfill the mission objectives. Finally, future work is presented in section 6.

2. TRAJECTORY PREDICTION OF A MOVING OBJECT

A UAV detects a moving object (obstacle or target) and records its spatial position. The object is identified and its dynamic model is retrieved from a database. Knowing that this model is an estimation of the true object dynamics (model error) and that the UAV sensors are noisy (measurement error), the object states must be estimated in order to predict its trajectory. UAV path planning will thereafter depend on the predicted object trajectory.

Suppose that the model in Fig. 1, retrieved from a database, depicts the dynamics of the object motion.

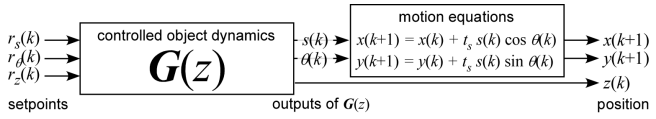


Fig. 1. Motion model of object: obstacle, target, UAV.

In Fig. 1, the transfer function matrix $\boldsymbol{G}(z)$ models the *closed-loop* dynamics of the detected object (e.g. combined vehicle and autopilot dynamics). The outputs of $\boldsymbol{G}(z)$ (xy-plane speed s, xy-plane heading θ and altitude z) thus follow the inputs of $\boldsymbol{G}(z)$ (xy-plane speed setpoint r_s, xy-plane heading setpoint r_θ, altitude setpoint r_z) according to the dynamics specified in $\boldsymbol{G}(z)$. Motion equations then relate the object's planar speed s and planar heading θ to its x-axis x and y-axis y positions. Note that the altitude of the object directly corresponds to its z-axis position.

2.1 State Estimation

An EKF is first used to estimate the states of the object whose motion model is depicted in Fig. 1. For Kalman filtering, a non-linear state-space representation of type

$$\boldsymbol{x}(k+1) = f\left[\boldsymbol{x}(k)\right] + \boldsymbol{\omega}(k) \tag{1}$$
$$\boldsymbol{y}(k) = h\left[\boldsymbol{x}(k)\right] + \boldsymbol{\nu}(k) \tag{2}$$

describing the object motion is required. Elements $f[\cdot]$ and $h[\cdot]$ represent non-linear functions of $\cdot$, whereas $\boldsymbol{x}$ and $\boldsymbol{y}$ are the state and measurement vectors. The vectors $\boldsymbol{\omega}$ and $\boldsymbol{\nu}$ are stochastic processes (Gaussian, zero-mean, white noise, statistically independent), which represent the uncertainty and perturbations on the model, and the uncertainty on the measurements respectively.

The EKF presented herein estimates the object states $\boldsymbol{x}$ at the next sample time. These estimated states are denoted $\hat{\boldsymbol{x}}(k+1|k)$, where the time index $k+1|k$ signifies an estimation at time $k+1$ from time k and where the symbol $\hat{\cdot}$ indicates the optimal estimation.

State and Measurement Vectors The states to be estimated by the EKF are the object setpoints ($\boldsymbol{x_r}$), outputs of $\boldsymbol{G}(z)$ ($\boldsymbol{x_y}$), xy-plane position ($\boldsymbol{x_n}$), and internal states of $\boldsymbol{G}(z)$ ($\boldsymbol{x_G}$). The state vector used for Kalman filtering is thus

$$\boldsymbol{x} = [\boldsymbol{x_r}^T \;\; \boldsymbol{x_G}^T \;\; \boldsymbol{x_y}^T \;\; \boldsymbol{x_n}^T]^T \tag{3}$$

where

$$\boldsymbol{x_r} = [r_s \;\; r_\theta \;\; r_z]^T \tag{4}$$
$$\boldsymbol{x_y} = [s \;\; \theta \;\; z]^T \tag{5}$$
$$\boldsymbol{x_n} = [x \;\; y]^T \tag{6}$$

and where $\boldsymbol{x_G}$ is composed of all states modeling the dynamics in $\boldsymbol{G}(z)$.

Upon detection of the object, the UAV measures its spatial position. The measurement vector of the EKF is thus

$$\boldsymbol{y} = [x^m \;\; y^m \;\; z^m]^T \tag{7}$$

where the superscript m indicates a measured state.

State Equation The state equation describes the temporal evolution of all system states. As shown in (3), the evolution of states $\boldsymbol{x_r}$, $\boldsymbol{x_G}$, $\boldsymbol{x_y}$ and $\boldsymbol{x_n}$ must be defined.

