



Queensland University of Technology
Brisbane Australia

This is the author's version of a work that was submitted/accepted for publication in the following source:

Mcfadyen, Aaron, Corke, Peter, & Mejias, Luis
(2014)
Visual predictive control of spiral motion.
IEEE Transactions on Robotics, 30(6), pp. 1441-1454.

This file was downloaded from: <http://eprints.qut.edu.au/78794/>

© Copyright 2014 IEEE

Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

Notice: *Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source:*

<http://dx.doi.org/10.1109/TRO.2014.2361425>

Visual Predictive Control of Spiral Motion

Aaron Mcfadyen, *Student Member, IEEE*, Peter Corke, *Fellow, IEEE*, and Luis Mejias, *Member, IEEE*

Abstract—This paper deals with constrained image-based visual servoing of circular and conical spiral motion about an unknown object approximating a single image point feature. Effective visual control of such trajectories has many applications for small unmanned aerial vehicles, including surveillance and inspection, forced landing (homing), and collision avoidance. A spherical camera model is used to derive a novel visual-predictive controller (VPC) using stability-based design methods for general nonlinear model-predictive control. In particular, a quasi-infinite horizon visual-predictive control scheme is derived. A terminal region, which is used as a constraint in the controller structure, can be used to guide appropriate reference image features for spiral tracking with respect to nominal stability and feasibility. Robustness properties are also discussed with respect to parameter uncertainty and additive noise. A comparison with competing visual-predictive control schemes is made, and some experimental results using a small quad rotor platform are given.

Index Terms—Nonlinear model-predictive control (NMPC), spirals, stability, visual servoing.

I. INTRODUCTION

RECENT advances in technology have seen small autonomous unmanned aerial systems (UAS) that have emerged in the commercial sector, offering a low-cost solution to an increasingly diverse set of applications. However, their full potential will not be realized until regulatory bodies¹ grant them regular and unrestricted access to the national airspace. This requires the development of a number of key-enabling technologies to increase onboard capability and ensure compliance with strict safety standards [1]. Of particular importance is a collision avoidance solution aimed at replicating pilot performance in conventionally piloted aircraft. Commonly referred to as Detect and Avoid, the pilot must independently identify the potential threat using visual cues and without active communication to the intruder aircraft [2]. A reactive control action must then be taken to ensure timely avoidance of both static and dynamic objects.

Coupling regulatory requirements with inherent size, weight, and power limitations of small UAS, cameras are a natural sensor choice for detection and tracking. Assuming availability of reliable range estimates, a number of conflict resolution ap-

Manuscript received October 3, 2013; revised May 9, 2014; accepted September 28, 2014. This paper was recommended for publication by Associate Editor T. Hamel and Editor G. Oriolo upon evaluation of the reviewer's comments. This work was supported by the Department of Education, Employment, and Workplace Relations under the Endeavour Awards program and the Australian Research Centre for Aerospace Automation.

A. Mcfadyen is with the Science and Engineering Faculty, Queensland University of Technology, Brisbane, Qld. 4007, Australia (e-mail: aaron.mcfadyen@qut.edu.au).

P. Corke and L. Mejias are with the Science and Engineering Faculty, Queensland University of Technology, Brisbane, Qld. 4007, Australia, and also with the Australian Centre for Robotic Vision, Australia (e-mail: peter.corke@ieee.org; luis.mejias@qut.edu.au).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2014.2361425

¹FAA (USA), CASA (Australia), EuroControl (Europe).

proaches could be considered. However, this is considerably challenging in a realistic operating environment. Intruder aircraft appear as pixel-sized low-contrast objects throughout the conflict. Objects can be of unknown size and lack discernible shape and growth in the image. As such, approaches based on visual looming will fail, and stereo vision becomes infeasible as the ratio of camera baseline to object range will be small. Passive ranging techniques require predefined motion to estimate range, which has the potential to degrade the collision geometry and decrease available avoidance time [3]. Additionally, timely convergence is not guaranteed [4]. Reliably identifying a collision threat using relative angular rate is difficult, but relative angular position can be estimated with greater consistency [5]. New approaches using these measurements directly, as opposed to range, are thus required.

Conical spirals offer a potential solution.² Frequently occurring in nature and broadly studied by mathematicians, artists and scientists, they describe the set of trajectories that circumscribe the surface of a cone [6]. By holding the conical apex at a fixed angle, a vehicle may track away from, toward, or about the apex according to a set of logarithmic equations. The idea can be reduced to two dimensions, resulting in planar spirals. An intruder aircraft in the Detect and Avoid environment could then be considered the conical apex and tracking an outward divergent spiral may ensure collision avoidance. Similar arguments can be made for other vision-based navigation tasks in which the object to be observed, inspected, or tracked approximates a single point in the image.

By approximating the object as the conical apex and regulating its relative angular position directly from visual observations, an image-based visual servoing (IBVS) approach can, therefore, provide a framework to guide conical spiral motion. It provides a reactive control approach robust to range error and camera calibration [7]. For complete visual control, at least four points from the same object are required, yet only a single point feature would be adequate for partial control in the vertical and lateral planes. Assuming object detection [9], [10], the difficulties lie in satisfying problem constraints and ensuring stability. Vehicle dynamics, actuator limitations, and sensor field of view should be explicitly considered in the design. Stable control is required to ensure the reference spiral is tracked, and safe operation is maintained.

In this paper, we focus on constrained image-based control of conical spiral motion, of which circular motion is a special case. Using stability-based design strategies for general nonlinear model-predictive control (NMPC), we derive a novel spherical visual-predictive control (VPC) approach that is stable and computationally efficient and provides good spiral tracking performance.

²They have been called logarithmic, equiangular, and conchospirals, with each name describing a particular characteristic of the resulting curve.

II. BACKGROUND

If a spiral is to be tracked in the task space, we would first require knowledge of the apex position: the position-based visual servoing approach. The control task then becomes relatively trivial, and as described previously, the difficulty lies in reliably estimating the apex range and subsequently approximating its position. Alternatively, a conical spiral may be tracked using image feedback directly: the IBVS approach.

IBVS involves regulating a set of image features $\mathbf{s}(t)$ toward a set of reference values $\mathbf{s}^*(t)$ using some or all camera degrees of freedom, defined by a control vector $\mathbf{u}(t)$. In order to do so, we require the relationship between image feature velocity $\dot{\mathbf{s}}(t)$ and camera velocity $\dot{\mathbf{x}}(t)$ defined by the image Jacobian or interaction matrix L_s . In the classical approach, we require an exponential decrease of image feature error $\mathbf{e}(t)$, resulting in the control defined by

$$\mathbf{u}(t) = -\lambda \hat{L}_s^+ \mathbf{e}(t) \quad (1)$$

where λ is a constant positive gain term, and \hat{L}_s^+ is the Moore–Penrose pseudo inverse of the image Jacobian [7]. Controllers for 2-D [11], [12] and 3-D [13], [14] spiral tracking have been derived using this principle. Similar methods that do not explicitly consider the image Jacobian have also been proposed [15]–[17], and control is based on relative bearing and elevation. Tuning a gain term is required to simultaneously satisfy problem constraints and ensure sufficiently fast convergence. Stability can then be guaranteed in this single case. The gain term is often empirically derived and can lead to suboptimal behavior.

Optimal control-based approaches have emerged to help address these issues. Control is derived by minimizing an objective function J over an infinite time horizon such that

$$\mathbf{u}(\cdot) = \underset{\bar{\mathbf{u}}(\cdot)}{\operatorname{argmin}} J(\dots, \dots) \quad (2)$$

where (\cdot) denotes a control sequence defined over the corresponding horizon. The objective function, however, is quite general and could contain not only image feature error, but a combination of image features, control effort, time, and other application-specific criteria. This allows flexibility in the control specifications while explicitly considering problem constraints. Included in the set of optimal control-based approaches are path planning [18], [19] and linear matrix inequalities (LMI) [20], [21] formulations of IBVS using linear differential inclusions or otherwise. Although global in nature, they require a unique solution to exist, which is not always guaranteed. They are often designed with robustness in mind and consider worst-case scenarios, which results in overly conservative control schemes with larger computational overhead. This is particularly relevant when the online computation of a minimax problem is required. Such approaches have not been applied to spiral tracking, but are likely to be unsuitable considering the requirement for multiple image features from the same object.

Nonlinear model-predictive or receding horizon control (NMPC) describes a particular class of optimal control-based approaches, solved over a finite time or prediction horizon T_p instead [22]. As opposed to a global solution, NMPC allows the control law to be recalculated periodically, incorporating a feedback mechanism into the system to help compensate for external

disturbances. Generally, accurate process models are required for effective control. As good approximations for both image and robot kinematic models can be derived *a priori*, the NMPC framework is well suited to image-based visual control. The term VPC is commonly used, and the optimal control problem to be solved becomes

$$\mathbf{u}(\cdot) = \underset{\bar{\mathbf{u}}(\cdot)}{\operatorname{argmin}} J(\mathbf{e}(t), \dots) |_t^{t+T_p}. \quad (3)$$

Solving a finite horizon as opposed to an infinite horizon optimal control problem reduces computational complexity. Significantly complex or large systems can be solved provided the horizon time is reasonable. In the VPC, the state vector may consist only of image features or a combination of vehicle states and image features. The nonlinear optimization problem is then solved over the resulting state space such that control, state, and visibility constraints are managed accordingly. Importantly, object pose is not reconstructed, and differences in various schemes often revolve around image feature representation and objective function structure [23]–[25]. The system dynamics are often linearized, and the objective function consists of a weighted sum of quadratic terms in state and control. Such an objective function will hereafter be referred to as the general form. In the special case, the control resulting from (3) is equal to that of (1) as \hat{L}_s^+ approximates a least-square solution to the inverse image kinematics.

Ensuring closed-loop stability for NMPC is difficult, but can be achieved by assuming the general form of the objective function and selecting suitable design parameters. These include the prediction horizon T_p , control horizon T_c , and objective function weighting matrices [26]. The approach is typically used for real applications, and although it has proved to be effective for VPC [27]–[29], defining the parameters often requires experience. They are chosen to provide a tradeoff in computational complexity and stability. Typically, the prediction and control horizon are equal, and tuning is predominantly on the prediction horizon length. If too short, stability issues may arise. If too long, the computational expense increases significantly, and the control horizon can be used to reduce complexity. To this end, move blocking approaches are commonly used. By considering a single constant control over multiple steps within the prediction horizon, the complexity of the optimization problem is considerably reduced. This has been studied in [30] and applied to VPC in [31]. Unfortunately, stability can no longer be guaranteed, constraint satisfaction may be difficult, and the resulting constant control may be suboptimal or infeasible. Tuning guidelines exist [32], but due to the variability in system dynamics, no formal rules have been established for general NMPC or VPC. For visual control, the nonlinearity of the image dynamics results in considerable differences in system behavior for each operating point, further complicating parameter selection. A loss of performance may then result as a compromise for ensured stability.

