2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Aircraft Collision Avoidance using Spherical Visual Predictive Control and Single Point Features

Aaron Mcfadyen*, Luis Mejias*, Peter Corke* and Cédric Pradalier**

Abstract— **This paper presents practical vision-based collision avoidance for objects approximating a single point feature. Using a spherical camera model, a visual predictive control scheme guides the aircraft around the object along a conical spiral trajectory. Visibility, state and control constraints are considered explicitly in the controller design by combining image and vehicle dynamics in the process model, and solving the nonlinear optimization problem over the resulting state space. Importantly, range is not required. Instead, the principles of conical spiral motion are used to design an objective function that simultaneously guides the aircraft along the avoidance trajectory, whilst providing an indication of the appropriate point to stop the spiral behaviour. Our approach is aimed at providing a potential solution to the *See and Avoid* problem for unmanned aircraft and is demonstrated through a series of experimental results using a small quadrotor platform.**

I. INTRODUCTION

Recent technological advances have seen unmanned aircraft (UAS) emerge in the civilian sector, offering significant economic and social benefits to an increasingly diverse set of applications. Many of these require operation in the national airspace during some or all flight phases, imposing an additional set of requirements on aircraft capability [1]. Of particular importance is a collision avoidance solution aimed at replicating *See and Avoid* behaviour in conventionally-piloted aircraft. In short, this is a form of decentralized short term collision avoidance in which the pilot must independently detect and avoid any unplanned hazard, be it static or dynamic [2]. To ensure compliance with strict safety standards, international regularity bodies[1] require any automated system to demonstrate an equivalent level of safety to manned aircraft [3]. This presents a set of challenging problems regarding object detection and conflict resolution.

A natural choice for detection and tracking is the use of passive, uncooperative sensors such as video cameras [4]. They can be fitted to any aircraft regardless of size, weight and power restrictions, whilst providing comparable initial detection distances to human pilots [5]. Initially, a distant object appears as a small, low contrast, slow moving point feature in the image until such time when the collision is potentially unavoidable. As such, reliably estimating object range is difficult. Lack of distinguishable shape, size and growth rate in the image renders approaches based on visual looming infeasible. Stereo vision will likely fail as the ratio of camera baseline to object range is small. Passive ranging techniques induce predefined motion to estimate range, which will decrease available avoidance time and potentially degrade collision geometry [6]. Alternatively, using relative angular rates to identify a collision threat has been suggested, but results show considerable performance issues [7]. However, relative angular position can be estimated and tracked with greater consistency [8] [9].

The limitations on object detection restrict the feasible set of conflict resolution approaches available. Without range, a large number of well established schemes [10] become inappropriate, prompting new methods. In particular, using angular measurements directly as feedback strongly resembles pilot behaviour, and is consistent with human visual navigation in a collision scenario [11]. It makes sense then to adopt such an approach for conflict resolution. To be effective however, a number of issues still need to be addressed when considering the *See and Avoid* environment. A large camera field of view is required to ensure that the target remains visible throughout the encounter. Using only a single point feature restricts the degrees of freedom that can be controlled, yet both vertical and lateral avoidance is desirable to increase the potential miss distance. At the same time, vehicle dynamics and actuator limitations need to be considered. Lastly, we need a means to cease avoidance behaviour and return to the original trajectory or reference flight condition without relying on range.

In this paper we address these issues for conflict resolution, having assumed object detection. We derive a novel vision based collision avoidance controller using a combination of spherical imaging, properties of conical spirals and visual predictive control (image-based visual servoing using model predictive control), resulting in the following contributions:

1) Extension of visual predictive control to spherical imaging using a single point feature.
2) Novel visual predictive control design that exploits the properties of conical spiral motion to ensure collision avoidance without estimating object range.
3) First practical implementation of visual predictive control for aerial vehicles

The paper is organized as follows. In section II we provide the problem background. In section III we explain the problem preliminaries, including conical spirals and spherical imaging. We derive the spherical visual predictive controller for collision avoidance of single targets in section IV and demonstrate the approach with a set of experiments using a small unmanned quadrotor in section V. Lastly, section VI presents conclusions and ongoing work.

* ARCAA, School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, Australia. aaron.mcfadyen@qut.edu.au
** Autonomous System Laboratory, ETH Zurich, Switzerland
[1]FAA (USA), CASA (Australia) and EuroControl (Europe)

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

II. Background & Related Work

The use of visual information to control a robot to perform a specific task is referred to as visual servoing [12]. Position-based visual servoing (PBVS) relies on recovering target pose estimates to derive feedback control in the task space. Multiple image feature points of the same object are required however, which violates our problem constraints. Classical image-based visual servoing (IBVS) approaches do not require pose estimation and provide an inexpensive, reactive control solution with inherent robustness to range and camera calibration errors. Feedback is derived directly from the image, similar to how a pilot may use visual cues to avoid collision, making it an attractive solution for *See and Avoid*.

The fundamental differences for our problem are the number of image features and observable image feature velocity or optic flow. Firstly, three or more feature points of the same object are typically required to control motion in all six camera degrees of freedom (DOF). With only a single feature point we may only control two DOF, so approaches reported in [13] are inappropriate. Secondly, due to the inverse relationship between range and feature velocity, we are unable to reliably observe optic flow corresponding to relative translational motion. This would require near and/or large objects, so collision avoidance using such principles alone would likely fail in our case [14].