The object setpoints $\boldsymbol{x_r}$ are completely unknown. They are thus modeled as independent random walks (Bar-Shalom et al., 2001)

$$\boldsymbol{x_r}(k+1) = \boldsymbol{x_r}(k) + \boldsymbol{\omega_r}(k) \tag{8}$$

where $\boldsymbol{\omega_r} = [\omega_{r_s} \;\; \omega_{r_\theta} \;\; \omega_{r_z}]^T$ is a random vector. The variances of the elements in $\boldsymbol{\omega_r}$ are set to model variations in the setpoints between sample times.

The temporal evolution of the states $\boldsymbol{x_y}$ and $\boldsymbol{x_G}$ are described by the object dynamics $\boldsymbol{G}(z)$. The discrete state-space representation of $\boldsymbol{G}(z)$ yields

$$\boldsymbol{x_G}(k+1) = \boldsymbol{A_G}\boldsymbol{x_G}(k) + \boldsymbol{B_G}\boldsymbol{x_r}(k) + \boldsymbol{\omega_G}(k) \tag{9}$$
$$\boldsymbol{x_y}(k) = \boldsymbol{C_G}\boldsymbol{x_G}(k) \tag{10}$$

where $\boldsymbol{A_G}$, $\boldsymbol{B_G}$, and $\boldsymbol{C_G}$ are the state matrices of $\boldsymbol{G}(z)$ and where the random vector $\boldsymbol{\omega_G}$ models the uncertainty on $\boldsymbol{G}(z)$ or models perturbations affecting the states of $\boldsymbol{G}(z)$.

The temporal evolution of the xy-plane object position is governed by the motion equations in Fig. 1, where t_s is the sampling period. These motion equations are non-linear

$$\boldsymbol{x_n}(k+1) = \boldsymbol{x_n}(k) + f\left[\boldsymbol{x_y}(k)\right] \tag{11}$$

and as such, are linearized at each sample time of the EKF algorithm. Thus,

$$\boldsymbol{x_n}(k+1) = \boldsymbol{x_n}(k) + \boldsymbol{B_{n_k}}\boldsymbol{x_y}(k) + \boldsymbol{\omega_n}(k) \tag{12}$$

where the matrix $\boldsymbol{B_n}$ results from the first order Taylor series approximation of (11) and where the random vector $\boldsymbol{\omega_n} = [\omega_x \;\; \omega_y]^T$ models errors due to approximating (11) by its linear counterpart.

Combining (8), (9), (10), and (11) yields the complete non-linear state equation,

$$\boldsymbol{x}(k+1) = f[\boldsymbol{x}(k)] + \boldsymbol{\omega}^{nl}(k) \tag{13}$$

where $\boldsymbol{\omega}^{nl} = [\boldsymbol{\omega_r}^T \;\; \boldsymbol{\omega_G}^T \;\; \boldsymbol{\omega_G}^T\boldsymbol{C_G}^T \;\; \boldsymbol{0}^T]^T$ with covariance matrix $\boldsymbol{W}^{nl}$. Combining (8), (9), (10), and (12) yields the linear counterpart of (13),

$$\boldsymbol{x}(k+1) = \boldsymbol{A}_k\boldsymbol{x}(k) + \boldsymbol{\omega}^l(k) \tag{14}$$

where the random vector $\boldsymbol{\omega}^l = [\boldsymbol{\omega_r}^T \;\; \boldsymbol{\omega_G}^T \;\; \boldsymbol{\omega_G}^T\boldsymbol{C_G}^T \;\; \boldsymbol{\omega_n}^T]^T$ with covariance matrix $\boldsymbol{W}^l$ models the uncertainty on all system states.

Measurement Equation The UAV measures the position of the object. The measurement equation is thus

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{\nu}(k) \tag{15}$$

where the matrix $\boldsymbol{C}$ locates the states measured in $\boldsymbol{x}$. The random vector $\boldsymbol{\nu}$ with covariance matrix $\boldsymbol{V}$ models uncertainty on all measurements (sensor noise).

2.2 Trajectory Prediction

Once the detected object's states have been estimated, its future trajectory can be predicted.

Prediction of Optimal Object Trajectory The prediction of the optimal object trajectory over a future horizon h_p is calculated from the non-linear state equation (13) and from the output equation

$$\boldsymbol{n}(k) = \boldsymbol{S}\boldsymbol{x}(k) \tag{16}$$

where $\boldsymbol{n} = [x\ y\ z]^T$ and where the matrix $\boldsymbol{S}$ locates the states in $\boldsymbol{x}$ pertaining to the object's position. Thus,

