Balancing Mission Requirement for Networked Autonomous Rotorcrafts Performing Video Reconnaissance

N. R. Gans, J.W. Curtis
Air Force Research Laboratory
Eglin AFB, FL, 32542, USA

P. Barooah, J. M. Sheah, W. E. Dixon
University of Florida
Gainesville FL, 32608, USA

Abstract

Current and emerging missions, including active surveillance and tracking, terminal guidance and search and rescue, require image-feedback from camera-equipped vehicles. Certain mission scenarios and sensor restrictions may require the collaboration of assets over an ad-hoc network. The development in this paper extends efforts to balance trade-offs between asset/sensor cone positioning to satisfy mission requirements and network requirements such as maintaining network connectivity. To address the trade-offs between asset positioning and network connectivity, a prioritized task-function based guidance law is developed for a simple scenario containing three assets tracking teams of mobile targets. One developed task-function maintains a communication network by ensuring the distance between the UAV's does not exceed a critical threshold. Additional task-functions enable assets to keep targets of interest in the image cone by regulating image features derived from the camera view. The UAV's are modeled as rotorcraft using an LTI model equipped with a gimbaled camera. Early simulation results are provided to examine the behavior of the assets for different configurations of objects observed by the asset cameras. Future efforts will seek to improve performance by modifying the control law, possibly adding time varying tasks shaped by trajectory planning techniques.

I. Introduction

Unmanned Aerial Vehicles (UAV's) are equipped with imaging sensors in many scenarios, including image-based guidance, navigation, and control; geolocation; environment mapping; target tracking and surveillance; etc. A historical problem in image-based estimation and control literature is controlling the motion of the camera to ensure that targets of interest stay in the field-of-view (FOV) (cf. [1]–[5]). This problem is especially difficult when the camera is monitoring an adversarial target in a dynamic environment, or when it is mounted on a vehicle with motion constraints. A practical approach to alleviate the FOV problem is to use multiple imaging sources [6], [7], such as cooperative camera-equipped UAV's.

The use of cooperative assets is a well-accepted approach. Current and emerging scenarios (e.g., wide area surveillance, environmental monitoring, search and rescue/destroy) can exploit coordination between multiple assets for improved performance. In many applications, such as tactical operations, portions of the network linking the collaborative assets may entail a mobile ad hoc network (MANET). Assets operating within a MANET are faced with challenges, such as maintaining connectivity due to the changing topology and link conditions of the network. These problems are exacerbated by high mobility, which is characteristic of networks of UAV's.

Fig. 1 illustrates an example scenario of a network of UAV's with cameras. In this scenario, a network of UAV's is tracking red forces and blue forces in an urban environment. The imaging goals are challenging because the urban environment limits the FOV, and the cameras are attached to UAV's with motion constraints. Coordinating the UAV's is challenging because network connectivity must be maintained in the presence of path loss, shadowing, and/or multi-path fading. Even with perfect location information, maintaining a fully connected network topology can be challenging. The potential tactical advantages of the scenario in Fig. 1 motivate the need for directed methods that manage trade-offs between UAV/sensor cone positioning to satisfy mission requirements and network requirements to ensure effective collaboration between the UAV's. Yet, literature that focuses on such issues appears sparse.

This paper presents an extension of our earlier work [8] addressing some of the challenges imposed by balancing camera positioning for image tasks against UAV positioning for efficient network operation. Motivated by the scenario depicted in Fig. 1, the imaging task involves a collection of UAV's keeping a set of moving objects in the camera FOV's, while also positioning the UAV's to ensure network connectivity. For example, one UAV tracks the blue force, while the other UAV's monitor sub-groups of the red force. The network is modeled as undirected, and successful communication is assumed possible if two nodes are within some specified maximum-link distance. The investigation is extended over our previous work by including realistic motion constraints on the camera using a rotorcraft model [9] with a single gimbaled camera.

Postdoctoral Associate, National Research Council and Air Force Research Laboratory Eglin AFB, FL, ngans@ufl.edu, Member AIAA
Research Scientist, Air Force Research Laboratory, Eglin AFB, FL, jess.curtis@eglin.af.mil, Member AIAA
Assistant Professor, Dept. of Mechanical & Aerospace Engineering, Gainesville, FL, pbarooah@ufl.edu
Associate Professor, Dept. of Electrical & Computer Engineering, Gainesville, FL, jshea@ece.ufl.edu
Associate Professor, Dept. of Mechanical & Aerospace Engineering, Gainesville, FL, wdixon@ufl.edu, Member AIAA

This research is supported in part by the US Department of Energy grant number DE-FG04-86NE37967. This work was accomplished as part of the DOE University Research Program in Robotics (URPR). This research was performed while Nicholas Gans held a National Research Council Research Associateship Award at the Air Force Research Laboratory.