The most applicable approach involves positioning the object or point in the image at a constant non-zero angular displacement from the optical axis. This induces circular or spiral-like motion about the object, and has been demonstrated in nature by insects and birds of prey [15]. The resulting trajectories circumscribe the surface of a cone, giving rise to the nomenclature conical spiral. Using variations of the classical IBVS control structure, lateral avoidance [16] [17] and more recently both lateral and vertical avoidance [18] have been reported. Such approaches still suffer from common drawbacks. In particular, using perspective or wide-angle cameras constrains the visible region, and therefore limits the set of conical spirals that can be followed. One way to manage such visibility issues is to use a spherical camera [19]. This was first exploited for aircraft, offering only lateral or vertical avoidance [20]. This was extended in [21] for both lateral and vertical avoidance and demonstrated experimentally using a quadrotor. Unfortunately any approach based on classical IBVS structure cannot explicitly consider vehicle dynamics and actuator limitations in the control design.

To incorporate such state and control constraints, optimal control based approaches such as visual predictive control (VPC) can be used [22]. Based on well established nonlinear model predictive control (NMPC) strategies [23], the structure is the same but the nonlinear optimization problem can now be defined over the image space or both image and state space. This allows consideration of not only state and control, but visibility constraints directly in the control design. If using a spherical camera, the visibility constraint could be used to ensure particular regions of the sphere are avoided. Additionally, robustness to measurement noise and small model mismatch has been demonstrated whilst providing decoupling effects. The approach lends itself naturally to our problem environment, yet few have realised its potential. In [24], angular position was used to avoid multiple objects however the approach was not flown and accurate range estimates were required.

As a final point, if indeed a stopping criteria is used, range is often required to cease avoidance behaviour at an appropriate point or time [16] [24]. Alternately, conditions on the heading or altitude [18] [20] [21] can be used but essentially decouple the stopping criteria from the visual control. As such, the avoidance behaviour could potentially stop before the reference image features are reached.

Considering the above, we cast the collision avoidance task into the visual predictive control framework. However, instead of relying on the control design to ensure visibility constraints, a spherical camera model is used to circumvent the issue and ensure the object remains visible. The state vector and objective function are then designed to simultaneously guide the aircraft along a safe spiral trajectory, whilst providing an indication of an appropriate point to stop avoidance. The stopping condition is thus coupled with the image features to avoid potential premature stopping.

III. Problem Preliminaries

A. Spherical Cameras

Spherical cameras provide a 4π steradian field of view with each image feature coordinate ($\mathbf{s}$) defined by an angle of colatitude (σ) and azimuth (γ), where $\sigma \in [0,\pi)$ and $\gamma \in [-\pi,\pi)$. For practical implementation, we consider a virtual spherical camera aligned to the body axis such that $\mathbf{s} = (\frac{\pi}{2}, 0)$ corresponds to the optical axis. We then approximate a section of the sphere with a real perspective camera. The optical axis (z_c') of the real camera is shown in figure 1(a). Using the unified imaging model [25] we can transform image features from perspective, catadioptric or fisheye to the unit sphere. In this case, assuming the spherical centre coincides with the focal point, the conical angles can be approximated by

$$\sigma \approx \arctan\left(\frac{v-\frac{h}{2}}{f}\right)+\frac{\pi}{2} \quad (1)$$

$$\gamma \approx \arctan\left(\frac{u-\frac{w}{2}}{f}\right)+\frac{3\pi}{4} \quad (2)$$

where f is the focal length, w is the image width, h is the image height and u and v are the pixel locations of the point in the x and y dimensions respectively.

B. Conical Spirals

A conical spiral describes motion about the surface of a regular cone according to a set of logarithmic equations. They are parametrized by a fixed set of conical angles, namely bearing (α) and elevation (β). The angles are measured in a spiral reference frame with respect to the apex such that $\beta \in [0,\pi)$ and $\alpha \in [-\pi,\pi)$. The spiral reference

frame has its origin attached to the body frame, but is free to rotate about its z axis only. Fixing β defines the particular cone on which the trajectory will lie, except in the degenerate case for $\beta = \frac{\pi}{2}$. Fixing α determines the specific trajectory on the cone. Some examples are shown in figure 1(b) and 1(c), with the equations of motion defined in [15]. Choosing $\beta \neq 0$ and $\|\alpha\| \geq \frac{\pi}{2}$ can ensure lateral and vertical collision avoidance for a static object, by forcing a divergent conical spiral that moves away from the apex. Such a trajectory is shown in black in figure 1, where the black square represents the conical apex. To track the reference conical angles, a transformation between the image features and conical angles is thus required.

Consider a spherical camera rigidly attached to an aircraft spiralling about an object, $\mathbf{p}$. The image features change with position and orientation but the conical angles will depend only on position and yaw angle. The object can be expressed in the camera frame by ${}^c\mathbf{p}$ such that

$$ {}^c\mathbf{p} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} r\sin\sigma\cos\gamma \\ r\sin\sigma\sin\gamma \\ r\cos\sigma \end{pmatrix} = r\,{}^c\tilde{\mathbf{p}} \tag{3} $$

where a tilde denotes a coordinate set unscaled by the range (r). In the spiral reference frame, equation (3) becomes

$$ {}^s\mathbf{p} = {}^bR_s\,{}^cR_b\,{}^c\mathbf{p} + {}^c\mathbf{t}_s \tag{4} $$

$$ r\,{}^s\tilde{\mathbf{p}} = r\,{}^bR_s\,{}^cR_b\,{}^c\tilde{\mathbf{p}} + {}^c\mathbf{t}_s \tag{5} $$