Alternatively, design methods exist that guarantee nominal closed-loop stability, without explicitly tuning system parameters. By altering the control structure, they rely only on the existence of a feasible solution at the initial time. Feasibility at each subsequent time is then guaranteed resulting in stable behavior. In the first approach, a terminal equality constraint can

be added [33], [34] such that (3) becomes

$$\mathbf{u}(\cdot) = \underset{\bar{\mathbf{u}}(\cdot)}{\operatorname{argmin}} J(\mathbf{e}(t), \dots) \mid_t^{t+T_p} \text{ s.t. } \mathbf{e}(t+T_p) = 0. \quad (4)$$

The reference state must be reached within a finite prediction horizon, which often leads to a restricted region of operation. Therefore, feasibility and thus, stability, becomes an issue for small prediction horizons. To ensure a satisfactory region of operation, a large prediction horizon is required which brings increased computational burden. Additionally, exact numerical satisfaction of the terminal equality requires an infinite number of calculations. Approximations must be made for tractability, which can compromise stability results.

The second approach is to apply a terminal penalty term and terminal region to the control structure, forcing the final state to lie in a bounded region about the reference value at the end of the horizon [35], [36]. Within this region, convergence is then guaranteed. Transforming (3) into this framework

$$\mathbf{u}(\cdot) = \underset{\bar{\mathbf{u}}(\cdot)}{\operatorname{argmin}} J(\mathbf{e}(t), \dots, \mathbf{e}(T_p)) \mid_t^{t+T_p} \text{ s.t. } \mathbf{e}(t+T_p) \in \Omega_s \quad (5)$$

where $\mathbf{e}(T_p)$ denotes a terminal penalty term, and Ω_s defines the terminal region. By including the terminal region constraint, the approach is less restrictive than the first. The requirement to satisfy a zero terminal constraint in finite time is removed, decreasing the potential for feasibility issues when using small prediction horizons. The existence, shape, and magnitude of the terminal region can also be used to discriminate between particularly nonlinear regions of the state space and help estimate the domain of attraction (region of operation). A large terminal region indicates the system dynamics are relatively linear about the associated reference state, resulting in a greater region of convergence. The reverse is true for small terminal regions. The terminal penalty term is used to virtually extend the horizon to infinity by approximating an upper bound on the objective function. As a result, the approach is commonly referred to as quasi-infinite horizon nonlinear model-predictive control (QIH-NMPC).

The terminal penalty term and terminal region depend on multiple system-specific factors and cannot be chosen freely. They are in general difficult to determine, but can be approximated offline. This improves the design process by allowing analysis of some system attributes before controller implementation. Additionally, by moving the majority of the computational burden offline, the online optimization can be solved sufficiently fast using modern solvers [37], [38]. Such solvers are capable of managing infeasible initial states, searching for alternate solutions when required.

Designing VPC using the quasi-infinite framework has not yet been explored. For tasks such as spiral tracking, coupling a spherical camera and the quasi-infinite horizon control design offers some significant benefits over traditional approaches. These include the following:

- 1) guaranteed nominal closed-loop stability;
- 2) a means to estimate the domain of attraction for any reference image features. This provides valuable information on the reachability of the reference state given the initial

conditions. For general image-based visual servoing, this is difficult or impossible;

- 3) a means to design suitable reference image features for region-reaching IBVS. We may only require a divergent spiral or one that exists entirely above or below the apex. Therefore, the required reference image features are not unique, but instead exist in a particular region of the image. Recent region-reaching controllers [39] will fail when using a single point feature, due to singularities at the region boundary; therefore, selection of specific image features may still be required. Suitable reference image features may then be selected *a priori* based on the attributes of the associated terminal region such that feasibility issues are less likely;
- 4) a large field of view offered by spherical imaging that ensures the object remains visible for an arbitrary spiral, regardless of the control input or external disturbances;
- 5) an efficient implementation as the computational complexity is managed offline, allowing adequate control without the need for move blocking. This is important for visual control, given its sensitivity to system latency through image-processing delays and the desire for optimal performance [40].

In this study, a number of significant contributions are presented including the following:

- 1) first application of VPC using the stability-based quasi-infinite NMPC design;
- 2) novel IBVS approach for spiral tracking, including circular motion as a special case;
- 3) comparable spiral-tracking performance with respect to recently proposed predictive control schemes, and a qualitative feasibility and robustness analysis regarding parameter uncertainty and additive noise;
- 4) extension of IBVS for aerial vehicles using spherical imaging, VPC, and single point features.

This paper is structured as follows. In Section III we provide a formal problem description and introduce spherical cameras, conical spirals, and VPC. In Section IV quasi-infinite horizon visual-predictive control (QIH-VPC) is derived. Controllers for circular and spiral motion are presented and analyzed in Sections V and VI, respectively. An analysis regarding the choice of reference image features is presented before feasibility and robustness issues are analyzed. A performance comparison with respect to recently proposed VPC schemes is also included. Experimental results are presented in Section VI, followed by concluding remarks in Section VIII.

III. PROBLEM DESCRIPTION

The following notation is used in this paper. For any scalar $x \in \mathbb{R}^1$, $|x|$ denotes its absolute value. For any vector $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|$ denotes the two norm, and \mathbf{x}_P denotes the weighted norm such that $\mathbf{x}_P^2 = \mathbf{x}^T P \mathbf{x}$, where $P \succ 0$ is a positive-definite matrix. The maximum and minimum real eigenvalues of a matrix P are given by $\lambda_{\max}(P)$ and $\lambda_{\min}(P)$, respectively. A vector-valued function is given by $\mathbf{f}(\cdot)$, and a reference value is denoted with an asterisk. For example, \mathbf{u}^* defines a reference control vector.

A. Conical Spirals

Consider a fixed world reference frame with its origin positioned at the apex of a stationary cone and its z -axis along the cone axis. The equations describing motion along a conical spiral path about the apex on the surface of the cone in Cartesian coordinates are given by

$${}^w x(t) = (r_0 - bt) \cos \left(\eta_0 - c \log \left(1 + \frac{at}{z_0} \right) \right) \quad (6)$$

$${}^w y(t) = (r_0 - bt) \sin \left(\eta_0 - c \log \left(1 + \frac{at}{z_0} \right) \right) \quad (7)$$

$${}^w z(t) = z_0 + at \quad (8)$$

where $a = v \cos(\alpha) \cos(\beta)$, $b = v \cos(\alpha) \sin(\beta)$, and $c = \tan(\alpha) / \sin(\beta)$ [6]. They are defined in terms of a constant speed v and conical angles $\mathbf{c} = [\alpha \beta]$. The elevation angle β defines the angle between the apex and the current position measured from the positive z -axis such that $0 < \beta < \pi$. The bearing angle α is defined as the angle between the projection of the velocity vector onto the xy plane and the apex such that $-\pi < \alpha < \pi$. The speed is defined as the magnitude of the translational velocity vector $\dot{\mathbf{x}}_t$ such that $v = \|\dot{\mathbf{x}}_t\| = (v_x^2 + v_y^2 + v_z^2)^{1/2}$. The initial range and vertical displacement from the apex are defined by r_0 and z_0 , respectively. The initial angular position from the apex measured from the positive x -axis is defined by η_0 . The variables are depicted in Fig. 1. Fixing the conical angles to a reference value such that $\mathbf{c}^* = [\beta^* \alpha^*]$ and solving (6)–(8) as $t \rightarrow \infty$, we obtain

$$\lim_{t \rightarrow \infty} {}^w x(t) = \begin{cases} 0, & 0 < |\alpha^*| < \frac{\pi}{2} \\ (l^2 - y^2)^{1/2}, & \frac{\pi}{2} = |\alpha^*| \\ \infty, & \frac{\pi}{2} < |\alpha^*| < \pi \end{cases} \quad (9)$$

$$\lim_{t \rightarrow \infty} {}^w y(t) = \begin{cases} 0, & 0 < |\alpha^*| < \frac{\pi}{2} \\ (l^2 - x^2)^{1/2}, & \frac{\pi}{2} = |\alpha^*| \\ \infty, & \frac{\pi}{2} < |\alpha^*| < \pi \end{cases} \quad (10)$$

$$\lim_{t \rightarrow \infty} {}^w z(t) = \begin{cases} \infty, & 0 \leq \beta^* < \frac{\pi}{2} \cap |\alpha^*| > \frac{\pi}{2} \\ 0, & \frac{\pi}{2} = \beta^* \cup |\alpha^*| < \frac{\pi}{2} \\ -\infty, & \frac{\pi}{2} < \beta^* \leq \pi \cap |\alpha^*| > \frac{\pi}{2} \\ h, & \frac{\pi}{2} = |\alpha^*| \forall \beta \end{cases} \quad (11)$$

where h and l are constants primarily determined by r_0 and β^* . Circular motion about the apex is a special case of conical motion when $\alpha^* = \pm\pi/2$ such that $x^2 + y^2 = l^2$. Divergent spiral motion whereby the spiral radius increases is denoted by ∞ , as strictly speaking, the limit does not exist. Considering the complete domain of the conical angles, degenerate cases exist for the median values. Specifically, for $\alpha^* = 0$, nonspiral motion toward the apex results. For $\beta = \pi/2$, motion is no longer on a conical spiral, but a planar spiral with $z = 0$. In Fig. 3, some example spirals and their corresponding reference conical angles are shown.

Now consider a spiral reference frame, attached to the vehicle moving along the conical spiral trajectory. The origin of the

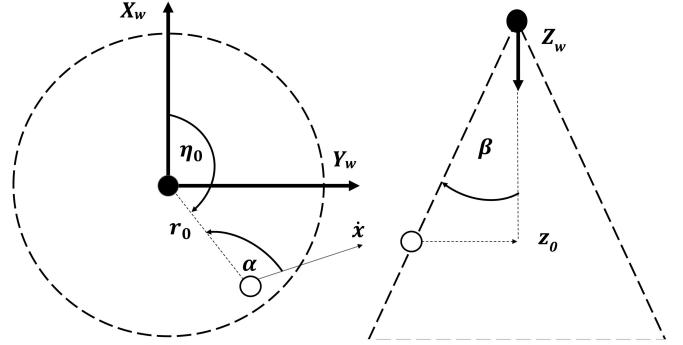


Fig. 1. Top view and side view of a reference cone (dashed), its apex (solid black dot), and parameters defining conical spiral motion.

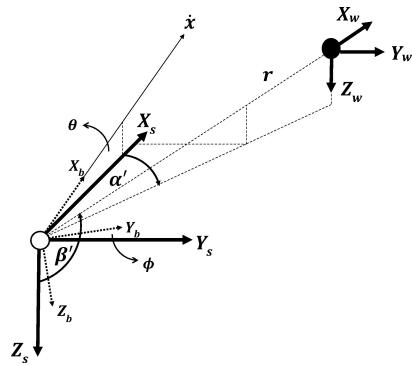


Fig. 2. World (w), spiral (s), and body (b) coordinate frames and elevation (β') and azimuth (α') angles in spiral frame.

spiral reference frame and body frame is initially aligned. The spiral frame inherits the body translational motion but only its angular velocity in the z -axis. The orientation of the world frame with respect to the spiral frame can be defined as a rotation ${}^s R_w$ about the vertical axis of the world frame such that

$${}^s R_w = R(\psi) \quad (12)$$

where ψ denotes the vehicle yaw angle. By the definition above and considering Fig. 2, the relationship between the bearing and elevation angle in the spiral and world frame is given by

$$\alpha'(t) = \alpha(t), \quad \beta'(t) = \pi - \beta(t) \quad (13)$$

where a dash denotes an angle measured in the spiral frame. Combining (6)–(8) with (13), a conical spiral can be tracked by regulating β' and α' . The introduction of the spiral reference frame allows the derivation of a convenient relationship between the image features and conical angles, which is presented in the following section.