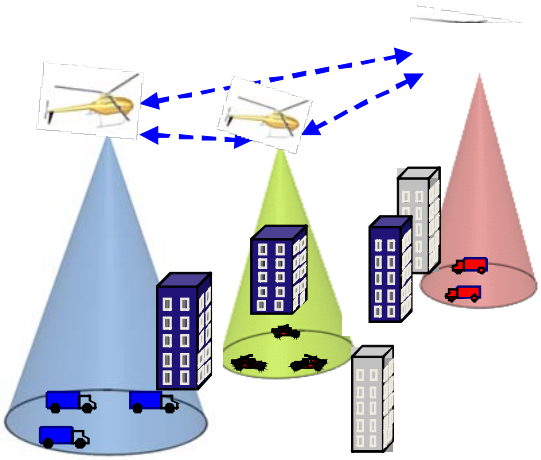

Fig. 1. Networked collection of UAV's performing image-based red force and blue force tracking in an urban environment.

To manage imaging and network trade-offs, a series of low dimensional task-functions are defined, based on current image features and the distance to the nearest neighbor in the network. For the image-based task functions, the desired task-function velocity is mapped to a time-varying feature velocity through corresponding task-function Jacobians. These task-function Jacobians are underdetermined and are suitable for task-priority kinematic control [10], [11]. The time-varying image feature velocity is mapped to camera motions through image-based visual servoing (IBVS) methods [12]. The resulting controller allows features to move within the image, and the camera will move to keep the features in the FOV. To maintain network continuity, an additional task is introduced to regulate the position of each camera to maintain distance to its nearest neighbor. If each UAV is within a maximum distance to its nearest neighbor, the network of three nodes will stay simply connected.

Underdetermined task-function based approaches have been developed for kinematic control of vehicle platoons in [13], [14]. Full-rank task functions were used in vision-based control in [15]. In contrast, we utilize low-rank (underdetermined) utility functions, which impose fewer constraints on camera motion. The initial development of these underdetermined functions appeared in [16].

The problem of group coordination of mobile robots for connectivity maintenance has been studied in several papers (cf. [17]–[20]). Most of these studies are concerned with the robots moving toward a goal destination while maintaining wireless connectivity. In other words, the position of the mobile agents – if there were no need to maintain connectivity – would tend to certain fixed location in space depending on the motion objective. In contrast, this paper focuses on the situation when there are no limiting goal positions. Rather, the unpredictable motion of the targets may cause the UAV's to move in a way that will sever connectivity in the absence of a connectivity maintaining control law.

Maintaining connectivity among moving agents with second order dynamics has been examined in [21], [22]. However, the uncontrolled motion of the agents in [21] was only due to initial velocities and second-order dynamics, not due to the need for maintaining targets within the FOV. A similar problem formulation was also considered in [22] for first order dynamic agents. In contrast to these studies, this paper examines the situation when sensing and maintaining connectivity are in conflict, e.g., when motion of the targets may cause the UAV's to move away from one another while maintaining connectivity may require them to stay close.

II. Rotorcraft, Camera and Network Model

Consider a velocity vector for a rotorcraft given by

$$v_h(t) = [v_{xh}, v_{yh}, v_{zh}, \omega_{xh}, \omega_{yh}, \omega_{zh}]^T \in \mathbb{R}^6 \tag{1}$$

where $v_x(t)$, $v_y(t)$ and $v_z(t)$ describe the linear velocity, and $\omega_x(t)$, $\omega_y(t)$ and $\omega_z(t)$ are the angular velocity, all measured in the body fixed frame such that the x-axis is oriented along the craft major axis (from tail to nose), the y-axis points out the right side of the rotorcraft and the z-axis is oriented through the bottom of the rotorcraft body.

A. Rotorcraft Model

The model used here was first developed by Shim, et al., [9]. It is a linear time-invariant model of a rotorcraft in a near hover state, giving six degree of freedom (DOF) motion of the rotorcraft. The model is given by

$$\dot{q}_h = A_h q_h + B_h u_h. \tag{2}$$

where

$$q_h = [v_{xh}, v_{yh}, \omega_{xh}, \omega_{yh}, \Phi, \Theta, a, b, v_{zh}, \omega_{zh}, r]^T \in \mathbb{R}^{11} \quad (3)$$

$$u = [u_1, u_2, u_3, u_4]^T. \quad (4)$$

In (3), $\Phi(t)$, $\Theta(t)$ are the roll and pitch angle, $r(t)$ is a feedback gyro rate, and $a(t)$, $b(t)$ are the blade flapping angles. In (4), $u_1(t)$ and $u_2(t)$ are inputs to the flap angles, $u_3(t)$ is an input to the pitch angle, and $u_4(t)$ is an input to the yaw rate. The state matrix $A \in \mathbb{R}^{11\times 11}$ and input matrix $B \in \mathbb{R}^{11\times 4}$ are given by