where ${}^bR_s$ and ${}^cR_b$ define rotation matrices from body to spiral and camera to body frames respectively. The camera focal point and body centre of mass (and thus spiral frame origin) are separated by the vector ${}^c\mathbf{t}_s$. Dividing through by r and assuming $r \gg \|{}^c\mathbf{t}_s\|$ and ${}^cR_s = {}^bR_s{}^cR_b$, we obtain

$$ {}^s\tilde{\mathbf{p}} = {}^cR_s\,{}^c\tilde{\mathbf{p}} \tag{6} $$

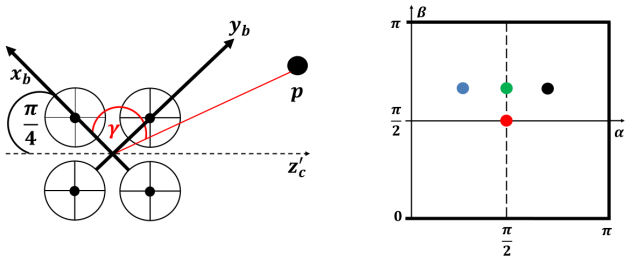
Expanding using spherical coordinates

$$ \begin{pmatrix} \cos\beta'\cos\alpha \\ \cos\beta'\sin\alpha \\ \sin\beta' \end{pmatrix} = {}^cR_s \begin{pmatrix} \cos\sigma\cos\gamma \\ \cos\sigma\sin\gamma \\ \sin\sigma \end{pmatrix} \tag{7} $$

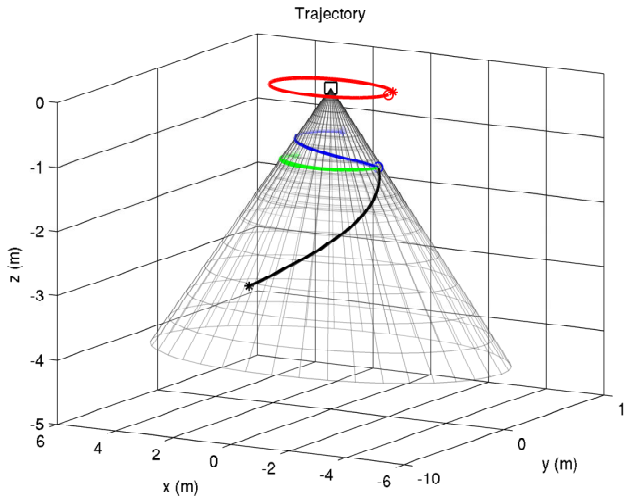
where $\beta' = \pi - \beta$ and

$$ {}^cR_s = \begin{pmatrix} \cos\theta & \sin\theta\sin\phi & \cos\phi\sin\theta \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{pmatrix} \tag{8} $$

Knowing our orientation in pitch (θ) and roll (ϕ), we can solve for the conical angles (α, β). Alternately, if we measure the image features from a de-rotated image in pitch and yaw then the conical angles are in fact the measured image features, and we can control the conical angles directly from visual feedback. As such, the reference image features determine the type of conical spiral trajectory.



(a) Camera orientation (b) Conical angles



(c) Conical spirals

Fig. 1. Camera orientation and example conical spirals including converging (*blue*), diverging (*black*), degenerate case for $\beta = \frac{\pi}{2}$ (*red*) and special case for circular motion (*green*).

C. Dynamics

To apply the visual predictive control framework and include state constraints, the aircraft dynamics must be augmented with the image dynamics to derive the process model used for prediction. Using a point mass model and black box identification for a small AscTec Hummingbird quadrotor, a simplified set of decoupled linear equations using 0^{th} and 1^{st} order approximations were found. The dynamic equations in the body frame are given by

$$ \dot{v}_z = -g + \frac{u_T}{m} \tag{9} $$

$$ \dot{\omega}_z = -\lambda\omega_z + \lambda u_\omega \tag{10} $$

$$ \dot{\psi} = \omega_z \tag{11} $$

$$ \dot{v}_x = -\frac{T}{m}u_\theta \tag{12} $$

$$ \dot{v}_y = -\frac{T}{m}u_\phi \tag{13} $$

where $[v_x\ v_y\ v_z]$ denote translational velocity components, ω_z the angular velocity about the z axis and ψ the yaw angle. The controls $[u_T\ u_\omega\ u_\theta\ u_\phi]$ define thrust, yaw rate, pitch and roll commands. The empirically derived damping constant $\lambda = 5.9$, m is the mass, g is the acceleration due to gravity and T is the thrust. Partitioning the dynamics into the x and y axis components and the z axis components, we define $\mathbf{z} =$

$[v_z\ \omega_z\ \psi]$ and $\mathbf{z}' = [v_x\ v_y]$ where $'$ in this case does not imply derivative, but used for notational convenience. Similarly, we define $\mathbf{u} = [u_T\ u_\omega]$ and $\mathbf{u}' = [u_\theta\ u_\phi]$ such that complete control of the quadrotor is defined by $\hat{\mathbf{u}} = [\mathbf{u}\ \mathbf{u}']$.