$$\begin{bmatrix} \hat{\boldsymbol{n}}^{nl}(k+1|k) \\ \hat{\boldsymbol{n}}^{nl}(k+2|k) \\ \hat{\boldsymbol{n}}^{nl}(k+3|k) \\ \vdots \\ \hat{\boldsymbol{n}}^{nl}(k+h_p|k) \end{bmatrix} = \boldsymbol{S} \begin{bmatrix} \hat{\boldsymbol{x}}(k+1|k) \\ f[\hat{\boldsymbol{x}}(k+1|k)] \\ f[f[\hat{\boldsymbol{x}}(k+1|k)]] \\ \vdots \\ f[\cdots f[\hat{\boldsymbol{x}}(k+1|k)]\cdots] \end{bmatrix} \tag{17}$$

where $\hat{\boldsymbol{x}}(k+1|k)$ are the states estimated by the EKF at the next sample time. The left-hand side of (17) can be condensed into

$$\hat{\boldsymbol{n}}^{nl}(1\!:\!h_p|k) \tag{18}$$

where the time index $1\!:\!h_p|k$ groups predictions $\hat{\boldsymbol{n}}^{nl}$ from sample time $k+1|k$ to sample time $k+h_p|k$.

Variance of the Position Prediction Error The quality of the predicted trajectory is evaluated by computing the variance of the position prediction error,

$$\boldsymbol{\sigma^2}(k+\tau|k) = \mathrm{var}\{\boldsymbol{n}(k+\tau) - \hat{\boldsymbol{n}}^{l}(k+\tau|k)\} \tag{19}$$

where $\boldsymbol{n}(k+\tau)$ and $\hat{\boldsymbol{n}}^{l}(k+\tau|k)$ are the modeled object trajectory and predicted optimal object trajectory, evaluated using the linear state equation (14) (with $\boldsymbol{\omega_n}(k) = \boldsymbol{0}$) and output equation (16). Along the prediction horizon h_p, one obtains

$$\boldsymbol{\sigma^2}(1\!:\!h_p|k) = \begin{bmatrix} \boldsymbol{S}\boldsymbol{P}_{k+1|k}\boldsymbol{S}^T \\ \boldsymbol{S}\boldsymbol{M}_{1,1}\boldsymbol{P}_{k+1|k}\boldsymbol{M}_{1,1}^T\boldsymbol{S}^T + \boldsymbol{S}\boldsymbol{W}^{nl}\boldsymbol{S}^T \\ \boldsymbol{S}\boldsymbol{M}_{2,2}\boldsymbol{P}_{k+1|k}\boldsymbol{M}_{2,2}^T\boldsymbol{S}^T + \\ \boldsymbol{S}\boldsymbol{M}_{1,2}\boldsymbol{W}^{nl}\boldsymbol{M}_{1,2}^T\boldsymbol{S}^T + \boldsymbol{S}\boldsymbol{W}^{nl}\boldsymbol{S}^T \\ \vdots \\ \boldsymbol{S}\boldsymbol{M}_{h_p-1,h_p-1}\boldsymbol{P}_{k+1|k}\boldsymbol{M}_{h_p-1,h_p-1}^T\boldsymbol{S}^T + \\ \boldsymbol{S}\boldsymbol{M}_{h_p-2,h_p-1}\boldsymbol{W}^{nl}\boldsymbol{M}_{h_p-2,h_p-1}^T\boldsymbol{S}^T + \cdots + \boldsymbol{S}\boldsymbol{W}^{nl}\boldsymbol{S}^T \end{bmatrix} \tag{20}$$

where

$$\begin{aligned} \boldsymbol{M}_{1,\tau} &= \boldsymbol{A}_{k+\tau} \\ \boldsymbol{M}_{2,\tau} &= \boldsymbol{A}_{k+\tau}\boldsymbol{A}_{k+\tau-1} \\ \boldsymbol{M}_{3,\tau} &= \boldsymbol{A}_{k+\tau}\boldsymbol{A}_{k+\tau-1}\boldsymbol{A}_{k+\tau-2} \\ &\vdots \\ \boldsymbol{M}_{h_p-1,\tau} &= \boldsymbol{A}_{k+\tau}\boldsymbol{A}_{k+\tau-1}\cdots\boldsymbol{A}_{k+\tau-h_p+2} \end{aligned}$$

for $\tau \in \{1, 2, \ldots, h_p - 1\}$. Matrix $\boldsymbol{P}_{k+1|k}$ is the covariance of the EKF estimation error.

3. PATH PLANNER

The state estimation and trajectory prediction algorithms of section 2 are tested in a path planner which relies on decentralized predictive control to guide a fleet of UAVs in an unknown three-dimensional environment. The mission objective is to ensure that each UAV intercepts the same moving target while avoiding several moving obstacles. UAVs within communication range cooperate to improve the mission performances. A decentralized cooperative control structure regulates the flow of information between UAVs.