B. Spherical Cameras

Ideal spherical cameras offer a 4π steradian field of view useful for many visual navigation applications. A unified imaging model can be used to map image features represented by a number of camera types, including wide angle and

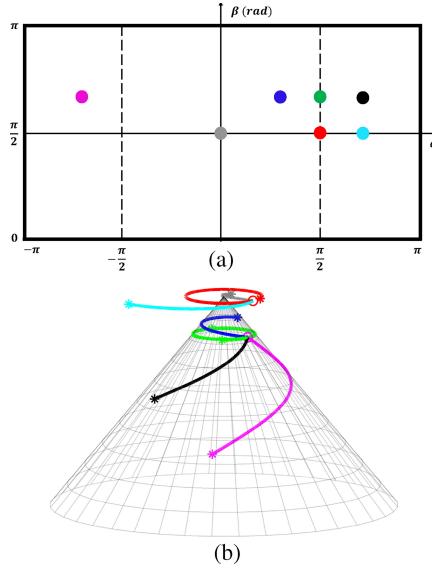


Fig. 3. (a) Example reference conical angles depicted in α - β space. (b) Corresponding conical spiral trajectories. Notice the degenerate cases (light blue, red, gray), circular motion (red, green), divergent spirals (magenta, black), and convergent spiral (dark blue). An “o” and “*” mark the trajectory start and finish positions, respectively.

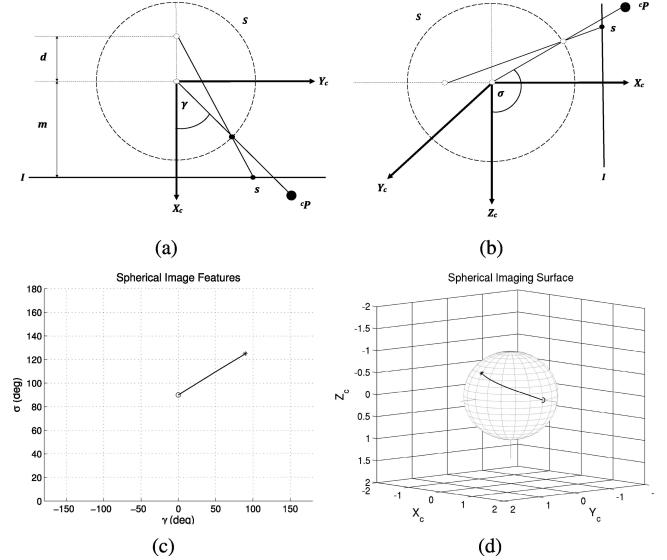


Fig. 4. Simplified spherical projection model and example image feature trajectory. (a) Azimuth angle. (b) Colatitude angle. (c) Image feature trajectory in Cartesian representation. (d) Image feature trajectory on spherical imaging surface.

perspective, to an ideal imaging sphere [41]. A point feature is first mapped to the surface of a unit sphere S and then reprojected onto an imaging plane I . The mapping depends on the camera parameters (d, m) and mirror type [42]. The resulting image feature s can be represented using an azimuth γ and colatitude σ angle measured from the spherical center such that $s = [\sigma \gamma]$. A simplified representation of the spherical camera geometry and example image feature motion of the sphere is depicted in Fig. 4. Note that tracking a great circle over the sphere appears as a straight line on a Cartesian representation of the σ - γ space.

Consider a spherical camera rigidly attached to an aerial vehicle observing a point object, considered to be the conical apex. In the camera frame, the apex is denoted by ${}^c\mathbf{p}$ such that

$${}^c\mathbf{p}(t) = \begin{pmatrix} X(t) \\ Y(t) \\ Z(t) \end{pmatrix} = \begin{pmatrix} r(t)S\sigma(t)C\gamma(t) \\ r(t)S\sigma(t)S\gamma(t) \\ r(t)C\sigma(t) \end{pmatrix} = r(t){}^c\tilde{\mathbf{p}}(t) \quad (14)$$

where $C = \cos(\cdot)$, $S = \sin(\cdot)$, and $r(t)$ is the distance to the object, and a tilde denotes a unit vector. Recalling the origin of the body and spiral frames to be equal, then in the spiral reference frame, (14) becomes

$${}^s\mathbf{p}(t) = {}^sR_b(t){}^bR_c{}^c\mathbf{p}(t) + {}^s\mathbf{t}_c \quad (15)$$

$$r(t){}^s\tilde{\mathbf{p}}(t) = r(t){}^sR_b(t){}^bR_c{}^c\tilde{\mathbf{p}}(t) + {}^s\mathbf{t}_c \quad (16)$$

where ${}^sR_b(t)$ and bR_c define rotation matrices from body to spiral and camera to body frames, respectively. The camera focal point and origin of the body frame are separated by the vector ${}^s\mathbf{t}_c$, which has the same magnitude as ${}^s\mathbf{t}_c$. Dividing through by

$r(t)$ and assuming $r(t) \gg \|{}^s\mathbf{t}_c\|$

$${}^s\tilde{\mathbf{p}}(t) = {}^sR_c(t){}^b\tilde{\mathbf{p}}(t) \quad (17)$$

where ${}^sR_c(t) = {}^sR_b(t){}^bR_c$. Substituting for \mathbf{p} using spherical coordinates

$$\begin{pmatrix} S\beta'(t)C\alpha(t) \\ S\beta'(t)S\alpha(t) \\ C\beta'(t) \end{pmatrix} = {}^sR_c \begin{pmatrix} S\sigma(t)C\gamma(t) \\ S\sigma(t)S\gamma(t) \\ C\sigma(t) \end{pmatrix} \quad (18)$$

where

$${}^sR_c(t) = R(\theta) R(\phi) \quad (19)$$

$${}^sR_c(t) = \begin{pmatrix} C\theta(t) & S\theta(t)S\phi(t) & C\phi(t)S\theta(t) \\ 0 & C\phi(t) & -S\phi(t) \\ -S\theta(t) & C\theta(t)S\phi(t) & C\theta(t)C\phi(t) \end{pmatrix} \quad (20)$$

and $\theta(t)$ and $\phi(t)$ denote camera pitch and roll angles, respectively. The conical angles can then be derived by solving (18) using the measured spherical image features and vehicle orientation.

Depending on the vehicle dynamics, the reference spiral will typically be followed with nonzero pitch and roll. This is certainly the case for fixed wing aircraft, in which coordinated turns require nonzero yaw and roll angle, and climbing and descending require nonzero pitch. In this case, the reference image features and reference conical angles would not be equal. However, by using derotated images, it is reasonable to assume $\theta(t) \approx \phi(t) \approx 0 \forall t$; therefore, ${}^sR_c = I_3$ and $s^* \approx c^*$. This way, the problem is generalized for any aircraft type. Reference image features that directly correspond to a particular conical spiral could first be selected. The spiral can then be tracked by directly regulating these image features assuming the image kinematics are known.

The image kinematics define the relationship between image feature velocity $\dot{\mathbf{s}}(t)$ and camera translational and angular velocity components $\dot{\mathbf{x}}(t)$ such that

$$\dot{\mathbf{s}}(t) = L_s(\mathbf{s}(t), r(t)) \dot{\mathbf{x}}(t) \quad (21)$$

where L_s is the spherical image Jacobian [43], and $\dot{\mathbf{x}}(t) \in \mathbb{R}^6$. Most aircraft are underactuated; therefore, we can only control some of these velocities. We can then express $\dot{\mathbf{x}}(t)$ in terms of the controlled velocities $\mathbf{u}(t)$ and the remaining velocities $\mathbf{v}(t)$ such that $\dot{\mathbf{x}}(t) = [\mathbf{u}(t) \ \mathbf{v}(t)]$, where $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{v} \in \mathbb{R}^n$, and $m + n = 6$. Considering the general case for aerial vehicles, we can track a spiral trajectory with a fixed forward velocity v_x and variable vertical velocity v_z and yaw rate³ ω_z . We cannot control all three velocities using visual feedback as only a single image feature is being observed. Setting $\mathbf{u} = [v_z \ \omega_z]$ and $\mathbf{v} = [v_x \ v_y \ \omega_x \ \omega_y]$, the image kinematics can be rewritten as

$$\dot{\mathbf{s}}(t) = L^u(\mathbf{s}(t), r(t))\mathbf{u}(t) + L^v(\mathbf{s}(t), r(t))\mathbf{v}(t) \quad (22)$$

where

$$L^u(\cdot) = \begin{pmatrix} \frac{S\sigma(t)}{r(t)} & 0 \\ 0 & -1 \end{pmatrix} \quad (23)$$

$$L^v(\cdot) = \begin{pmatrix} \frac{-C\sigma(t)C\gamma(t)}{r(t)} & \frac{-C\sigma(t)S\gamma(t)}{r(t)} & S\gamma(t) & -C\gamma(t) \\ \frac{S\gamma(t)}{r(t)S\sigma(t)} & \frac{-C\gamma(t)}{r(t)S\sigma(t)} & \frac{C\gamma(t)C\sigma(t)}{S\sigma(t)} & \frac{S\gamma(t)C\sigma(t)}{S\sigma(t)} \end{pmatrix} \quad (24)$$

and L_u and L_v represent partitions of the spherical image Jacobian such that $L_s = [L^u(\cdot) \ L^v(\cdot)]$. Partitioning this way ensures decoupled control in the vertical ($z - x$) and lateral ($x - y$) planes in the camera frame. Forward velocity can then be regulated by an external controller. Assuming a fixed reference range value,⁴ as is usually done in visual servoing, (22) reduces to

$$\dot{\mathbf{s}}(t) = \hat{L}^u(\mathbf{s}(t))\mathbf{u}(t) + \hat{L}^v(\mathbf{s}(t))\mathbf{v}(t) \quad (25)$$

where \hat{L}^u and \hat{L}^v denote approximations of the partitioned image Jacobian taken for $r(t) = r^*$. Assuming the forward velocity v_x remains relatively constant and the image features are taken from derotated images, $\mathbf{v}(t)$ can be considered constant as $v_y(t) \approx \omega_x(t) \approx \omega_y(t) \approx 0$. Rewriting (25) in standard ODE form, the local process model can be defined by

$$\dot{\mathbf{s}}(t) = \mathbf{f}(\mathbf{s}(t), \mathbf{u}(t)) \quad (26)$$

$$\mathbf{f}(\mathbf{s}(t), \mathbf{u}(t)) = \hat{L}^u(\mathbf{s}(t))\mathbf{u}(t) + \hat{L}^v(\mathbf{s}(t))\mathbf{v} \quad (27)$$

where \mathbf{v} may be passed as a predetermined parameter.

C. General Control Problem

The general control problem can now be defined.