The image dynamics relate the camera velocity ($\mathbf{v}$) to the image feature velocity ($\dot{\mathbf{s}}$). For a spherical projection model

$$\dot{\mathbf{s}} = L_s^* \mathbf{v} \tag{14}$$

where L_s^* defines the spherical image Jacobian using a fixed reference range value (r^*) and is given by

$$L_s^* = \begin{pmatrix} \frac{-\cos\sigma\cos\gamma}{r^*} & \frac{-\cos\sigma\sin\gamma}{r^*} & \frac{\sin\sigma}{r^*} & \sin\gamma & -\cos\gamma & 0 \\ \frac{\sin\gamma}{r^*\sin\sigma} & \frac{-\cos\gamma}{r^*\sin\sigma} & 0 & \frac{\cos\gamma\cos\sigma}{\sin\sigma} & \frac{\sin\gamma\cos\sigma}{\sin\sigma} & -1 \end{pmatrix} \tag{15}$$

By aligning the camera and body frame, the camera and body velocities are equal such that $\mathbf{v} = [v_x\ v_y\ v_z\ \omega_x\ \omega_y\ \omega_z]^T$, where ω_x and ω_y are the roll and pitch angular velocity. For conical spiral motion, we require a constant forward velocity and zero y translational velocity. If we then measure the image features from a de-rotated image, we can safely assume $\omega_x \approx \omega_y \approx v_y \approx 0$. In its expanded form, (14) then reduces to

$$\dot{\sigma} = \frac{-\cos\sigma\cos\gamma}{r^*} v_x + \frac{\sin\sigma}{r^*} v_z \tag{16}$$

$$\dot{\gamma} = \frac{\sin\gamma}{r^*\sin\sigma} v_x - \omega_z \tag{17}$$

We can now augment (9) - (11) with (16 - 17) to define the nonlinear state equations for the combined process model

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{18}$$

$$\mathbf{x} = [\mathbf{z}\ \mathbf{s}] = [\psi\ v_z\ \omega_z\ \sigma\ \gamma] \tag{19}$$

where $\mathbf{x}$ denotes the system state and $\mathbf{f}$ the set of dynamic equations. Notice that v_x must be passed as a parameter and controlled independently. Specifically, the remaining quadrotor states ($\mathbf{z}'$) are controlled using $\mathbf{u}'$, which can be derived from the reference x and y translational velocities. As linearisation of the quadrotor dynamics was performed around hover, a set of feed forward terms can then be used to minimize any coupling effects using the current roll and pitch angles.

IV. Spherical Visual Predictive Control

The basic idea behind many VPC and NMPC schemes is the same. At each sampling time (k) an optimal control sequence $U = \{\bar{\mathbf{u}}_k, \ldots, \bar{\mathbf{u}}_{k+N-1}\}$ is found by minimizing an objective function (J_N) over a finite horizon (N), subject to the system dynamics and constraints. Only the first element of the control policy ($\bar{\mathbf{u}}_k$) is implemented before new measurements are received. The process model is then re-initiated with the new measurements, the horizon is moved one step forward and the process repeats. The bar denotes an internal variable, calculated over the prediction horizon using the process model. This is required as the actual and predicted states and controls can, and will in general, be different to the real system.

The main differences between various formulations include how the process model (and thus state vector) is defined, the objective function structure and associated constraint domain. In our case, the spherical image features augment part of the quadrotor state vector and the optimization is performed over the combined state and image space. Having defined the process model (19), we can now define our objective function, associated constraints and resulting control problem. At each sampling time find $\bar{\mathbf{u}}_k$ that corresponds to J^{OPT} such that

$$J^{OPT}(\mathbf{x}_k, \bar{\mathbf{u}}_k) = \min_U \{J_N(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k)\} \tag{20}$$

$$\text{s.t. } \bar{\mathbf{x}}_{k+1} = \mathbf{f}(\bar{\mathbf{x}}_k,\ \bar{\mathbf{u}}_k), \quad \forall k, \ldots, N_p - 1$$

$$\bar{\mathbf{x}}_k = \mathbf{x}_k$$

where $N_p = k + N$ and

$$J_N(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) = E(\bar{\mathbf{x}}_{N_p}) + \sum_k^{N_p - 1} F(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) \tag{21}$$

$$F(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) = (\bar{\mathbf{x}}_k \ominus \mathbf{x}^*)^T Q (\bar{\mathbf{x}}_k \ominus \mathbf{x}^*) + \bar{\mathbf{u}}_k^T R \bar{\mathbf{u}}_k \tag{22}$$

$$E(\bar{\mathbf{x}}_{N_p}) = (\bar{\mathbf{x}}_{N_p} \ominus \mathbf{x}^*)^T P (\bar{\mathbf{x}}_{N_p} \ominus \mathbf{x}^*) \tag{23}$$

The reference state is given by $\mathbf{x}^*$ and $\ominus$ denotes the modulo 2π subtraction required to bound the spherical image feature error. The weighting matrices are defined such that $Q \succ 0$, $R \succeq 0$ and $P \succeq 0$ where $\succ$ and $\succeq$ denote positive definiteness and semi-definiteness respectively. The terminal penalty matrix (P) is used to penalize deviation from the reference state at the end of the prediction horizon. Recalling $\mathbf{x}_k = [\mathbf{z}_k\ \mathbf{s}_k]$, the optimization is performed with respect to the constraint domain $\mathbf{K}$, defined by

$$\mathbf{K} = \begin{cases} \mathbf{s}_k \in \mathbb{R}^2 \,|\, \mathbf{s}_{min} \leq \mathbf{s}_k \leq \mathbf{s}_{max} & \text{(24a)} \\ \mathbf{u}_k \in \mathbb{R}^2 \,|\, \mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} & \text{(24b)} \\ \mathbf{z}_k \in \mathbb{R}^3 \,|\, \mathbf{z}_{min} \leq \mathbf{z}_k \leq \mathbf{z}_{max} & \text{(24c)} \end{cases}$$

Visibility constraints (24a) are somewhat handled with the application of a spherical camera, so are used to avoid the polar regions and ensure the spherical image Jacobian is always defined. The control constraints (24b) bound the quadrotor thrust and yaw rate commands to an admissible region. The state constraints (24c) ensure controls are issued such that the quadrotor state lies inside a desirable region.