$$A_h = \begin{bmatrix} -0.0954 & 0 & 0 & 0 & 0 & -g & g & 0 & 0 & 0 & 0 \\ 0 & -0.2221 & 0 & 0 & g & 0 & 0 & g & 0 & 0 & 0 \\ -0.2047 & 0.1521 & 0 & 0 & 0 & 0 & 22.14 & 32.995 & 0 & 0 & 0 \\ -0.0836 & -0.0514 & 0 & 0 & 0 & 0 & 67.74 & 142.50 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -2.6645 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0.5543 & -2.6645 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -28.85 & -121.2 & -0.5377 & 5.7974 & 0 \\ 0 & 0 & -0.0178 & 0 & 0 & 0 & 0 & 0 & .0746 & -4.4017 & -46.959 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.3394 & -5.483 \end{bmatrix}$$

$$B_h = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.5912 & 1.9729 & 0 & 0 \\ -2.4055 & -0.0993 & 0 & 0 \\ 0 & 0 & 116.952 & 0 \\ 0 & 0 & -0.0178 & 0 \\ 0 & 0 & -46.969 & 15.2454 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where the scalar g is the gravitational constant. See [9] for more details, including the eigenvalues of the developed system.

Shim, et al., develop a stabilizing controller composed of three control loops [9]. The first control loop controls rotorcraft attitude, the second loop control the linear velocity, and the outer loop controls position. In contrast, we propose a multi-input multi-output (MIMO) control strategy. Using LQR design to solve for a constant gain matrix K_h, (2)-(4) can be rewritten as

$$\dot{q}_h = A_h q_h + B_h K_h q_d.$$

The input signal $q_d(t) \in \mathbb{R}^{11}$ is given by

$$q_h = [v_{xhd}, v_{yhd}, 0, 0, 0, 0, 0, 0, v_{zhd}, \omega_{zhd}, 0]^T \quad (5)$$

where the subscript d designates a desired value of the appropriate variable. This enables the rotorcraft to track a desired trajectory and heading in Euclidean space,

B. Camera Model

Consider a camera with coordinate frame $\mathcal{F}_c(t)$ as shown in Fig. 2. The camera views a collection of k feature points in front of the camera. These points have coordinates $M_i(t) \in \mathbb{R}^3$ defined as

$$M_i = [X_i, Y_i, Z_i]^T, \ \forall i \in \{1 \dots k\}$$

in the camera frame. An image of the points is captured, resulting in a projection to a set of points in the image plane. These image points are given by the normalized coordinates

$$\breve{m}_i = [\frac{X_i}{Z_i}, \frac{Y_i}{Z_i}, 1]^T = [x_i, y_i, 1]^T, \ \forall i \in \{1 \dots k\}.$$

Since the last element of the three-dimensional normalized coordinates is superfluous, it will not be considered in the sequel. Define the coordinates $m_i(t) \in \mathbb{R}^2$ as

$$m_i = [x_i, y_i]^T \quad (6)$$

with velocity $\dot{m}_i(t) \in \mathbb{R}^2$ in the image plane given by

$$\dot{m}_i = [\dot{x}_i, \dot{y}_i]^T. \tag{7}$$

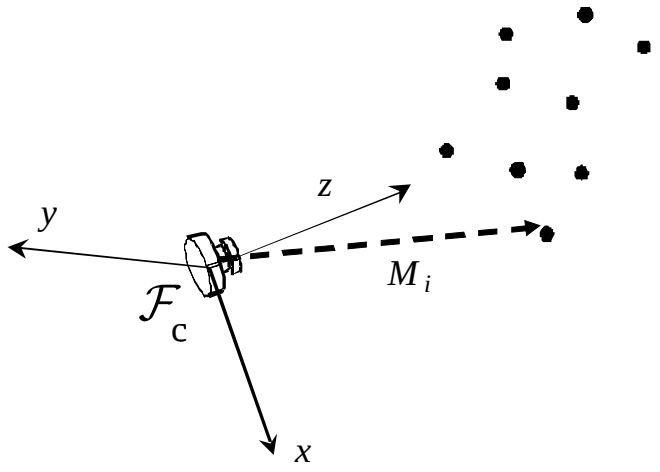

Fig. 2. Camera model

Given the collection of k feature points, along with their coordinates and velocity vectors, a state position vector $m(t)$ and state velocity vector $\dot{m}(t)$ are defined as

$$m = [m_1^T, m_2^T, \cdots, m_k^T]^T$$
$$\dot{m} = [\dot{m}_i^T, \dot{m}_2^T, \cdots, \dot{m}_k^T]^T.$$