³These may be used directly or transformed into angular displacement commands, which is commonly done for fixed wing aircraft.

⁴This is only true for motion on a circular plane, and model mismatch results otherwise.

- 1) Set the desired conical angles, $\mathbf{c}^* = [\beta^* \ \alpha^*]$ according to a particular application. For collision avoidance, a circular or divergent spiral is preferred; therefore, $\beta^* \neq \pi/2$, and $\alpha^* \geq \pi/2$. As the image features are derotated using (17)–(20), the reference image features and conical angles are equal such that $\mathbf{s}^* = \mathbf{c}^*$.

- 2) Measure the image features $\mathbf{s}(t) = [\sigma(t) \ \gamma(t)]$, and find the feature error $\mathbf{e}(t)$ according to

$$\mathbf{e}(t) = \mathbf{s}(t) \ominus \mathbf{s}^* \quad (28)$$

where \ominus denotes a modulo 2π subtraction such that $\mathbf{e}(t)$ is the smallest angular difference on the sphere. Any co-latitude and azimuth angle outside the bounded region defined by the spherical surface can be mapped back to a point within the bounded set in a similar manner.

- 3) Find the control $\mathbf{u}(t)$ such that the

$$\lim_{t \rightarrow \infty} \mathbf{e}(t) = 0 \quad (29)$$

s.t.

$$\dot{\mathbf{s}}(t) = \mathbf{f}(\mathbf{s}(t), \mathbf{u}(t)) \quad (30)$$

$$\mathbf{u}(t) \in \mathbb{U} \subset \mathbb{R}^2 \quad \forall t \geq 0 \quad (31)$$

$$\mathbf{s}(t) \in \mathcal{S}^2 \quad \forall t \geq 0 \quad (32)$$

where \mathbb{U} defines the control constraint domain, and \mathcal{S}^2 defines the set of points on a two-sphere. It can be thought of as 2-D manifold in Euclidean space representing the spherical imaging surface.

D. Classical Visual-Predictive Control

VPC is based on well-established NMPC [22]. It requires a reference state vector, a process model, an objective function, and a solver for the resulting optimization problem. At each sampling instance, a control sequence of camera velocities is found that minimizes the objective function under a set of nonlinear visibility, state, and control constraints. The objective function is calculated over a finite prediction horizon T_p using a process model that may include a combination of image kinematics and vehicle dynamics. The first of the control sequence is applied before the process is repeated each time a new measurement arrives. Essentially, a constrained finite horizon nonlinear optimal control problem is solved at each sampling instance.

Typically, the objective function takes the general form; however, it is possible to manipulate the objective function structure online. Terms may be added, removed, or altered, reflecting their relative importance to the intended application. Any objective function weighting matrices may be constant or time varying over the prediction horizon or successive optimizations. Recently, a progressively incremented positive-definite state penalty matrix $Q(\tau)$ is used to penalize states more heavily toward the end of the horizon [31]. No penalty on control was used and is essentially handled by the control constraints. The control problem can be defined in continuous time as

$$\mathbf{u}(\cdot) = \underset{\bar{\mathbf{u}}(\cdot)}{\operatorname{argmin}} J(\mathbf{e}(t)) \quad (33)$$

$$J(\mathbf{e}(t)) = \int_t^{t+T_p} \mathbf{e}(\tau)^2 Q(\tau) d\tau \quad (34)$$

s.t

$$\dot{\bar{s}} = \mathbf{f}(\bar{s}, \bar{\mathbf{u}}), \quad \bar{s}(t) = s_0 \quad (35a)$$

$$Q(\tau) = 2Q(\tau - T) \quad (35b)$$

$$\bar{\mathbf{u}}(\tau) = \rho, \quad \tau \in [t, t + T_p] \quad (35c)$$

$$\rho \in \mathbb{U} \quad (35d)$$

where T defines the sampling time. Internal variables used in prediction are distinguished using a bar as they will not in general be equal to their actual future values. The trajectory of the predicted image features $\bar{s}(\cdot)$ results from adopting the control sequence $\bar{\mathbf{u}}(\cdot)$, starting from the true initial image features s_0 . Move blocking is used as a single constant control ρ is applied over the entire prediction horizon. Although computational efficient, move blocking results in potentially suboptimal solutions, having not considered the full richness of control actions over the prediction horizon.

Although local stability is ensured, it is difficult to infer the domain of attraction for a given reference or consider any global stability properties. Large prediction horizons increase the computational complexity, while small horizons will inherit similar stability properties as classical image-based approaches. This VPC scheme shall be referred to as classical VPC (C-VPC) and will be used for comparative purposes.

IV. QUASI-INFINITE HORIZON VISUAL-PREDICTIVE CONTROL

Transforming the VPC problem into the quasi-infinite horizon nonlinear control framework requires the offline calculation of some additional parameters. First, the image kinematics are shifted and linearized about the reference point. Second, a stabilizing locally linear controller is used to derive a terminal penalty matrix P to augment the objective function. Third, a terminal region Ω_ζ about the operating point is calculated and added as an extra nonlinear constraint. The online optimization problem defined in (33)–(35) can then be reformulated with the inclusion of these additional controller parameters.

A. Problem Setup

Considering the application, the reference image features s^* , range r^* , and forward velocity v_x^* are first selected. The control constraint domain $\mathbb{U} \subset \mathbb{R}^2$ must then be defined and is typically a function of the vehicle limitations such as turn or climb rate. Assuming a constant forward velocity, the reference controls \mathbf{u}^* can then be calculated using circular motion equations or otherwise. If $\mathbf{u}^* \notin \mathbb{U}$, then the reference spiral is infeasible, and new reference image features, range or forward velocity must be selected until the control constraints are satisfied. This is not a drawback of the controller but results from the vehicles physical constraints and chosen application.

The image kinematics representing the process model must then be shifted about the reference operating point such that $\mathbf{f}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$. Rewriting (27) for the shifted model

$$\begin{aligned} \mathbf{f}(\mathbf{s}(t), \mathbf{u}(t)) &= \hat{L}^u(\mathbf{s}(t) + \mathbf{s}^*)(\mathbf{u}(t) + \mathbf{u}^*) \\ &\quad + \hat{L}^v(\mathbf{s}(t) + \mathbf{s}^*)\mathbf{v} \end{aligned} \quad (36)$$

where the associated control constraint domain must also be shifted such that $\mathbb{U} = \{\mathbf{u}^{\min} \leq (\mathbf{u}(t) + \mathbf{u}^*) \leq \mathbf{u}^{\max}\}$. Recalling that \mathbf{v} can be approximated as a constant maintained by external controllers, this system can then be linearized about $\mathbf{s}(t) = \mathbf{0}$ and $\mathbf{u}(t) = \mathbf{0}$. Using small disturbance theory, the resulting state-space representation for the process model is

$$\dot{\mathbf{s}}(t) = A\mathbf{s}(t) + B(\mathbf{u}(t)\mathbf{v})^T \quad (37)$$

where $A = \partial\mathbf{f}(\mathbf{s}, \mathbf{u})/\partial\mathbf{s}$, and $B = \partial\mathbf{f}(\mathbf{s}, \mathbf{u})/\partial\mathbf{u}$. The reference control and image features for this system are now zero vectors as a result of the shifted process model. Equations (36) and (37) have been derived with consideration to the necessary conditions required for general QIH-NMPC control design as stated in A1–A3 in the Appendix.

B. Terminal Penalty Term

The terminal penalty matrix P is used to apply a quadratic terminal cost $\bar{s}(t + T_p)^2$ within the objective function. The term is used to penalize deviation from the reference state at the end of the prediction horizon and provide an upper bound on the objective function $\forall t > T_p$ where a fictitious locally linear state feedback controller is assumed. This linear controller is derived for the shifted process model defined by (37) and used directly to calculate P .

First, a static gain matrix K must be defined about the reference value for a linear state feedback controller such that $\mathbf{u}(t) = K\mathbf{s}(t)$. The choice of controller is somewhat arbitrary; however, by solving a linear quadratic regulator (LQR) type problem, an optimal linear controller may be found. The associated Riccati equation

$$A^T S + S A - (S B) R^{-1} (B^T S) + Q = 0 \quad (38)$$

is first solved for S , where the state and control weighting matrices are defined by Q and R , respectively. Then, $K = -R^{-1}(B^T S)$ is found such that asymptotic stability of the closed-loop system ($A_k = A + BK$) is guaranteed in a region about the reference values. Note the sign reversals that are required due to the use of an LQR and the definition for $\mathbf{u}(t)$. As $\sigma^* \notin \{0, \pi\}$, the linearized system is both controllable and observable such that the pair A, B is stabilizable. The existence of a suitable LQR controller is, thus, ensured provided $Q \succeq 0, R \succ 0$.

Second, the maximum real eigenvalue of the closed-loop system $\lambda_{\max}(A_k)$ is found by solving the Lyapunov equation

$$(A_k + \kappa I)^T P + P(A_k + \kappa I) = -(Q + K^T R K) \quad (39)$$

for P . The constant constant κ is chosen such that $0 \leq \kappa < -\lambda_{\max}(A_k)$ to ensure (39) admits a unique positive-definite solution. To ensure a larger terminal region while retaining good control performance, the poles of the closed-loop system are shifted such that $\kappa = -0.9\lambda_{\max}(A_k)$. It was shown in [26] and verified for visual control that κ should be chosen near, but not equal to $-\lambda_{\max}(A_k)$. If $\kappa \ll -\lambda_{\max}(A_k)$, a smaller terminal region would result. If $\kappa = -\lambda_{\max}(A_k)$, then the terminal penalty matrix will be large and may degrade control performance. Note that other values for κ are suitable, but the limits on κ for acceptable performance for VPC have yet to be determined.

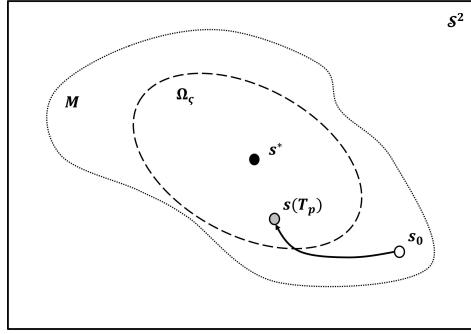


Fig. 5. Example feasible initial image feature s_0 moving from the feasible region M to the terminal region Ω_ζ over the prediction horizon T_p .

C. Terminal Region

The terminal region Ω_ζ is an ellipse on the spherical imaging surface defined such that $\mathbf{s}^T P \mathbf{s} \leq \zeta$ where $\zeta \geq 0$. The method for calculating the terminal region is based on the methodology of [36] and extended to provide an explicit formulation amenable to direct implementation in MATLAB. To find the terminal region Ω_ζ , a series of optimization problems are solved offline, defined by Algorithm 1 and using (40)–(44), shown below. Each optimization problem can be solved using the global optimization function *MultiSearch*. It requires nomination of a local solver and associated solver method. To this end, the *fmincon* function with sequential quadratic programming (SQP) solver can be used.