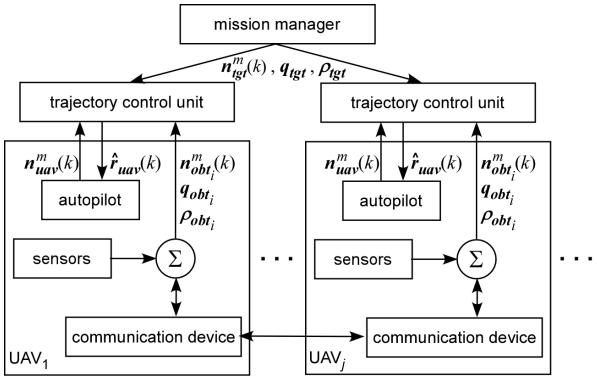


Fig. 2. Path planner.

In this particular path planner (Fig. 2), each UAV is equipped with sensors, a communication device, and an autopilot. As the UAVs navigate in the unknown environment, they detect only the obstacles residing within their respective sensor range. UAVs also exchange obstacle information ($\boldsymbol{n^m_{obt}}$, $\boldsymbol{q_{obt}}$, $\boldsymbol{\rho_{obt}}$: see section 4.1), with UAVs who are in-range only, by means of the communication devices mounted onboard every vehicle. At each sample time, UAVs transmit all acquired obstacle information (from sensors and/or from communicating UAVs) to their personal TCU. TCUs also receive, from their respective autopilot, the measured position of the vehicle at each sample time ($\boldsymbol{n^m_{uav}}$). Furthermore, at each sample time, the mission manager supplies every TCU with information on the tracked target ($\boldsymbol{n^m_{tgt}}$, $\boldsymbol{q_{tgt}}$, $\boldsymbol{\rho_{tgt}}$: see section 4.2). Each TCU then determines its optimal UAV autopilot setpoints ($\boldsymbol{\hat{r}_{uav}}$); setpoints that will guide the vehicle towards the moving target while avoiding moving obstacles. Finally, each autopilot translates the setpoints from its TCU into lower-level commands supplied to the UAV actuators, all while maintaining the stability of the vehicle.

UAVs, obstacles and target are modeled as ellipsoids arbitrarily oriented in space. These ellipsoids can represent the true object shapes or may model safety regions surrounding the objects. The UAV sensor ranges and communication ranges are also modeled as ellipsoids of various sizes and orientations. These ellipsoids are centered upon the UAVs center of mass and mimic the vehicles' movements as they travel through space.

4. TRAJECTORY CONTROL UNIT

Fig. 3 depicts the TCU for a single UAV. This TCU employs an MP control scheme derived from the GlobPC (Desbiens et al., 2000) to guide the UAV in a dynamic environment.

The MP controller depicted in Fig. 3 was originally conceived by Boivin et al. (2008) to guide a UAV in the static environment. In the present paper, path planning for

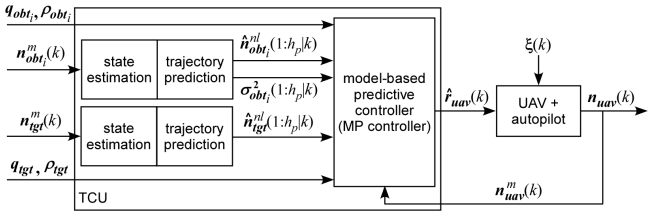


Fig. 3. Trajectory control unit.

UAVs accounts for predicted target and obstacle trajectories (*state estimation* and *trajectory prediction* blocks in Fig. 3). The MP controller selects optimal UAV setpoints ($\hat{\boldsymbol{r}}_{\boldsymbol{uav}}$) that obey the mission objectives quantified by the objective function j defined in section 4.4.

4.1 Obstacle Trajectory Prediction

At each sample time, the UAV transmits the measured location $\boldsymbol{n}^m_{\boldsymbol{obt}}$ (center of mass), and the exact time-invariant shape $\boldsymbol{q}_{\boldsymbol{obt}}$ (semi-axes of ellipsoid) and orientation $\boldsymbol{\rho}_{\boldsymbol{obt}}$ (Euler rotation angles) of known obstacles to its TCU (Figs. 2, 3). The TCU then employs algorithms of section 2 to estimate the states and predict the trajectory of these obstacles' center of mass. Assuming that the i^{th} obstacle motion is governed by (13) and (16), its predicted optimal trajectory over the prediction horizon h_p is