The velocity of the features points in the image plane is given as a function of the camera velocity $v_c(t) = [v_{xc}, v_{yc}, v_{zc}, \omega_{xc}, \omega_{yc}, \omega_{zc},]^T \in \mathbb{R}^6$ by the relationship

$$\dot{m} = Lv_c, \tag{8}$$

where $L(t) \in \mathbb{R}^{2k\times 6}$ is the image Jacobian. The image Jacobian for the feature points, $L(t) \in \mathbb{R}^{2k\times 6}$, is given by concatenating a set of k submatrices $L_i(t) \in \mathbb{R}^{2\times 6}$ [15], with $L_i(t)$ given as

$$L_i = \begin{bmatrix} \frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & -x_i y_i & 1+x_i & -y_i \\ 0 & \frac{1}{Z_i} & \frac{y_i}{Z_i} & -1-y_i^2 & x_i y_i & x_i \end{bmatrix}. \tag{9}$$

In the case that the feature points are not static in an inertial frame world frame (e.g., the feature points are tracked on moving objects), the time derivative of the feature points is given by

$$\dot{m} = Lv_c + \varepsilon, \tag{10}$$

where $\varepsilon(t)$ is an unknown, bounded function.

In this development, the camera is attached to the body of the helicopter. For simplicity, we assume the origin of the camera-fixed axes corresponds to the origin of the rotorcraft-fixed axes. The camera has a single angular degree of freedom (such as provided by a single gimbal) about the camera y-axis, and the camera y-axis and rotorcraft y-axis are colinear. Thus, the camera has linear and angular velocity given by

$$\begin{bmatrix} v_{xc} \\ v_{yc} \\ v_{zc} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} v_{xh} \\ v_{yh} \\ v_{zh} \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} \omega_{xc} \\ \omega_{yc} \\ \omega_{zc} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \omega_{xh} \\ \omega_{yh} \\ \omega_{zh} \end{bmatrix} + \begin{bmatrix} \omega_\theta \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

where $\theta(t)$ is the angle of the camera relative to the rotorcraft (e.g. the gimbal angle), and $\omega_{\theta c}(t)$ is the angular rate of the camera (e.g. the gimbal angular rate). The relationships in (11) and (12) can be inverted to give the controlled rotorcraft

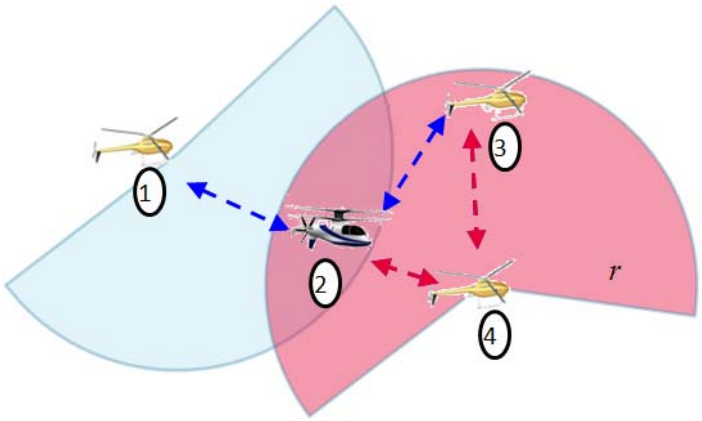

Fig. 3. Illustration of network model

velocity $v_{hd}(t)$ as a function of a desired camera velocity $v_{cd}(t)$ as

$$\begin{bmatrix} v_{xhd} \\ v_{yhd} \\ v_{zhd} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} v_{xcd} \\ v_{ycd} \\ v_{zcd} \end{bmatrix} \tag{13}$$

$$\begin{bmatrix} \omega_{xhd} \\ \omega_{yhd} \\ \omega_{zhd} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \omega_{xcd} \\ \omega_{ycd} \\ \omega_{zcd} \end{bmatrix} - \begin{bmatrix} \omega_{\theta c} \\ 0 \\ 0 \end{bmatrix}. \tag{14}$$

C. Network Model

The network of N assets is modeled as a graph $G = (V, E)$ with N vertices. Each UAV $j,\ j \in [1, \ldots, N]$ is a vertex of the graph, and is located at a position T_j. There is an edge $E(i, j)$ between two assets i and $j,\ i \neq j$, if the distance between them is less than a prespecified positive number r, i.e., if $d_{ij} < r$, where the distance is defined as

$$d_{ij} = \|T_i - T_j\|,$$

and r is a communication range. This is illustrated in Fig. 3. This paper considers only the case $N = 3$. In this case, the graph remains connected if each vertex has an adjacency of at least one, i.e. each UAV is within a radius r to at least one other UAV.