The terminal region is invariant under the locally linear controller, which itself is fictitious in the sense it is never implemented and only used to derive these controller parameters. Feasibility of the open-loop optimal control problem at the initial time implies the image features will lie within the terminal region at the end of the prediction horizon, ensuring closed-loop asymptotic stability. An example is shown in Fig. 5, where an arbitrary terminal region Ω_ζ and feasible region M are drawn over the spherical surface S^2 . Strictly speaking, $\Omega_\zeta \neq M$, but it is reasonable to assume that the larger the terminal region, the larger the domain of attraction such that $\Omega_\zeta \subseteq M$. Importantly, a larger terminal region will generally result in a larger terminal penalty matrix. The penalty matrix must not be too large such that control performance is degraded. This is an important tradeoff when selecting the prediction horizon. A smaller prediction horizon may be used for a larger terminal region to decrease computational overhead without degrading performance

$$\mathbf{s}^\pi = \operatorname{argmax}_{\mathbf{s}} \|\mathbf{s}\| \quad (40)$$

$$\mathbf{s}^\zeta = \operatorname{argmin}_{\mathbf{s}} \{\kappa \cdot \mathbf{s}^T P \mathbf{s} - \mathbf{s}^T P \phi(\mathbf{s})\} \quad (41)$$

$$\phi(\mathbf{s}) = \mathbf{f}(\mathbf{s}, \mathbf{u}) - A_k \mathbf{s} \quad (42)$$

$$\mathbf{0} \leq K \mathbf{s} - \mathbf{u}^{\min} \quad (43a)$$

$$\mathbf{0} \leq \mathbf{u}^{\max} - K \mathbf{s} \quad (43b)$$

$$0 \leq \varsigma_j - \mathbf{s}^T P \mathbf{s}. \quad (44)$$

Algorithm 1 Terminal Region Ω_ζ Calculation

```

Set  $j = 1$ 
PROBLEM A
Solve (40) subject to (43)
set  $\varsigma_j = \mathbf{s}^\pi P \mathbf{s}^\pi$ 
PROBLEM B
Set  $\mathbf{s}^\zeta = \mathbf{s}^\pi$ 
while  $\mathbf{s}^\zeta > 0$  do
    Solve (40) s.t (43)–(44)
    Solve (41) s.t (42)–(44)
    Set  $j = j + 1$ 
    set  $\varsigma_j = \frac{2}{3} \varsigma_{j-1}$ 
end while
Set  $\varsigma = \mathbf{s}^{\varsigma_{j-1}} P \mathbf{s}^{\varsigma_{j-1}}$ 

```

Remark 1: Finding the true maximal terminal region is difficult, but the approach outlined in this paper can be considered a suitable approximation having verified the results of [36] with the above Algorithm. Note also that solving the first optimization in problem B is generally not required, but its solution is used as an initial value to speed up the second optimization in problem B. Additionally, when using *fmincon* in MATLAB, (40) must be transformed to a minimization problem.

D. Online Implementation

The online QIH-VPC optimization problem can now be defined as

$$\mathbf{u}(\cdot) = \operatorname{argmin}_{\bar{\mathbf{u}}(\cdot)} J(\mathbf{s}(t), \bar{\mathbf{u}}(\cdot)) \quad (45)$$

where

$$\begin{aligned} J(\mathbf{s}(t), \bar{\mathbf{u}}(\cdot)) = & \int_t^{t+T_p} \bar{\mathbf{s}}(\tau)_Q^2 + \bar{\mathbf{u}}(\tau)_R^2 d\tau \\ & + \bar{\mathbf{s}}(t+T_p)_P^2 \end{aligned} \quad (46)$$

s.t

$$\dot{\bar{\mathbf{s}}} = \mathbf{f}(\bar{\mathbf{s}}, \bar{\mathbf{u}}), \quad \bar{\mathbf{s}}(t) = \mathbf{s}_0 \quad (47a)$$

$$\bar{\mathbf{u}}(\tau) \in \mathbb{U}, \quad \tau \in [t, t+T_p] \quad (47b)$$

$$\bar{\mathbf{s}}(t+T_p) \in \Omega_\zeta \quad (47c)$$

where $Q \succ 0$, $R \succ 0$, $P \succ 0$, and $T_p = T_c$ to avoid move blocking. Due to the spherical camera model, visibility constraints do not need to be explicitly considered in the control design.⁵ Note also that because the framework considers a shifted process model, \mathbf{s} can be used instead of \mathbf{e} in J .

V. CIRCULAR MOTION

In this section, the controller performance for circular motion is analyzed. Simulations were conducted using the constraints defined in Table I, which appears in Section VI, as well as the parameters defined in Table II, which also appears in

⁵Visibility constraints can be included to avoid polar regions.

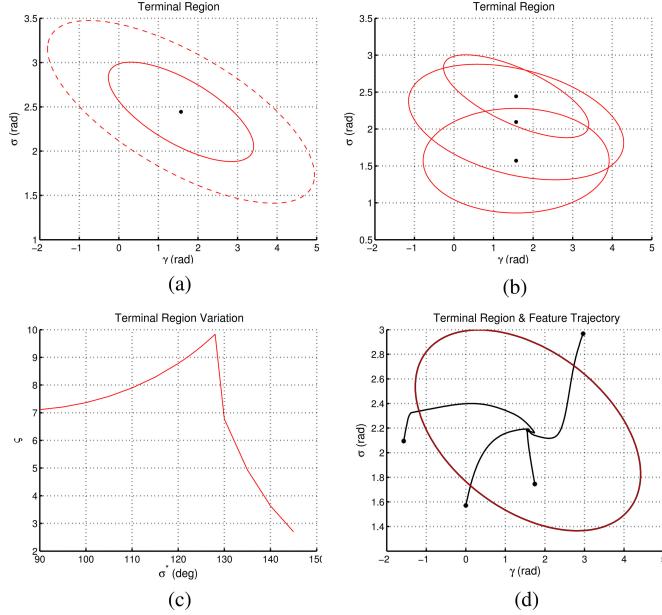


Fig. 6. Terminal region behavior with s^* . (a) Terminal region contraction for $s^* = [140\pi/180 \ \pi/2]$. (b) Terminal regions for $\sigma^* \in \{\pi/2, 2\pi/3, 5\pi/6\}$. (c) Terminal region constant ξ with varied σ^* . (d) Example image feature trajectories for $s^* = [125\pi/180 \ \pi/2]$ and varied s_0 .

Section VI, unless stated otherwise. Solutions to the online optimization problem (45)–(47) were obtained using the ACADO Toolkit [38]. This allowed for the development of deployable code suitable for real applications and optimized for the VPC problem. For the C-VPC case (33)–(35), MATLAB’s *fmincon* was used in order to remain consistent with previous works. Both solvers use an SQP-based Active-Set strategy to solve the nonlinear optimization problem. To ensure a true comparison can be made however, each solvers parameters were adjusted. Using the general form of the objective function for the VPC problem, simulations showed an RMS difference of less than 0.005 between the resulting control output.

A. Reference Image Features

The reference image features s can be chosen arbitrarily within the domain of s . As $\sigma^* \notin \{0, \pi\}$, assumptions A1–A3 are satisfied for circular motion. The reference control $u^* = [0 \ \omega_z^*]$ can then be calculated using simple circular motion equations. The nominal forward velocity v_x^* is defined as $v_x^* = \omega_z^* r_2^*$, where r_2^* defines the radius of the circle on the lateral xy plane such that $r_2^* = r^* \cos(\sigma^* - \pi/2)$.

With $\gamma^* = \pi/2$ and $r^* = 2$, the terminal region variation with σ^* is compared. Fig. 6(a) shows the terminal region centered about $\sigma^* = 5\pi/6$. Fig. 6(b) shows the terminal regions centered about $\sigma^* \in \{\pi/2, 2\pi/3, 5\pi/6\}$. Symmetry about $\sigma^* = \pi/2$ is observed for $\sigma^* < \pi/2$; therefore, results are omitted. For $\pi/2 \leq \sigma^* < 130\pi/180$, the terminal region is dictated by the control constraints. This corresponds to solving only problem A in Algorithm 1. If the control constraint domain is increased for a given σ^* , a larger terminal region will result. In the polar regions, the terminal region is no longer defined primarily due to

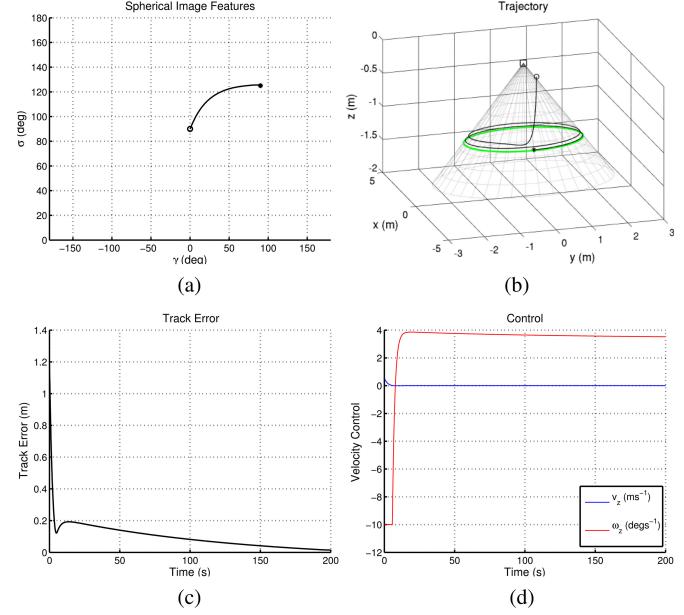


Fig. 7. Example QIH-VPC simulation for $s^* = [125\pi/180 \ \pi/2]$. (a) Image feature trajectory. (b) Camera trajectory. The reference circular trajectory (green) is also shown. (c) Track error. This is defined as the minimum two norm of the difference between any point on the reference circle and the current position. (d) Control effort.

control constraints. Because of the increased nonlinearity of the system, a number of iterations are required to find the terminal region for which the fictitious linear controller guarantees convergence. In Fig. 6(a), the dashed ellipse bounds the terminal region according to the control constraints, while the red ellipse shows the actual terminal region considering the system nonlinearity. This terminal region along with those for two different reference image features is depicted in Fig. 6(b).

The variation in the terminal region constant ξ , which can be considered a proxy for the terminal region area, with σ^* is shown in Fig. 6(c). The region increases until $\sigma^* \approx 128\pi/180$ due to an increased domain for the linear controller, before rapidly decreasing as we approach the polar regions. The result depends on the choice of weighting matrices Q and R , but we can safely assume these to be constant predefined matrices chosen based on some desired vehicle performance. The result is important as it suggests selecting σ^* near $125\pi/180$ may ensure a larger domain of attraction and thus larger feasible set for s_0 . A set of simulations for $\sigma^* = 125\pi/180$ and varied s_0 is shown in Fig. 6(d) with a detailed example for $s_0 = [\pi/2 \ 0]$ shown in Fig. 7. In each simulation, $r_0 = 2$, and the resulting camera trajectory converges to the desired circular path.