In the following sections, the sampling period is set to 0.04 and the control horizon is equal to the prediction horizon of 10. This is chosen for stability and computational reasons [23].

A. Collision Avoidance

Traditionally, Q and R are used to provide a performance trade off in reaching the desired state without excessive control energy. By including the yaw angle in the state vector and placing a small weight on the corresponding Q entry, solving (20) forces the aircraft to implicitly prefer a spiral trajectory. The objective function minimum is reached when the aircraft reaches its initial heading whilst established on the spiral trajectory, then increases as the spiral is continued. So the objective function value provides an indication of an appropriate point to stop spiral behaviour and resume a normal flight, without estimating range. The remaining

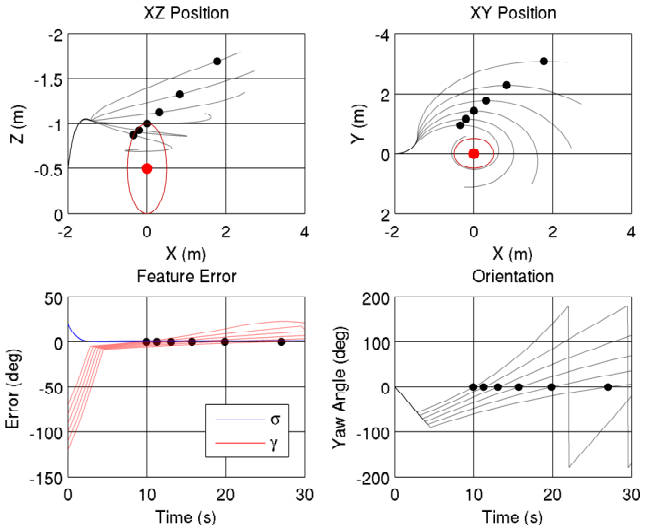


Fig. 2. Position, image feature error and yaw angle for $\gamma^* = 70^\circ$ to 120° at 10° increments with $\sigma^* = 70^\circ$ and $\psi^* = 0$. Black dots mark J^{OPT}, red dots the object position with embracing circle a safety sphere of 0.5m radius.

entries in Q and R are kept comparatively low, having enforced associated control and state constraints in (24b) and (24c). Figure 2 show this concept in simulation for various reference image features ($\mathbf{s}^*$) where $\mathbf{s}^* = [\sigma^* \ \gamma^*]$.

Obviously the objective function will never reach zero considering the image noise characteristics and flight disturbances. This means the objective function value has to be thresholded, either on the absolute value or the derivative. For this paper, we apply a small threshold (ε) on the absolute value and when reached, resume the initial flight condition before object detection. In this way, the image features are coupled to the stopping criteria and range is not required.

Although $\mathbf{s}^*$ may be chosen freely, it should be selected based on the objects image position upon initial detection. We take a similar approach to [21] to ensure the object does not cross in front of the aircraft. We select $\sigma^* \neq \frac{\pi}{2}$ and $||\gamma^*|| \geq \frac{\pi}{2}$ to ensure a divergent conical spiral is flown.

V. Results

For safety, repeatability and regulatory reasons, we provide experimental results for a small AscTec Hummingbird quadrotor operating indoors. The ACADO Toolkit was used to solve the nonlinear optimization using sequential quadratic programming, with the advantage of directly generating deployable embedded code optimized for our application [26]. Simulations were first performed in MATLAB to verify the control design and obtain approximate cost function parameters and nonlinear constraint sets for the real platform.

A Vicon motion capture system was used to measure the quadrotor position and yaw angle only, replacing a typical GPS/INS system found on outdoor UAS. Two LQRI controllers and EKF filters were derived to estimate and regulate the lateral translational velocity based on (12)-(13). Additional EKF filters were derived to estimate vertical velocity and yaw rate. The image features were obtained directly from camera feedback or estimated using a virtual point. They are de-rotated at each iteration, keeping the process model in the controller accurate. The control design is shown in figure 3.

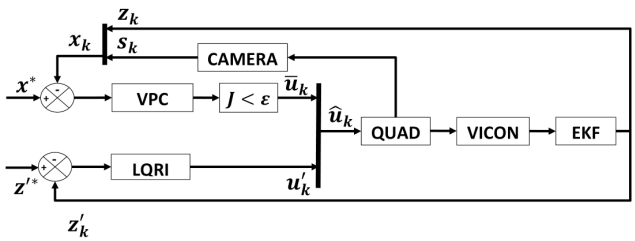


Fig. 3. Quadrotor control architecture. Recalling, $\mathbf{z}$ and $\mathbf{z}'$ are the quadrotor states and $\mathbf{s}$ the spherical image features. Controls are defined by $\mathbf{u}$ and $\mathbf{u}'$.

Important to note, we do not use velocity estimates from the Vicon only position. As such, Vicon feedback is not used in the visual control except for the case where a virtual point is required. This is due to field of view limitations using a perspective camera to approximate a spherical imaging section as shown in figure 1(a).