III. Task Function-Based Kinematic Control

The guidance and control objective of this paper is to keep a set of feature points within each camera FOV while regulating distance between rotorcrafts. To achieve these objectives, a set of task functions are defined using image feature coordinates and the relative positions of the rotorcrafts. However, the task functions may compete in the sense that reducing the error of one task function may increase the error of another. To avoid competition between task functions, a task-priority kinematic control [10], [11] is used.

Let $\phi(t) \in \mathbb{R}^n$ denote some task function of the feature point coordinates $m_i\,(t)$ as

$$\phi = f(m_1, \ldots, m_k)$$

with derivative

$$\dot{\phi} = \sum_{i=1}^{k} \frac{\partial f}{\partial m_i} \dot{m}_i = J(m)\dot{m}, \tag{15}$$

where $J(m) \in \mathbb{R}^{n \times 2k}$ is the task Jacobian matrix. The task functions in this paper are of dimension $n \leq 2$.

The objective is to drive the feature points along a desired velocity $\dot{m}_\phi(t)$ such that $\phi(t)$ follows a desired trajectory $\phi_d(t)$. Given the underdetermined structure of the Jacobian matrix, there are an infinite number of solutions to this problem. The typical solution based on the minimum-norm approach [11] is given as

$$\begin{aligned} \dot{m}_\phi &= J^\dagger \left[\dot{\phi}_d - \lambda(\phi - \phi_d)\right] \\ &= J^T \left(JJ^T\right)^{-1} \left[\dot{\phi}_d - \lambda(\phi - \phi_d)\right], \end{aligned} \tag{16}$$

where λ is a positive scalar gain constant, and $J^{\dagger}(m) \in R^{2k\times n}$ denotes the minimum-norm general inverse of $J(m)$. The desired feature point derivative $\dot{m}_{\phi}(t)$ is then mapped to a camera velocity. As discussed in Section II-B, the image Jacobian $L(t)$ maps the camera velocity $v_c(t)$ to the feature derivative. In turn, $v_c(t)$ is mapped to $v_h(t)$ via (13) and (14). Given the velocity and state constraints imposed by the rotorcraft model, the following procedure is used to generate control terms for the rotorcraft and camera gimbal to achieve a feature point velocity as close as possible to the desired velocity $\dot{m}_{\phi}$.

Define a five-element vector of the camera velocity terms linked to rotorcraft velocity and the gimbal velocity, $v_{c5}(t) = [v_{xc}(t), v_{yc}(t), v_{zc}(t), \omega_{\theta c}(t), \omega_{zc}(t)]^T$. The corresponding five column image Jacobian $L_5(t)$ is composed of the first, second, third, fourth and sixth columns of $L(t)$, as defined in (9). Define a six-element feed-forward vector of camera velocity terms that correspond to uncontrolled rotorcraft velocity terms, $v_{cff}(t) = [0, 0, 0, \omega_{xc}(t), \omega_{yc}(t), 0]^T$. Based on (16), the desired camera velocity $v_{c5d}(t)$ is designed as

$$v_{c5d} = L_5^{+}(\dot{m}_{\phi} + Lv_{cff}) = \left(L_5^T L_5\right)^{-1} L_5^T(\dot{m}_{\phi} - Lv_{cff}), \tag{17}$$

where $L_5^{+}(t) \in \mathbb{R}^{5\times 2k}$ denotes the least squares general inverse used for $L_5(t)$[1]. The gimbal velocity is then given by $\omega_{\theta c}(t)$, and the remaining camera velocity terms are then mapped to rotorcraft velocity $v_h(t)$ via (13) and (14).

A set of task functions can be combined in various ways. The simplest approach is to add the desired feature velocity from each task function to create the total desired feature point velocity. For example, consider two tasks $\phi_a(t)$ and $\phi_b(t)$ with resulting desired feature velocities $\dot{m}_a(t)$ and $\dot{m}_b(t)$. The camera velocity is then given as

$$v_{c5d} = L_5^{+}\left(\dot{m}_a + \dot{m}_{\phi b} - Lv_{cff}\right). \tag{18}$$

Since the task function velocities are undetermined, an optional method is to choose one task as primary, and project the other tasks into the null space of the primary task derivative [10], [11] as

$$\begin{aligned} v_{c5d} &= L_5^{+}\left(\dot{m}_a + (I - J_a^{\dagger}J_a)\dot{m}_b - Lv_{cff}\right) \\ &= L_5^{+}\left(J_a^{\dagger}\dot{\phi}_a + (I - J_a^{\dagger}J_a)J_b^{\dagger}\dot{\phi}_b - Lv_{cff}\right), \end{aligned} \tag{19}$$

where $J_a(m_a)$ and $J_b(m_b)$ are the task Jacobian matrices with respect to $\phi_a(t)$ and $\phi_b(t)$, respectively.