For each reference image feature, the terminal region can be used to quantitatively determine the associated degree of nonlinearity and domain of attraction. It could be used to select suitable reference values for spiral tracking, before implementation. In a sense, guiding design choices based on expected performance. Consider an application such as collision avoidance. Choosing $\alpha^* \pm \pi/2$ and $\sigma^* = 125\pi/180$ could ensure a larger set of feasible initial states and therefore provide better assurance that the reference features (and spiral) will be tracked such that avoidance will be achieved.

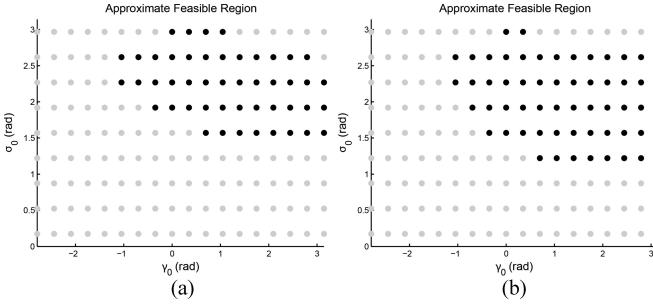


Fig. 8. Feasibility analysis for $s^* = [125\pi/180 \ \pi/2]$ and simulation time of 30 s. Feasible (black) and infeasible (gray) s_0 are shown. (a) Approximate set of feasible initial states for $T_p = 2$. (b) Approximate set of feasible initial states for $T_p = 15$.

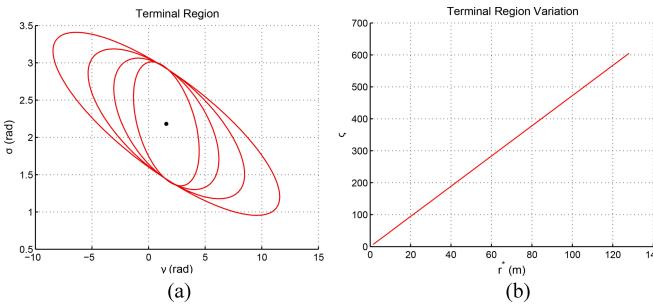


Fig. 9. Terminal region variation with parameter uncertainty for $s^* = [125\pi/180 \ \pi/2]$ (a) Terminal region variation with r^* . The terminal regions from the smallest to the largest correspond to $r^* \in \{2, 4, 8, 16\}$, respectively. (b) Terminal region constant variation with $r^* \in [0.5, 125]$.

Although analytically determining the feasible region may be possible in the nominal case, the difference between the actual system and model suggests that it would be best to sample s_0 over \mathcal{S}^2 and simulate. The resulting state at the end of the first iteration, and whether it lies in the terminal region, can be used to determine feasibility [44]. Such results are depicted in Fig. 8 for short and long horizons by sampling s_0 at $20\pi/180$ intervals over \mathcal{S}^2 . For the sampled set $\hat{\mathcal{S}}^2$, a larger feasible initial state space, and thus stable region of convergence, is noticed for longer prediction horizons. It was noted in [8] that it is difficult to determine the region of convergence for classical IBVS. These results provide some valuable insight into this important issue.

B. Robustness

The quantitative robustness properties in the context of unknown time-varying range parameter and bounded uncertainty on image feature measurements and actuators are investigated. This is important to consider, since the approach is designed for the ideal case. Even for circular motion, $r(t) \neq r^*$ unless already established on the circular path. Based on previous results, we use $s^* = [125\pi/180, \ \pi/2]$. The terminal region variation for $r^* \in \{2, 4, 8, 16\}$ is shown in Fig. 9(a), and the variation in the terminal region constant for $r^* = [0.5, 125]$ is shown in Fig. 9(b). The existence and shape of the terminal regions are important, suggesting a controller designed with a smaller r^* may be suitable when $r(t) > r^*$. This is shown by the successively smaller terminal regions as r^* decreases.

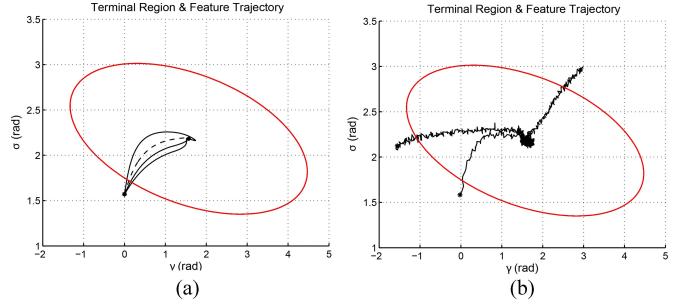


Fig. 10. Robustness characteristics with respect to parameter uncertainty and additive noise. (a) Example image feature trajectories for $r^* = 2$ and $r_0 \in \{1, 2, 3, 4\}$. The dashed line shows $r_0 = 2$. (b) Example image feature trajectories for $s_0 = [125\pi/180 \ \pi/2]$ and additive measurement and process noise $q(t)$.

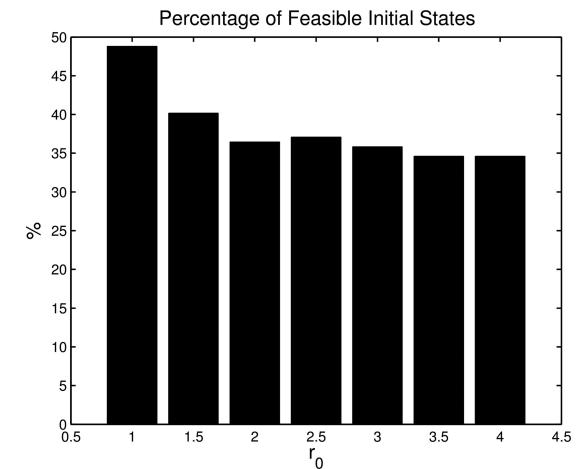


Fig. 11. Example feasible region statistics for $s^* = [125\pi/180 \ \pi/2]$ and simulation time 30 s.

In Fig. 10(a), the controller from the previous section is used, but r_0 is varied instead. In some cases, the initial object range is up to twice that used in the process model. The QIH-VPC controller is shown to handle this large parameter uncertainty, converging to the circular trajectory. Uncertainty on image feature measurements, imperfect actuation, and range parameter uncertainty is then included. Uncertainty in the form of additive white noise $q(t) \sim \mathcal{N}(0, 0.02^2)$ is added, providing consistency with related works [31]. The results are given in Fig. 10(b), showing convergence to the desired reference in all cases. Although the approach is not designed for robustness, uncertainty and model mismatch are managed well.

Some example feasibility statistics for $r_0 = [1.5, 4]$ sampled at 0.5 intervals for all $s_0 \in \hat{\mathcal{S}}^2$ are given in Fig. 11. The approximate percentage of feasible initial features for each initial range is shown. Interestingly, the results are comparable, suggesting that the initial range has minimal impact on the feasible and, thus stable, initial image feature set. Collectively, the results suggest that despite the parameter uncertainty in the process model, performance is not significantly degraded. Similar robustness characteristics are exhibited by classical IBVS, but unlike QIH-VPC, a gain term must be tuned to ensure stable behavior.

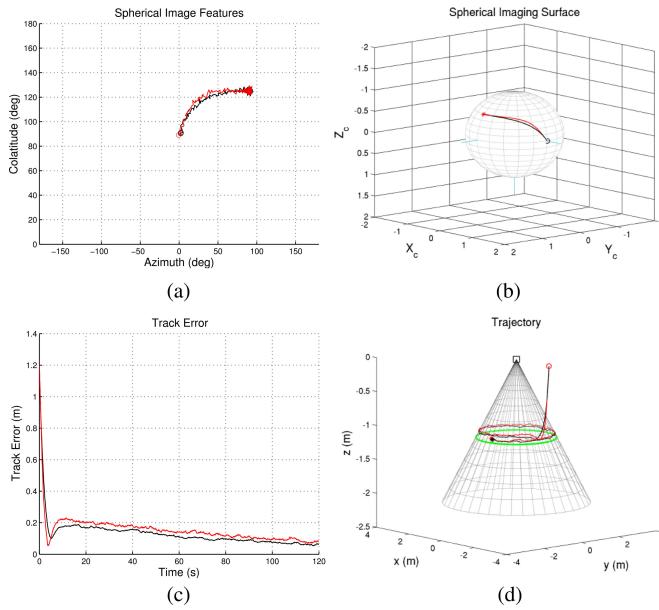


Fig. 12. Example circular motion using C-VPC (red) and QIH-VPC (black). (a) Image feature trajectory. (b) Spherical imaging surface. (c) Track error. This is defined as the minimum two norm of the difference between any point on the reference circle and the current position. (d) Camera trajectory.

Remark 2: One way to guarantee robust stability, without explicitly considering uncertainty in the control design, is to shrink the terminal region along with an appropriately calculated prediction horizon [35]. Other control design such as H_∞ control and LMI explicitly consider a bounded uncertainty on the unknown model parameters, but require the solution of a computationally expensive minimax problem online. These approaches have only recently been applied to visual servoing [21], resulting in unsatisfactory computation expense for implementation.

C. Performance Comparison

A performance comparison is made between the QIH-VPC and C-VPC schemes. The C-VPC must now include a control penalty term $\bar{u}(\tau)_R^2$ in the corresponding objective function (34) for a fair comparison. An example set of results for $s^* = [125\pi/180 \quad \pi/2]$, $r_0 = 2$, and $s_0 = [\pi/2 \quad 0]$ is shown in Fig. 12 for a realistic case, including parameter uncertainty and additive noise $q(t)$. The QIH-VPC provides comparable performance to the C-VPC scheme. Similar image feature trajectories are observed with a marginal reduction in track error for the QIH-VPC scheme. Track error is defined as the minimum two norm of the difference between any point on the reference circle and the current position. Similar robustness qualities are displayed for both predictive schemes, managing model mismatch, and added uncertainty.

The required control effort for both schemes is shown in Fig. 13. The magnitude of the total control effort for each $s_0 \in \hat{S}^2$ for the QIH-VPC and C-VPC schemes is first calculated. Then, their ratio is shown as a 2-D surface plot. A long prediction horizon is used, and a reduction in control effort for the QIH-VPC scheme results in a ratio less than unity. The average ratio is 0.9716, showing a general reduction in control

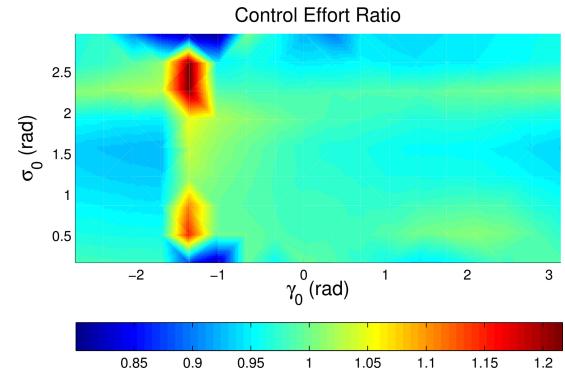


Fig. 13. Comparison of total control effort for $s^* = [125\pi/180 \quad \pi/2]$, $T_p = 15$, and simulation time 30 s. The ratio of control effort for QIH-VPC over C-VPC is shown. A value less than unity depicts reduced control effort using QIH-VPC at the corresponding s_0 . The average ratio is 0.9716.

effort for the QIH-VPC scheme. This translates into the ability to efficiently navigate through all regions of the state space, including particularly nonlinear regions such as the polar caps.