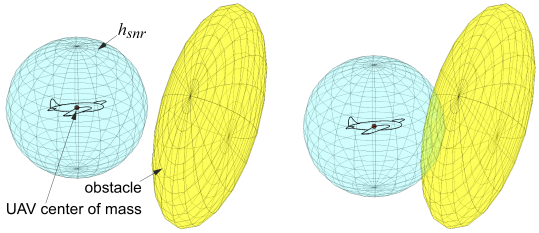
$$\hat{\boldsymbol{n}}^{nl}_{\boldsymbol{obt}_i}(1\!:\!h_p|k) \tag{21}$$

The quality of this trajectory (variance of the position prediction error) is calculated as shown in section 2.2 and yields

$$\boldsymbol{\sigma}^{\boldsymbol{2}}_{\boldsymbol{obt}_i}(1\!:\!h_p|k) \tag{22}$$

Equations (21) and (22) are the basis of the collision avoidance criterion j_{obt} presented in section 4.4.

Obstacle detection scheme An obstacle is detected by the UAV when the vehicle's ellipsoidal sensor range h_{snr} intersects the ellipsoidal obstacle (Fig. 4).



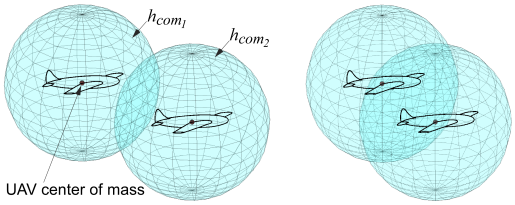
(a) No detection. (b) Detection.

Fig. 4. Obstacle detection scheme.

UAV communication scheme Two UAVs can communicate when the ellipsoids modeling their respective communication ranges h_{com} encompass the other vehicle's center of mass (Fig. 5). UAVs do not communicate if no obstacle has been detected. Furthermore, the knowledge of an obstacle's existence is lost (and thus its trajectory is no longer predicted) if the obstacle exits all sensor ranges.

4.2 Target Trajectory Prediction

At each sample time, a mission manager supplies the TCU with the measured location $\boldsymbol{n}^m_{\boldsymbol{tgt}}$ (center of mass),



(a) No communication. (b) Communication.

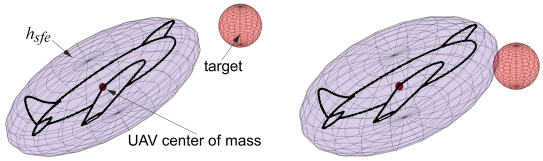
Fig. 5. UAV communication scheme.

and the exact time-invariant shape $\boldsymbol{q}_{\boldsymbol{tgt}}$ (semi-axes of ellipsoid) and orientation $\boldsymbol{\rho}_{\boldsymbol{tgt}}$ (Euler rotation angles) of a dynamic target (Figs. 2, 3). Hence, the target must not necessarily lie within sensor range before it is detected by the UAV. The TCU then employs the algorithms of section 2 to estimate the states and predict the trajectory of the target's center of mass. Assuming that the target's motion is governed by (13) and (16), the predicted optimal target trajectory over the prediction horizon h_p is

$$\hat{\boldsymbol{n}}^{nl}_{\boldsymbol{tgt}}(1\!:\!h_p|k) \tag{23}$$

Equation (23) is the basis of the target tracking criterion j_{tgt} defined in section 4.4.

Target interception scheme The UAV intercepts the target when its safety region h_{sfe} (ellipsoid tightly surrounding the UAV) intersects or encompasses the target (Fig. 6). The mission is deemed complete when all UAVs have intercepted the target.



(a) No interception. (b) Interception.

Fig. 6. Target interception scheme.

4.3 UAV Trajectory Prediction

At each sample time, the MP controller computes the optimal UAV setpoint trajectory on a control horizon h_c ($\hat{\boldsymbol{r}}_{\boldsymbol{uav}}(0\!:\!h_c-1|k)$) that yields the predicted optimal UAV trajectory on the prediction horizon h_p ($\hat{\boldsymbol{n}}_{\boldsymbol{uav}}(1:h_p|k)$). Section 4.4 provides a detailed description of the objective function j governing the choice of the UAV setpoint trajectory.

The predicted UAV trajectory on h_p,

$$\boldsymbol{n}_{\boldsymbol{uav}}(1\!:\!h_p|k) = \boldsymbol{n}^d_{\boldsymbol{uav}}(1\!:\!h_p|k) + \boldsymbol{n}^s_{\boldsymbol{uav}}(1\!:\!h_p|k) \tag{24}$$

is the summation of a deterministic and a stochastic prediction. The motion model governing the deterministic prediction is illustrated in Fig. 1, where $\boldsymbol{G}(z)$ models the closed-loop dynamics of the UAV (i.e. combined UAV and autopilot dynamics). As for the UAV unmeasured disturbances $\boldsymbol{\xi}$ (Fig. 3), they are estimated via an internal model observer. A stochastic predictor then estimates the future unmeasured disturbances which will affect the UAV trajectory ($\boldsymbol{n}^s_{\boldsymbol{uav}}(1:h_p|k)$). An integrating function is used to model the unmeasured perturbations in order to adequately represent non-stationary disturbances.