The approach in (19) will prevent the individual task systems from competing and negating each other, as the primary task will always be accomplished. Lower priority control tasks will be achieved to the extent that they do not interfere with higher priority tasks. Tertiary, quaternary, etc. tasks can be prioritized by repeating this process and projecting each subsequent task into the null space of the preceding task Jacobians.

IV. Control Development

In this section, four task functions are presented as part of a distributed, task-priority kinematic controller. The objective is to keep three sets of feature points within the FOV of three cameras mounted on UAV's, while maintaining a communication network between the UAV's. Each camera is dedicated to observing a single set of feature points.

The first task function regulates the distance to the nearest camera, which will maintain a network connection for three cameras. Two task functions are designed to regulate the mean and variance of feature point coordinates. Regulating the mean at the camera center will keep the feature points centered in the FOV. Regulating the variance will restrict the distance between the feature points and keep features away from the edge of the FOV. The third task function maximizes motion perceptibility, which ensures desired image velocities can be met. These task functions are cascaded through null space projection and mapped to camera velocity, as described Section III.

Chebyshev's inequality proves that at least 75% of all values are within two standard deviations of the mean, and at least 89% of values are within three standard deviations. For a normally distributed random process, these limits are tighter, such that approximately 95% of all values will be with two standard deviations, and 99.7% of all values will be within three standard deviations. Consider a camera with a 512x512 pixel FOV and normally distributed feature points in the image plane. Regulating the mean of the feature point coordinates to the image center and the variance of the feature point coordinates to 128^2 will ensure that at least 95% of all points are in the FOV. For arbitrary distribution of feature points (e.g. uniformly distributed) points, regulating the variance to 86^2 will ensure that at least 89% of all points are in the FOV.

A. Task Function to Nearest Neighbor

For three camera-equipped assets modeled as a proximity graph, maintaining network connectivity can be modeled as regulating the distance from each UAV to its nearest neighbor. Network connectivity will be maintained if every UAV remains within a bounded distance r to at least one other UAV. To give the assets freedom of motion, it is desirable that the distance regulation function not be active until the distance δ_{ij} is beyond a certain threshold $\underline{r} < r$. To this end, we propose

[1]Since $J(m)$ is underdetermined (i.e. more columns than rows), and $L_5(t)$ is overdetermined (i.e. more rows than columns), the general inverses have different solutions and care is taken to denote them differently. Specifically, $\dagger$ denotes the minimum norm general inverse used for $J(t)$, and $+$ denotes the least-squares general inverse used for $L_5(t)$.

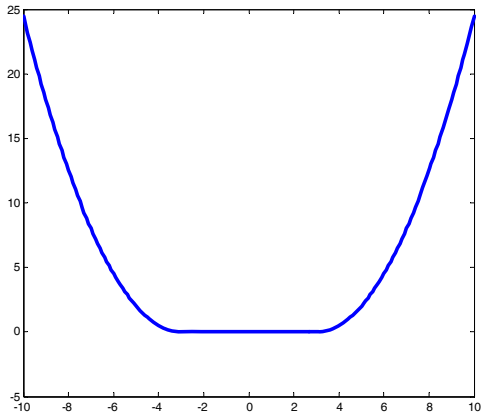

Fig. 4. An example of the window function for $p = 2$ and $r = 3$

the use of a smooth task function inspired by the p-times smooth bump function given in [23]. This smooth task function is given by

$$\phi_\delta = \sum_{i=1}^{3} \begin{cases} \frac{1}{p!} \int_{\underline{r}}^{\delta_i} (\tau - \underline{r})^p d\tau & \text{if } \tau > \underline{r} \\ \frac{1}{p!} \int_{-\underline{r}}^{\delta_i} (-\underline{r} - \tau)^p d\tau & \text{if } \tau < -\underline{r} \\ 0 & \text{else} \end{cases}$$

where $\delta = [\delta_1,\ \delta_2,\ \delta_3]^T$ is the coordinates of the nearest neighbor in the camera frame, $-\underline{r},\ \underline{r} \in \mathbb{R},\ -\underline{r} < \underline{r}$ define a window wherein $\phi_\delta = 0$, and $p \in \mathbb{R}$ gives the smoothness of the function ϕ_δ. Note that $\phi_\delta(t) = 0$ if $\delta_i \in [-\underline{r}, \underline{r}]\ \forall i$. Furthermore, $\phi_d(t)$ can be solved in closed form, evaluates to a positive definite function that can be differentiated p-times, and $\frac{\partial^p \phi_\delta}{\partial x^p} = \pm 1$ for $\phi_\delta \notin [-\underline{r}, \underline{r}]$. For example, for $p = 2$

$$\begin{aligned} \frac{1}{2} \int_A^{\delta_i} (t - \underline{r})^2 dt &= \frac{\delta_i^3}{6} - \frac{\underline{r}}{2}\delta_i^2 + \frac{\underline{r}^2}{2}\delta_i - \frac{\underline{r}^3}{6} \\ &= \frac{1}{6}(\delta_i - \underline{r})^3. \end{aligned}$$