The computational effort is of less impact, as it remains small and suitable for near real-time implementation for both schemes. Using MATLAB 2011b and the ACADO Toolkit running on an Intel Core 2 Duo CPU T8100 at 2.10 GHz with 2G RAM, the average computational time for the online optimisation (45) is under 42 ms.

VI. SPIRAL MOTION

Any deviation from circular motion results in model-mismatch due to varying range parameter. For divergent or convergent spiral motion, condition A1 is then violated as $\omega_z^*(t) = f(r(t))$ such that $u^* = f(t)$. One way to manage this is to include range as a state, but we are unable to predict its derivative and thus variation, until we are established on a spiral. As such, we cannot find a stable locally linear feedback controller for such motion. This results in the terminal region approximating a point: a terminal equality constraint. This is known to lead to stability, but is restrictive in the sense that it forces $s(T_p) = s^*$. In this case, a terminal penalty term does not make sense, and feasibility issues can arise. Alternatively, we could use a terminal penalty without terminal constraints and select an appropriate prediction horizon to aid stability [22]. This is similar to the C-VPC approach in the stability sense; however, the end states are not progressively weighted, and move blocking is not employed. In this section, simulations were conducted using constraints defined in Table I and parameters defined in Table II unless stated otherwise.

A. Performance Comparison

Controllers using a terminal equality constraint, terminal penalty term without terminal constraint, and the C-VPC scheme are compared. An example set of results for $s^* = [125\pi/180 \quad 120\pi/180]$, $r_0 = 2$, and $s_0 = [\pi/2 \quad 0]$ are shown in Fig. 14 for a realistic case including parameter uncertainty and additive noise $q(t)$. In all cases, the differences between the terminal equality constraint and terminal penalty term without terminal constraint are small, with the latter offering greater

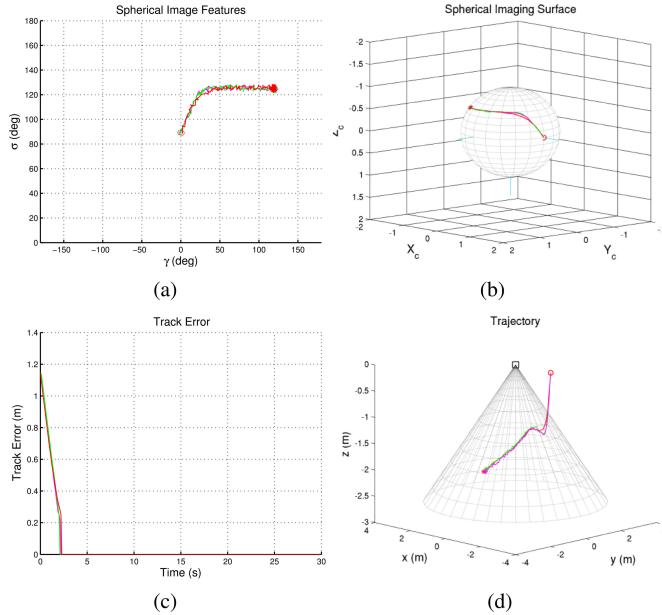


Fig. 14. Example spiral motion using C-VPC (red), terminal penalty term VPC (green), and terminal equality VPC (magenta). (a) Image feature trajectory. (b) Spherical imaging surface. (c) Track error. This is defined as the minimum two norm of the difference between any point on the reference cone and the current position. (d) Camera trajectory.

flexibility in the feasible set for s_0 . The performance of each controller is comparable with C-VPC, with the predictive scheme using only the terminal penalty term showing slightly faster convergence to the surface of the reference cone.

Remark 3: Divergent spirals may be followed for all time provided the object remains visible. For convergent spirals, a limit cycle will be reached that depends on the control constraints. The object will be continually circled at a fixed radius, albeit small for liberal control constraints.

VII. EXPERIMENTAL RESULTS

In this paper, we aim to show how QIH-VPC can be derived and applied to a problem such as spiral tracking. Based on simulated results, and in an attempt to collect some preliminary experimental results, we consider the controller designs in Section V for divergent and convergent spirals. We implement a controller for a small quadrotor using a terminal penalty term only as this can also manage circular motion. Although not strictly QIH-VPC in this case, we believe these are the first experimental results using VPC of any form to control an aerial vehicle. Results are provided to demonstrate practical VPC as a step toward a full QIH-VPC implementation.

The dynamics of a small AscTec Hummingbird quadrotor [45] were determined using black box identification and then included in the process model [25]. The controller parameters were then tuned for the quadrotor dynamics and are defined in Table II. A well-established open source object detection and tracking algorithm was used [46], and the conical apex approximated as the objects' centroid. Using the unified imaging model, the spherical image features were approximated from a perspective camera shifted by $135\pi/180$ from the body x -axis. A Vicon motion capture system provided ground truth

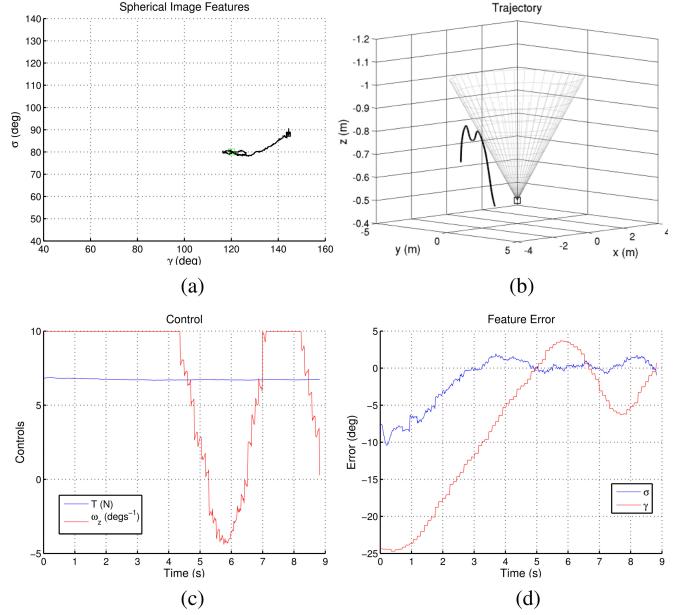


Fig. 15. Spiral motion of the quadrotor with camera in the loop for $s^* = [80\pi/180 \quad 120\pi/180]$ (a) Image features. (b) Vehicle trajectory. (c) Control effort. (d) Image feature error.

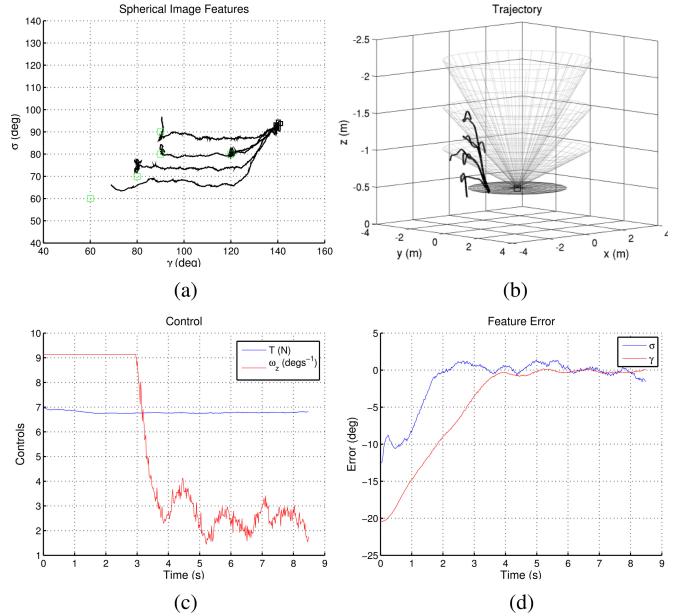


Fig. 16. Spiral motion of the quadrotor with the camera removed (using virtual points) for five separate flight trials with varied s^* . (a) Image features. (b) Vehicle trajectories. (c) Control effort for $s^* = [80\pi/180 \quad 120\pi/180]$. (d) Image feature error for $s^* = [80\pi/180 \quad 120\pi/180]$.

and was not used in the visual controller. With a real application in mind, such as Detect and Avoid style collision avoidance, range is not estimated, and instead a fixed reference value is used. In Fig. 15, the camera is in the loop, resulting in up to 100-ms delay in some cases. In Fig. 16, the camera is removed to omit any image processing delay, and the image features were calculated by assuming a known relative position with added noise (virtual point). This is done to distinguish between

TABLE I
CONTROL CONSTRAINTS

Parameter	Nominal (*)	Min	Max
v_z (ms ⁻¹)	0.00	-0.50	0.50
ω_z (degs ⁻¹)	2.87	-10.00	10.00

TABLE II
SIMULATION AND EXPERIMENTAL PARAMETERS

Parameter	Circular Motion	Spiral Motion	Experimental
T_p	15	15	10
T (s)	0.10	0.10	0.04
r^* (m)	2.00	2.00	2.00
v_x^* (ms ⁻¹)	0.10	0.10	0.20
v_z^* (ms ⁻¹)	0.00	-	-
ω_z^* (degs ⁻¹)	2.87	-	-
κ	0.2635	-	-
α	9.3937	-	-
K	$\begin{pmatrix} -0.7057 & 0.0079 \\ -0.0192 & 0.7078 \end{pmatrix}$	-	-
Q	$0.5I_2$	I_2	$0.01I_2$
R	I_2	0	0
P	$\begin{pmatrix} 17.4134 & 2.1785 \\ 2.1785 & 1.4304 \end{pmatrix}$	$2I_2$	$0.02I_2$
$q(t)$	$\mathcal{N}(0, 0.02^2)$	$\mathcal{N}(0, 0.02^2)$	-

controller performance issues and other issues not addressed in the paper, such as model-predictive control of delayed systems. In both cases, s^* is depicted by a green square.

We notice the effect of image delay immediately. Although the spiral is tracked well in the vertical plane, the delay causes oscillations in yaw which manifests as deviation about the reference azimuth. The image delay does not cause instability, just degraded tracking performance. This is another advantage of using MPC-based control schemes. Once the camera delay is removed, the results improve significantly. The image features are tracked well in both azimuth and colatitude, and the trajectory approximates a conical spiral.

VIII. CONCLUSION

Using stability-based design approaches, a QIH-VPC scheme for spiral tracking using a spherical camera model has been presented. Comparable performance to recently proposed predictive schemes is achieved, including a lower average control effort and robustness to model mismatch and external disturbances. Stability is guaranteed in the nominal case, and feasibility issues are less likely for small prediction horizons. The inclusion of the terminal region constraint provides some unique design advantages. It gives a quantitative representation of the degree of system nonlinearity and the domain of attraction at specific reference image features. This can be used to predetermine suitable reference values for predictive, classical, and region-reaching IBVS schemes.

For circular motion, determining the nominal range value at which the terminal region shrinks to a point would provide insights into the global domain of operation. The study could then be extended to design and implement a real controller for

applications such as Detect and Avoid, with some added safety guarantees. From a theoretical standpoint, a nonlinear controller could be used to derive the terminal region and penalty matrix. It is expected that a terminal region and not a terminal equality constraint may then result for divergent and convergent spirals to aid the design stages.