4.4 Objective Function and Constraints for UAV Path Planning

The MP controller used for UAV path planning is based on the receding horizon principle (Kwon and Han, 2005). At every sample time k, the TCU selects the optimal UAV autopilot setpoints over a chosen control horizon h_c (i.e. $\hat{\boldsymbol{r}}_{\boldsymbol{uav}}(0:h_c-1|k)$). This setpoint trajectory is obtained by minimizing an objective function j while respecting a set of constraints. The objective function is constructed such that the UAV attains the mission objectives, namely the interception of a moving target while avoiding a series of moving obstacles. The constraints ensure that the selected setpoints are confined within the autopilot's limits of operation.

Objective Function The objective function,

$$j(k) = j_r(k) + j_{obt}(k) + j_{tgt}(k) \tag{25}$$

is a summation of a setpoint weighting criterion, a collision avoidance criterion and a target tracking criterion respectively.

The setpoint weighting criterion,

$$j_r(k) = \Delta \boldsymbol{r}_{\boldsymbol{uav}}(0:h_c-1|k)^T \boldsymbol{\Lambda} \Delta \boldsymbol{r}_{\boldsymbol{uav}}(0:h_c-1|k) \tag{26}$$

seeks to minimize UAV setpoint increments over the control horizon h_c. In (26), the vector of setpoint increments $\Delta \boldsymbol{r}_{\boldsymbol{uav}}(0:h_c-1|k)$ are weighted according to $\boldsymbol{\Lambda}$, a $3h_c \times 3h_c$ diagonal matrix.

The collision avoidance criterion,

$$j_{obt}(k) = \sum_{i \in \mathbb{O}} \sum_{\tau=1}^{h_p} \frac{k_1}{\sqrt{|\boldsymbol{M}|}} exp\left\{ -\frac{1}{2} \boldsymbol{\chi}^T (k_2 \boldsymbol{M})^{-1} \boldsymbol{\chi} \right\} \tag{27}$$

seeks to distance the UAV from the predicted obstacle trajectories by minimizing the sum of the Gaussian collision probabilities over the prediction horizon h_p. In (27), $\mathbb{O}$ is the subset of obstacles known by the UAV, $|\cdot|$ denotes the determinant of the matrix, and

$$\boldsymbol{\chi} = \boldsymbol{n}_{\boldsymbol{uav}}(k+\tau|k) - \hat{\boldsymbol{n}}^{nl}_{\boldsymbol{obt}_i}(k+\tau|k) \tag{28}$$

$$\boldsymbol{M} = \boldsymbol{\sigma}^{\boldsymbol{2}}_{\boldsymbol{obt}_i}(k+\tau|k) \tag{29}$$

Moreover, k_1 and k_2 are constants; the former is chosen such that collision avoidance takes precedence over target tracking, while the latter ensures that the potential field j_{obt} surrounds the obstacle. Recall that matrix $\boldsymbol{\sigma}^{\boldsymbol{2}}_{\boldsymbol{obt}_i}$ models the variance on the obstacle's center of mass only, and as such, must be resized in order to account for the obstacle's shape.

The target tracking criterion,

$$j_{tgt}(k) = \sum_{\tau=1}^{h_p} \delta_{uav\text{-}tgt}(k+\tau|k) \tag{30}$$

seeks to minimize the distance between the predicted UAV position and the predicted optimal target position along the prediction horizon h_p. Thus,

$$\delta_{uav\text{-}tgt}(k+\tau|k) = \left\| \boldsymbol{n}_{\boldsymbol{uav}}(k+\tau|k) - \hat{\boldsymbol{n}}^{\boldsymbol{nl}}_{\boldsymbol{tgt}}(k+\tau|k) \right\|_2 \tag{31}$$

where $\|\cdot\|_2$ denotes the vector 2-norm (Euclidean norm).