These characteristics make the function useful for systems with high order dynamics. For the kinematic systems in this paper, $p = 1$ is sufficient, in which case the polynomial evaluates to

$$\int_A^{\delta_i} (t - \underline{r}) dt = \frac{1}{2}(\delta_i - \underline{r})^2$$

which is the familiar quadratic. This is illustrated for the case of $p = 2$ and $r = 3$ in Fig. 4.

The derivative of $\phi_\delta(t)$ can be expressed as

$$\begin{aligned} \dot{\phi}_\delta &= \sum_{i=1}^{3} \frac{\partial \phi_\delta}{\partial \delta_i} \dot{\delta}_i = J_\delta \dot{\delta} \\ &= J_\delta (v_c + \varepsilon_\delta), \end{aligned} \tag{20}$$

where $J_\delta(t) \in \mathbb{R}^{1\times 5}$ is a task function Jacobian, and $\varepsilon_\delta \in \mathbb{R}^5$ is a disturbance due to the velocity of the nearest neighbor. The Jacobian $J_\delta(t)$ is defined as

$$J_\delta = [\text{sgn}(\delta)^T, 0_2],$$

where $\text{sgn}(\cdot)$ is the vector signum function, and $0_2 \in \mathbb{R}^{1\times 2}$ is an all zero row-vector.

The objective is to regulate $\phi_\delta(t) \to 0$, under the assumption that the current velocity of an UAV's nearest neighbor is available through the connected communication network (i.e., ε_δ is known). Based on the task function time derivative in (20), a stabilizing camera velocity $v_{c\delta}$ can be designed as

$$v_{c\delta} = -\lambda_\delta J_\delta^\dagger \phi_\delta - \varepsilon_\delta, \tag{21}$$

where λ_δ is a positive scalar gain constant. Combining (20) and (21) yields the exponentially stable system

$$\dot{\phi}_\delta = -\lambda_\delta \phi_\delta.$$

B. Task Function for Mean of Image Points

Controlling the mean of feature point coordinates will help to ensure the feature points are centered around a position in the image plane. Let $\phi_m(t) \in \mathbb{R}^2$ denote a task function defined as the sample mean

$$\phi_m = \frac{1}{k}\sum_{i=1}^{k} m_i = \bar{m}. \tag{22}$$

The derivative of $\phi_m(t)$ can be expressed as

$$\begin{aligned}\dot{\phi}_m &= \frac{1}{k}\sum_{i=1}^{k} \frac{\partial \phi_m}{\partial m_i}\dot{m}_i = J_m \dot{m} \\ &= J_m\left(L v_c + \varepsilon\right),\end{aligned} \tag{23}$$

where $J_m(t) \in R^{2\times 2k}$ is a task function Jacobian defined as

$$J_m = \frac{1}{k}\left[I_2, \cdots, I_2\right],$$

where I_2 is the 2×2 identity matrix and is repeated k times, and $L(t)$, $v_c(t)$ and $\varepsilon(t)$ are introduced in (10).

As described in [16], to regulate the mean to a set point ϕ_{md}, PID control design can be used to developed as a stabilizing image velocity $\dot{m}_m$ as

$$\dot{m}_m = -J_m^{\dagger}\left(\lambda_m \phi_{me} + \lambda_{mi}\int_0^t \phi_{me} dt + \lambda_{md}\frac{d}{dt}\phi_{me}\right), \tag{24}$$

where $\phi_{me}(t) = \phi_m(t) - \phi_{md}$, and $\lambda_{mi}, \lambda_{md} \in R^+$ are constant gains. Substituting $\dot{m}_m(t)$ in (8) gives

$$v_c = L^+ \dot{m}_m.$$

If $k > 3$, $L^+(t)$ is underdetermined, meaning there are six degrees of freedom available in $v_c(t)$ to control more than six terms in $\dot{m}_m(t)$. This problem will be partially addressed by maximizing perceptibility in Section IV-D.