APPENDIX

The QIH-NMPC framework requires the following conditions to hold for the process model $f(\mathbf{x}(t), \mathbf{u}(t))$ and associated states $\mathbf{x}(t)$ and controls $\mathbf{u}(t)$.

- A1 That $f(\mathbf{x}(t), \mathbf{u}(t))$ is twice continuously differentiable and $f(\mathbf{0}, \mathbf{0}) = \mathbf{0}$. For a general nonzero reference, the process model must be shifted such that $f(\mathbf{x}^*, \mathbf{u}^*) = \mathbf{0}$.
- A2 $\mathbb{U} \subset \mathbb{R}^m$, $\mathbf{0} \in \mathbb{U}$ and contained in the interior of \mathbb{U} .
- A3 $f(\mathbf{x}(t), \mathbf{u}(t))$ has a unique solution for any initial state \mathbf{x}_0 and piecewise continuous $\mathbf{u}(\cdot) \in \mathbb{U}$.

ACKNOWLEDGMENT

The authors would like to thank Endeavour Awards Australia and the Autonomous System Laboratory at ETH Zurich for their partial support of this study, with particular thanks to M. Burri, S. Richter, C. Pradalier, and R. Siegwart.

REFERENCES

- [1] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, "On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the national airspace system," *Progr. Aerosp. Sci.*, vol. 44, nos. 7/8, pp. 503–519, Oct/Nov. 2008.
- [2] Australian Transport Safety Bureau, "Limitations of the see-and-avoid principle," Australian Government, Canberra, A.C.T., Australia, Tech. Rep., Nov. 2004.
- [3] O. Shakernia, M. Chen, and V. Raska, "Passive ranging for UAV sense and avoid applications," in *Proc. AIAA Infotech@Aerospace Conf.*, Mar. 2005, pp. 1–10.
- [4] R. Gray, "Spiral guidance law for passive range estimation," *Proc. SPIE*, vol. 1950, pp. 225–233, Oct. 1993.
- [5] T. Bruggemann and L. Mejias, "Airborne collision scenario flight tests: Impact of angle measurement errors on reactive vision-based avoidance control," in *Proc. 15th Aust. Aerosp.*, pp. 661–671, Feb. 2013.
- [6] K. Boyadzhiev, "Spiral and conchospirals in the flight of insects," *Coll. Math. J.*, vol. 30, no. 1, pp. 22–31, Jan. 1999.
- [7] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servoing," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.
- [8] P. Chaumette and S. Hutchinson, "Visual servo control part I: Basic approaches," *IEEE Robot. Autom.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.
- [9] R. Carnie, R. Walker, and P. Corke, "Image processing algorithms for UAV sense and avoid," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2006, pp. 2848–2853.
- [10] J. Lai, J. Ford, L. Mejias, and P. O'Shea, "Characterization of sky-region morphological-temporal airborne collision detection," *J. Field Robot.*, vol. 30, no. 2, pp. 171–193, Mar/Apr. 2013.
- [11] L. Mejias, I. Mondragon, and P. Campoy, "Omnidirectional bearing-only see and avoid for small aerial robots," in *Proc. IEEE Int. Conf. Autom., Robot. Appl.*, Dec. 2011, pp. 23–28.
- [12] A. D. Mcfadyen and L. Mejias, "Visual servoing approach to collision avoidance for aircraft," in *Proc. 28th Int. Congr. Aeronautical Sci.*, Sep. 2012, pp. 24–28.
- [13] L. Mejias, J. Ford, and J. Lai, "Towards the implementation of vision-based UAS sense-and-avoid system," in *Proc. 27th Int. Congr. Aeronautical Sci.*, Sep. 2010, pp. 1–10.
- [14] A. Mcfadyen, P. Corke, and L. Mejias, "Rotorcraft collision avoidance using spherical image-based visual servoing and single point features," in *Proc. IEEE/RSJ Int. Conf. Robot. Intell. Syst.*, Oct. 2012, pp. 10–10.

- [15] J. Saunders and R. Beard, "Vision-based reactive multiple obstacle avoidance for micro air vehicles," *Proc. Amer. Control Conf.*, Jun. 2009, pp. 5253–5258.
- [16] R. Sharma, J. Saunders, and R. Beard, "Reactive path planning for micro air vehicles using bearing only measurements," *J. Intell. Robot. Syst.*, vol. 65, pp. 409–416, Jan. 2012.
- [17] X. Yang, L. Mejias, and T. Bruggemann, "A 3D collision avoidance strategy for UAVs in an non-cooperative environment," *J. Intell. Robot. Syst.*, vol. 70, no. 1–4, pp. 315–327, Apr. 2013.
- [18] G. Chesi and Y. Hung, "Global path-planning for constrained and optimal visual servoing," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 22–31, Oct. 2007.
- [19] M. Kazemi, K. Gupta, and M. Mehrandezh, "Path-planning for visual servoing: A review and issues," in *Visual Servoing Via Advanced Numerical Methods*. New York, NY, USA: Springer, 2010, pp. 189–207.
- [20] P. Danes and D. Bellot, "Towards an LMI approach to multi criteria visual servoing in robotics," *Eur. J. Control*, vol. 12, no. 1, pp. 86–110, 2006.
- [21] T. Wang, W. Xie, G. Liu, and Y. Zhao, "Quasi-min-max model predictive control for image-based visual servoing," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2012, pp. 98–103.
- [22] R. Findeisen and F. Allgower, "An introduction to nonlinear model predictive control," in *Proc. 21st Benelux Meet. Syst. Control*, 2002, pp. 119–141.
- [23] Y. Wang, H. Lang, and C. De Silva, "An autonomous grasping system using visual servoing and nonlinear model predictive control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 86–91.
- [24] D. Lee, H. Lim, and H. Kim, "Obstacle avoidance using image-based visual servoing integrated with nonlinear model predictive control," in *Proc. IEEE Conf. Decision Control*, 2011, pp. 5689–5694.
- [25] A. Mcfadyen, L. Mejias, P. Corke, and C. Pradalier, "Aircraft collision avoidance using spherical visual predictive control and single point features," in *Proc. IEEE/RSJ Int. Conf. Robot. Intell. Syst.*, Nov. 2013, pp. 50–56.
- [26] H. Chen and F. Allgower, "Nonlinear model predictive control schemes with guaranteed stability," in *Nonlinear Model Based Process Control*. Norwell, MA, USA: Kluwer, 1998, pp. 465–494.
- [27] M. Sauvee, P. Poignet, and E. Dombre, "Ultrasound image-based visual servoing of a surgical instrument through nonlinear model predictive control," *Int. J. Robot. Res.*, vol. 27, pp. 25–40, 2008.
- [28] A. Burlacu, C. Copot, E. Vercera, and C. Lazar, "Real time visual predictive control of manipulation systems," in *Proc. IEEE Int. Conf. Adv. Robot.*, 2011, pp. 383–388.
- [29] A. Burlacu and C. Lazar, "Reference trajectory-based visual predictive control," *Adv. Robot.*, vol. 26, no. 8–9, pp. 1035–1054, Jul. 2012.
- [30] R. Coagienard, P. Grieder, E. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *J. Process. Control*, vol. 17, no. 6, pp. 563–570, Jul. 2007.
- [31] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Trans. Automat. Control*, vol. 25, no. 5, pp. 933–939, Sep. 2010.
- [32] J. Garriga and M. Soroush, "Model predictive control tuning methods: A review," *Ind. Eng. Chem. Res.*, vol. 49, no. 8, pp. 3505–3515, Mar. 2010.
- [33] D. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Trans. Automat. Control*, vol. 35, no. 7, pp. 814–824, Jul. 1990.
- [34] D. Mayne, J. Rawling, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.
- [35] H. Michalska and D. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Trans. Automat. Control*, vol. 38, no. 11, pp. 1623–1633, Nov. 1993.
- [36] H. Chen and F. Allgower, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, Oct. 1998.
- [37] J. Lofberg, "YALMIP: a toolbox for modelling and optimization in MATLAB," in *Proc. IEEE Int. Symp. Comput. Aided Control Syst. Design*, Sep. 2004, pp. 284–289.
- [38] B. Huska and H. Ferreau, and M. Diehl, "ACADO Toolkit—An open source framework for automatic control and dynamic optimization," in *Optimal Control Applicat. Methods*, vol. 32, no. 3, pp. 298–312, May-Jun. 2011.
- [39] A. Remazeilles, N. Mansard, and F. Chaumette, "A qualitative visual servoing to ensure the visibility constraint," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2006, pp. 4297–4303.
- [40] P. Corke and M. Good, "Dynamic effects in visual closed-loop systems," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 671–683, Oct. 1996.
- [41] P. Corke and R. Mahony, "Sensing and control on the sphere," in *Robot. Res.*, 2011, vol. 70, pp. 71–85 Springer: Berlin, Heidelberg, Germany, 2011.
- [42] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications," in *Proc. Eur. Conf. Comput. Vis.*, Jun.-Jul. 2000, pp. 445–461.
- [43] P. Corke, "Spherical image-based visual servoing and structure estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 5550–5555.
- [44] W. Chen, D. Ballance, and J. O'Reilly, "Model predictive control of nonlinear systems: computational burden and stability," *IEEE Control Theory Appl.*, vol. 147, no. 4, pp. 387–394, Jul. 2000.
- [45] M. Burri, Y. Nikolic, C. Hurzeler, G. Caprari, and R. Siegwart, "Aerial service robots for visual inspection of thermal power plant boiler systems," in *Proc. Int. Conf. Applied Robot. Power Ind.*, Sep. 2012, pp. 70–75.
- [46] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, (2000). [Online]. Available at: <http://code.opencv.org/projects/opencv/wiki/CiteOpenCV>



Aaron Mcfadyen (S'14) received the Bachelor of Engineering (aerospace avionics) degree from Queensland University of Technology, Brisbane, Qld., Australia. He is working toward the Ph.D. degree with the Australian Research Centre for Aerospace Automation.

He is a Research Engineer with the Australian Research Centre for Aerospace Automation, Queensland University of Technology. His research interests include robotic vision and model-predictive control for autonomous aircraft (UAS) control.



Peter Corke (F'08) received the Ph.D. degree with the University of Melbourne, Parkville, Vic., Australia.

He is a Professor of robotic vision with the School of Electronic Engineering and Computer Systems, Queensland University of Technology, Brisbane, Australia, and Director of the ARC Centre of Excellence for Robotic Vision. His research interests include robotic vision, field robotics, and control.



Luis Mejias (M'08) received the Bachelor of Engineering (electronic) degree from UNEXPO, Barquisimeto, Venezuela, and the Master and Ph.D. degrees from Universidad Politécnica de Madrid, Madrid, Spain.

He is a Senior Lecturer in aerospace avionics with the Queensland University of Technology, Brisbane, Qld., Australia, and a Research Scientist with the Australian Research Centre for Aerospace Automation in the same university. His research interests include computer vision for UAS navigation.