Constraints Numerical minimization of the cost function j is subject to constraints. Firstly, the UAV setpoint increments on h_c ($\Delta \boldsymbol{r}_{\boldsymbol{uav}}(0:h_c-1|k)$) and UAV setpoints on h_c ($\boldsymbol{r}_{\boldsymbol{uav}}(0:h_c-1|k)$) must be limited in order to ensure the safe operation of the autopilot. Thus,

$$\Delta \boldsymbol{r}^{min}_{\boldsymbol{uav}} \leq \Delta \boldsymbol{r}_{\boldsymbol{uav}}(0:h_c-1|k) \leq \Delta \boldsymbol{r}^{max}_{\boldsymbol{uav}} \tag{32}$$

$$\boldsymbol{r}^{min}_{\boldsymbol{uav}} \leq \boldsymbol{r}_{\boldsymbol{uav}}(0:h_c-1|k) \leq \boldsymbol{r}^{max}_{\boldsymbol{uav}} \tag{33}$$

Secondly, the predicted UAV trajectory $\boldsymbol{n}_{\boldsymbol{uav}}(1:h_p|k)$ should be contained within the span of the vehicle's sensor range h_{snr},

$$\boldsymbol{n}_{\boldsymbol{uav}}(1:h_p|k) \subseteq h_{snr} \tag{34}$$

thus ensuring that a UAV trajectory does not intersect an undetected obstacle.

Thirdly, the UAV speed $\mathcal{S}_{uav}$ (s_{uav} is the xy-plane speed of $\mathcal{S}_{uav}$) must also be constrained along the prediction horizon h_p. Thus,

$$\mathcal{S}^{min}_{uav} \leq \mathcal{S}_{uav}(1:h_p|k) \leq \mathcal{S}^{max}_{uav} \tag{35}$$

where the vehicle's minimum speed is chosen to obey requirements for a safe flight and where the maximum UAV speed must respect the vehicle's physical limitations.

5. SIMULATIONS AND RESULTS

Two test cases involving the path planner described in sections 3 and 4 are now shown. These test cases demonstrate the performance of the path planner when UAVs accomplish their mission with and without cooperation.

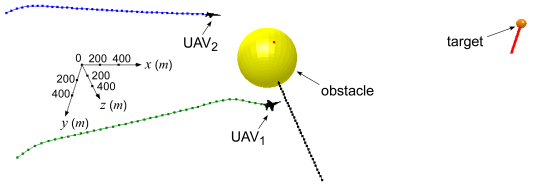
The mission scenario for both test cases involves two UAVs, one spherical target of $50\,m$ radius and one spherical obstacle of $300\,m$ radius. In the first test case, each UAV is surrounded by a spherical safety region and a spherical sensor range of $10\,m$ and $550\,m$ radii respectively. In the second test case, an additional sphere of radius $1500\,m$, denoting the communication horizon, surrounds each UAV. The UAVs, obstacle and target initial conditions are identical in both test cases. The displacements executed by the obstacle and target are also identical in both test cases. The target setpoints throughout both simulations are maintained at $\boldsymbol{r}_{\boldsymbol{tgt}} = [20\,km/h \;\; -60^\circ \;\; -400\,m]^T$. The obstacle setpoints up to sample time $k = 39$ are $\boldsymbol{r}_{\boldsymbol{obt}} = [60\,km/h \;\; -110^\circ \;\; -400\,m]^T$. At sample time $k = 40$ the obstacle heading is modified to $r_{obt_\theta} = -70^\circ$. The sampling period $t_s = 2\,s$ is used for both simulations.

5.1 Test Case 1: No Cooperation between UAVs

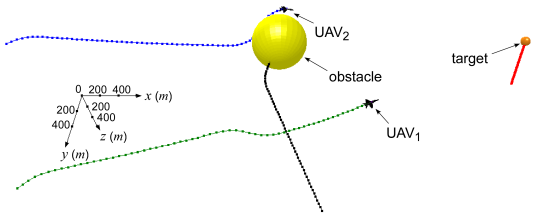
Figure 7 illustrates the performance of the path planner when UAVs do not exchange obstacle information. By time $t = 80\,s$, UAV_1 has already detected the obstacle and taken a course of action which avoids the obstacle's predicted trajectory. At this time, UAV_2 has not detected the obstacle and thus its trajectory remains unaffected by the oncoming obstacle. By time $t = 120\,s$, one notes that UAV_2 was forced to execute drastic maneuvers to avoid the oncoming obstacle. Consequently, UAV_2 will intercept the moving target after UAV_1 (see Fig. 7(c)). The total mission time (time required for both UAVs to intercept the target) for this simulation is $t_{msn} = 230\,s$.

5.2 Test Case 2: Cooperation between UAVs

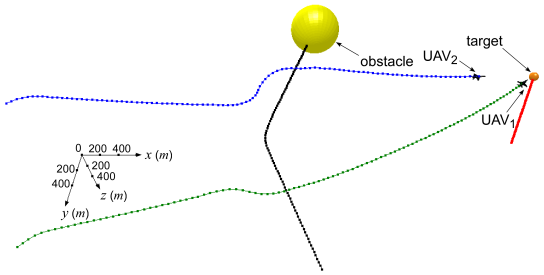
Figure 8 illustrates the performance of the path planner when UAVs in communication range exchange obstacle



(a) Time $t = 80\,s$ (sample time $k = 40$).