A. Spiral Tracking

A simple detection algorithm was used to detect a stationary object, with the centre of the detected region used to approximate a point feature [27]. In this way detection is relatively consistent with image processing delay and noise present, helping to model a realistic scenario. Example results using object detection and a virtual point are shown in figure 4.

The spiral is tracked with reasonable accuracy in both lateral and vertical planes in both cases, with smoother control resulting when a virtual point is used. The discrepancy can be attributed to image processing delay and unmodelled disturbances resulting in larger fluctuations in control. The affect translates to small consistent oscillations about the reference azimuth. Control constraints have been enforced, keeping the quadrotor in a stable flight configuration in both cases, despite image delay.

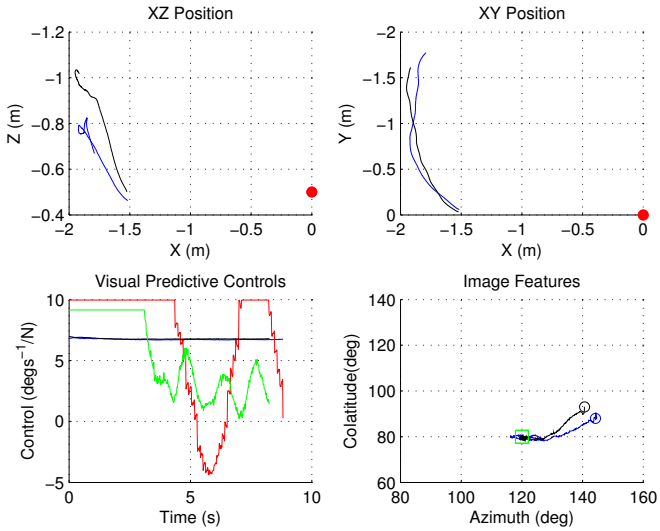


Fig. 4. Position, controls and image features for $\gamma^* = 120^\circ$, $\sigma^* = 80^\circ$ using a camera (*black*) and virtual point (*blue*). The green square denotes $\mathbf{s}^*$ and red dot object position. Only VPC controls, u_ω and u_T are given for camera (*red* and *black*) and virtual point (*green* and *blue*) respectively.

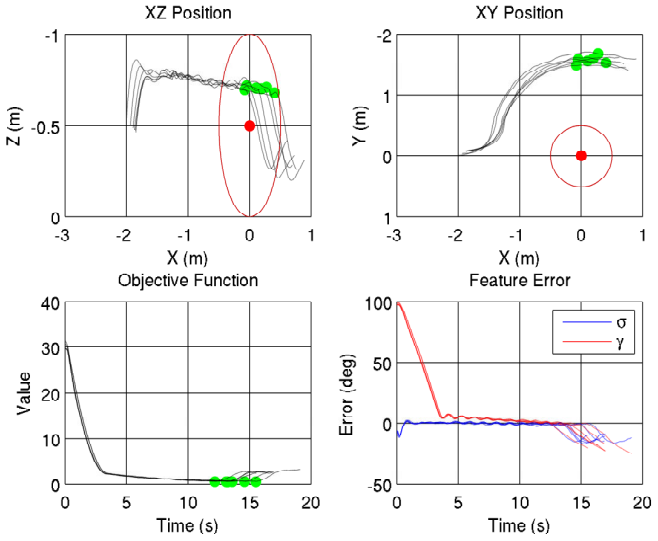


Fig. 5. Position, objective function and feature error for $\gamma^* = 110°$, $\sigma^* = 80°$ and $\varepsilon = 0.6$.

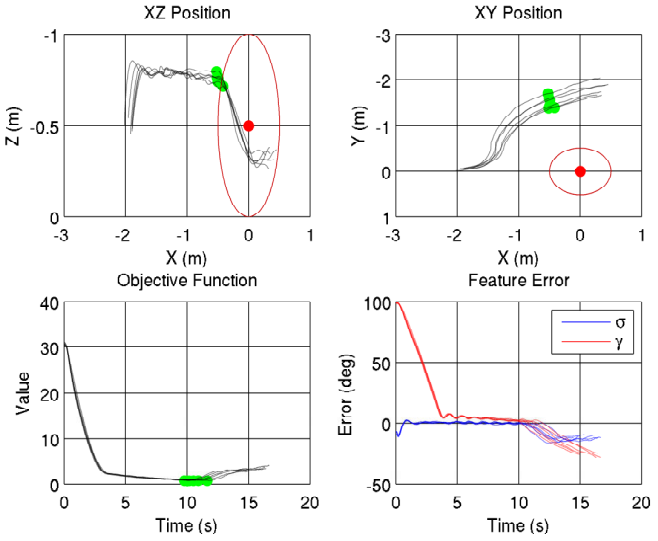


Fig. 6. Position, objective function and feature error for $\gamma^* = 110°$, $\sigma^* = 80°$ and $\varepsilon = 0.8$.

B. Collision Avoidance

A virtual point was used for collision avoidance to ensure a divergent spiral can be tracked from any initial position, and therefore ensure collision avoidance. The quadrotor moves at a fixed forward velocity of 0.2ms^{-1} with time to collision between 10 and 20 seconds. The object initially appears at a height of 0.5m and displaced laterally by 2.0m. The desired range value (r^*) used in the controller is set to 2m. Clearly, scaling these dimensions by a factor of 100, a realistic scenario can be created. In the remaining figures, the red dot denotes the object position with embracing circle a safety sphere of 0.5m radius.

Figure 5 and 6 show the avoidance behaviour for two different threshold values with same reference image features and Q, P and R matrices. In figure 7 different reference image features and weighting matrices and used, resulting in a different objective function threshold. In all cases the green solid circles denote the first time at which the threshold has been exceeded such that $J^{OPT} \leq \varepsilon$. When the green dot appears at the end of the trajectory, then $J^{OPT} > \varepsilon$ and the threshold has not been exceeded yet. In this case, continuing the flight would force the quadrotor to spiral the object continuously until $J^{OPT} \leq \varepsilon$, provided $\gamma^* \geq 90°$. Otherwise a collision may result if the quadrotor dynamics and state constraints allow large velocities[2].

To test the controllers' inherent robustness to small disturbances, we simulate a set of dynamic objects with constant velocity given by $\mathbf{v}_t$ such that $||\mathbf{v}_t|| \leq ||\mathbf{v}||$. This is essentially similar to adding a small disturbance to the translational velocity components of the process model. Without altering the process model, the results are shown in figure 8.

In most cases, the controller is able to reach the desired image features for both static and dynamic objects, regardless of the reference value. Avoidance is stopped at an appropriate

[2]Large velocities are required for a converging spiral to eventually reach the object, otherwise a limit is reached and the object is circled.

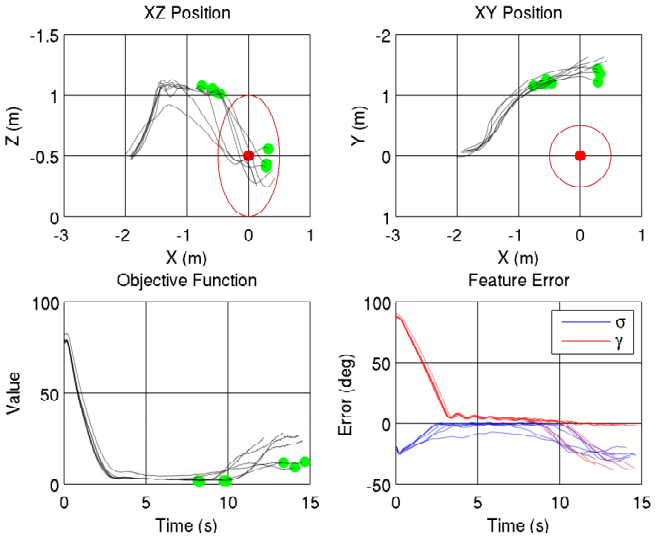


Fig. 7. Position, objective function and feature error for $\gamma^* = 90°$, $\sigma^* = 70°$ and $\varepsilon = 1.9$.

time and could be improved with further tuning of the objective function threshold. Although a large number of diverse collision avoidance scenarios would be required to optimize the threshold, initial results demonstrate the approaches feasibility and suggest a more conservative value should be chosen considering the safety implications. The threshold is largely dependent on the weighting matrices and not the reference image features. As the matrices are directly related to aircraft performance, one could imagine they would be fixed for a given airframe. So only the threshold needs to be tuned, regardless of the collision avoidance scenario.

VI. Conclusion & Further Work

In this paper we used a spherical camera model and visual predictive control to derived an intuitive collision avoidance controller, free of range estimation, applicable to the *See and Avoid* problem. The design structure lends itself naturally to the problem, managing platform constraints and modest image delay and uncertainties. This in turn allows

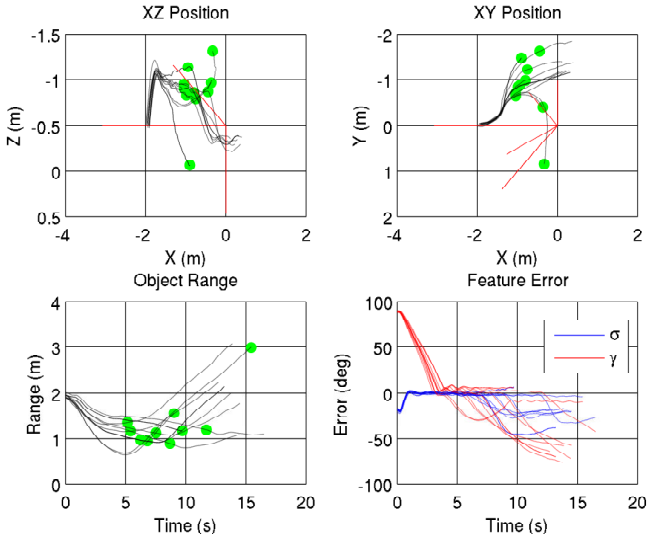


Fig. 8. Position, object range and feature error for $\gamma^* = 90°$, $\sigma^* = 70°$ and $\varepsilon = 1.9$. All object trajectories start at $(0,\ 0,\ -0.5)$ and are shown in red in the two upper plots. There are 5 different object trajectories, including head-on and crossing with the remaining including climb or descent. Each are simulated at two separate $\mathbf{v}_t$ values, resulting in 10 collision scenarios.

consideration of static and slow moving targets. Notably, the first practical implementation of visual predictive control for aerial vehicles has resulted.

Although we cannot always expect to satisfy the cost threshold, tuning a single parameter reduces the development work required and would allow us to use standard collision avoidance performance metrics to provide a statistical analysis of the system performance.

Guaranteeing global stability is a difficult task for visual predictive control. As with the classical approach, only local asymptotic stability can be assured through appropriate choice of prediction horizon. Explicitly designing for stability and robustness may highlight more stable regions of the sphere. This may then influence the choice of reference image features and thus provide better assurance the true conical spiral is tracked. Current work includes applying such design approaches to visual predictive control in addition to delay compensation for improved control.