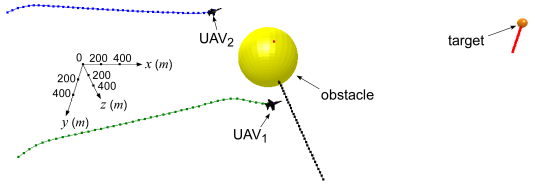


(b) Time $t = 120\,s$ (sample time $k = 60$).

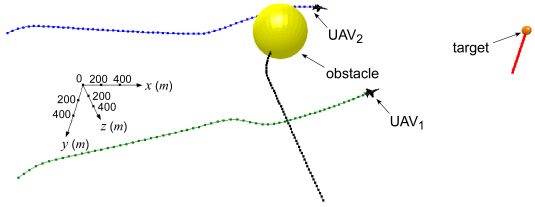


(c) Time $t = 200\,s$ (sample time $k = 100$).

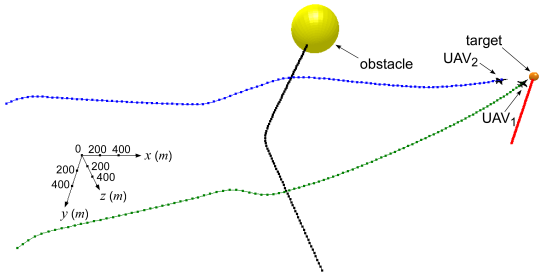
Fig. 7. Simulation: path planning without cooperation.



(a) Time $t = 80\,s$ (sample time $k = 40$).



(b) Time $t = 120\,s$ (sample time $k = 60$).



(c) Time $t = 200\,s$ (sample time $k = 100$).

Fig. 8. Simulation: path planning with cooperation.

information. Since UAV_1 is the first vehicle to detect the obstacle, its trajectory remains unchanged. By time $t = 80\,s$, UAV_1 has communicated the obstacle's presence to UAV_2; one can see how UAV_2 has begun to adjust its trajectory to avoid the oncoming obstacle. By time $t = 120\,s$, UAV_2 had plenty of time to plot an evasive course of action. As a result, this UAV's chosen trajectory is much smoother. At time $t = 200\,s$ one notes that UAV_2 will intercept the moving target faster than in the first test case. The total mission time for this particular simulation is $t_{msn} = 220\,s$.

6. FUTURE WORK

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. Data reconciliation will also be added to deal with repeated measurements of the same object taken by UAVs in communication. Finally, the proposed algorithms shall be tested in a Hardware-in-the-Loop system (Manaï et al., 2005) and on a real UAV.

REFERENCES

Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, Inc., New York, 2001.

E. Boivin, A. Desbiens, and E. Gagnon. UAV collision avoidance using cooperative predictive control. Submitted for publication at the Mediterranean Conference on Control and Automation, Corsica, France, June 2008.

A. Desbiens, D. Hodouin, and E. Plamondon. Global predictive control: A unified control structure for decoupling setpoint tracking, feedforward compensation and disturbance rejection dynamics. In *IEE Proceedings - Control Theory and Applications*, volume 147, pages 465–475, July 2000.

A. Dogan and U. Zengin. Unmanned aerial vehicle dynamic-target pursuit by using probabilistic threat exposure map. *AIAA Journal of Guidance, Control, and Dynamics*, 29(4):944–954, July–August 2006.

A. Elnagar. Motion prediction of moving objects. In *Proceedings of the Second IASTED International Conference of Control and Applications*, pages 448–451, 1999.

W. H. Kwon and S. Han. *Receding Horizon Control*. Springer-Verlag, London, first edition, 2005.

M. Manaï, A. Desbiens, and E. Gagnon. Identification of a UAV and design of a hardware-in-the-loop system for nonlinear control purposes. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005. San Francisco, USA, Paper 2005-6483.

L. R. Newcome. *Unmanned Aviation: A Brief History of Unmanned Aerial Vehicles*. AIAA General Publication Series, Reston Virginia, first edition, 2004.

R. W. Penney. Collision avoidance within flight dynamics constraints for UAV applications. *Aeronautical Journal*, 109(1094):193–199, April 2005.

C. Prévost, A. Desbiens, and E. Gagnon. Extended Kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle. In *Proceedings of the 2007 American Control Conference*, pages 1805–1810, July 2007.