VII. Acknowledgements

This work was funded by Endeavour Research Fellowship Award and the Australian Research Centre for Aerospace Automation (ARCAA). We also thank the Autonomous Systems Laboratory at ETH Zurich, in particular M.Burri and R. Siegwart.

References

[1] K. Dalamagkidis, K. Valavanis, and L. Piegl, "On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the national airspace system," *Progress in Aerospace Sciences*, vol. 44, no. 7-8, pp. 503-519, 2008

[2] Australian Transport Safety Bureau (ATSB), "Limitations of the see-and-avoid principle," Australian Government, Tech. Rep, Nov. 2004.

[3] C. Geyer, C. Singh, and S. Chamberlain, "Avoiding collisions between aircraft: state of the art and requirements for UAVs operating in civilian airspace," *Technical Report: CMU-RI-TR-08-03*, March 2008.

[4] B. Karhoff, J. Limb, S. Oravsky, and A. Shephard, "Eyes in the domestic skies: an assessment of sense and avoid technology for the army's warrior unmanned aircraft," *Proc. IEEE Systems and Information Engineering Design Synopsium*, pp. 36-42, 2006

[5] L. Mejias, S. McNamara, J. Lai, and J. Ford, "Vision-based detection and tracking of aerial targets for uav collision avoidance," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, IROS'10*, pp. 87-92, Oct. 2010

[6] O. Shakernia, M. Z. Chen, and V. M. Raska, "Passive ranging for uav sense and avoid applications,"*Proc. AIAA Infotech@Aerospace Conf.*, pp. 1-10, March 2005

[7] M. Kochenderfer, J. Griffith, and J. Kuchar, "Hazard alerting using line-of-sight rate," *Proc. AIAA Guidance, Navigation and Control Conference and Exhibit*, Aug. 2008.

[8] J. Lai, L. Mejias, and J. Ford, "Airborne vision-based collision-detection system," *Journal of Field Robotics*, vol. 28, no. 2, pp. 137-157, March 2011.

[9] T. Bruggemann and L. Mejias, "Airborne collision scenario flight tests: impact of angle measurement errors on reactive vision-based avoidance control," *Proc. 15th Australian Int. Aerospace, AIAC'13*, Feb. 2013.

[10] J. Kuchar and L. C. Yang, "Survey of conflict detection and resolution modelling methods," *IEEE Trans. Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179-189, Dec. 2000.

[11] B. Fajen and W. Warren, "Behavioural dynamics of steering, obstacle avoidance, and route selection," *Journal of Experimental Psychology-Human Perception and Performance*, vol. 29, no. 2, pp. 343-361, 2003.

[12] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robotics and Automation*, vol. 12, no. 5, pp. 651-670, Oct. 1996.

[13] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, "Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle," *IEEE Trans. Robotics*, vol. 25, no. 3, pp. 743-749, June 2009.

[14] A. Beyeler, J. Zufferey, and D. Floreano, "Vision-based control of near-obstacle flight," *Autonomous Robots*, vol. 27, pp. 201-219, July 2009.

[15] K. Boyadzhiev, "Spirals and conchospirals in flight of insects," *The College Mathematics Journal*, vol. 30, pp. 23-31, Jan. 1999.

[16] R. Sharma, J. Saunders, and R. Beard, "Reactive path planning for micro air vehicles using bearing only measurements," *Journal of Intelligent Robotic Systems*, vol. 65, pp. 409-416, Jan. 2012.

[17] L. Mejias, I. Mondragon, and P. Campoy, "Omnidirectional bearing-only see and avoid for small aerial robots," *Proc. IEEE Int. Conf. Automation, Robotics and Applications. ICARA'11*, pp. 23-28, Dec. 2011

[18] X. Yang, L. Mejias, and T. Bruggemann, "A 3D collision avoidance strategy for uavs in a non-cooperative environment," *Journal of Intelligent Robotic Systems*, pp. 1-13, Aug. 2012.

[19] P. Corke, "Spherical image-based visual servo and structure estimation," *Proc. IEEE Int. Conf. Robotics and Automation, ICRA'10*, pp. 5550-5555, Sep. 2010

[20] A. Mcfadyen and L. Mejias, "Visual servoing approach to collision avoidance for aircraft," *Proc. 28th Int. Congress of the Aeronautical Sciences, ICAS'12*, Sep. 2012.

[21] A. Mcfadyen, P. Corke and L. Mejias, "Rotorcraft collision avoidance using spherical image-based visual servoing and single point features," *Proc. 28th IEEE/RSJ. Int. Conf. Intelligent Robots and Systems, IROS'12*, pp. 1199-1205, Oct. 2012.

[22] A. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Trans. Robotics*, vol. 26, no. 5, pp. 933-939, Oct. 2010

[23] D. Mayne, J. Rawlings, C. Rao and M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, No. 6, pp. 789-814, June 2000.

[24] D. Lee, H. Lim, and H. Kim "Obstacle avoidance using image-based visual servoing integrated with nonlinear model predictive control," *Proc. IEEE Int. Conf. Decision and Control,CDC'11*, pp. 5689-5694, Dec. 2011.

[25] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications," *Proc. 6th European Conf. Computer Vision, ECCV'08*, pp. 445-461, June 2000

[26] B. Houska, H. Ferreau, and M. Diehl "ACADO Toolkit - An open source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods.*, vol. 32, no. 3, pp. 298-312, 2011

[27] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000