C. Task Function for Variance of Image Points

Regulating the variance of the feature point coordinates will regulate the spread of the feature points in the image. That is, regulating the variance will control how far the feature points drift from the mean value. To quantify the objective of regulating the variance, a sample variance task function $\phi_v(t) \in \mathbb{R}^2$ is defined as

$$\phi_v = \frac{1}{k}\sum_{i=1}^{k}\begin{bmatrix} (x_i - \bar{x})^2 \\ (y_i - \bar{y})^2 \end{bmatrix},$$

where $\bar{x}(t)$ and $\bar{y}(t)$ are the mean of all the x and y components of $m_i(t), i \in \{1 \ldots k\}$, respectively. The derivative of $\phi_v(t)$ is given by

$$\dot{\phi}_v = J_v \dot{m} \tag{25}$$

$$= J_v\left(L v_c + \varepsilon\right), \tag{26}$$

where $L(t)$, $v_c(t)$ and $\varepsilon(t)$ were given in (10), and $J_v(t) \in \mathbb{R}^{2\times 2k}$ is a task function Jacobian given by

$$J_v = \frac{2}{k}\begin{bmatrix} x_1 - \bar{x} & 0 \\ 0 & y_1 - \bar{y} \end{bmatrix}, \begin{bmatrix} x_2 - \bar{x} & 0 \\ 0 & y_2 - \bar{y} \end{bmatrix} \cdots, \begin{bmatrix} x_k - \bar{x} & 0 \\ 0 & y_k - \bar{y} \end{bmatrix}.$$

As described in [16], regulation of the variance to a set point ϕ_{vd} can be accomplished by using a PID feedback control to design a stabilizing image velocity $\dot{m}_v$ given by

$$\dot{m}_v = -J_v^{\dagger}\left(\lambda_v \phi_{ve} + \lambda_{vi}\int_0^t \phi_{ve} dt + \lambda_{vd}\frac{d}{dt}\phi_{ve}\right) \tag{27}$$

where $\phi_{ve}(t) = \phi_v(t) - \phi_{vd}$, and $\lambda_{vi}, \lambda_{vd} \in \mathbb{R}^+$ are constant gains.

D. Task Function for Perceptibility of Image Points

Sharma and Hutchinson presented the concept of motion perceptibility in [24]. Related to the concept of manipulability [25], perceptibility gives a measure of how well a camera can perceive the motion of objects in the FOV. Roughly speaking, if perceptibility is high, small object or camera velocities will result in notable feature velocities in the image plane (e.g., high optical flow). This is especially important if there are more than three feature points, as the available feature point velocities are constrained due to an overdetermined image Jacobian. Maintaining a high perceptibility helps to ensure a larger span of available feature point velocity vectors.

Perceptibility is a scalar function of the image Jacobian $L(t)$, defined as

$$w_p = \sqrt{\det(L^T L)} = \prod_{i=1}^{3} \sigma_i$$

where $\sigma_i(t) \in \mathbb{R}^+$ are the singular values of $L(t)$. Maximizing $w_v(t)$ is accomplished by maximizing each $\sigma_i(t)$. The matrix $L^T(t)L(t) \in \mathbb{R}^{3\times 3}$ is positive definite and symmetric, so the eigenvalues of $L^T(t)L(t)$ are equal to $\sigma_i^2(t)$. The trace of a matrix is equal to the sum of its eigenvalues. Therefore, the trace of $L^T(t)L(t)$ is related to the singular values by

$$Tr(L^T L_5) = \sum_{i=1}^{3} \sigma_i^2.$$

Increasing the trace of $L^T(t)L(t)$ will also increase the perceptibility.

The trace of $L^T(t)L(t)$ is given by

$$\begin{aligned} Tr(L^T L_5) &= \sum_{i=1}^{k} \Big(2x_i^2 y_i^2 + \left(y_i^2+1\right)^2 \\ &\quad + \left(x_i^2+1\right)^2 + x_i^2 + y_i^2\Big) \end{aligned}$$

A task function for perceptibility can be defined as

$$\phi_p = \frac{1}{\sum_{i=1}^{k} (x_i^2 + y_i^2)}.$$

Since it is desired to increase $Tr(L^T L)$, regulating $\phi_p(t)$ to 0 will result in increasing the trace. The time derivative of $\phi_p(t)$ is given by

$$\begin{aligned} \dot{\phi}_p &= -2\phi_p^2 \sum_{i=1}^{k} \begin{bmatrix} x_i & y_i \end{bmatrix} \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} \\ &= J_p(m)\dot{m} = J_p(m)\left(Lv_c + \varepsilon\right) \end{aligned} \tag{28}$$

where $L(t)$, $v_v(t)$ and $\varepsilon(t)$ were given in (10), and $J_p(m) \in \mathbb{R}^{1\times 2k}$ is the task function Jacobian for perceptibility. The matrix $J_p(m)$ is undefined only for the nongeneral case that $\forall i,\ m_i = 0$.

To regulate $\phi_p(t) \to 0$, the feature point velocity $\dot{m}_p(t) \in \mathbb{R}^{2k}$ is designed as

$$\dot{m}_p = -\lambda_p J_p^{\dagger} \phi_p, \tag{29}$$

where λ_p is a positive scalar gain constant. Combining (28) and (29) gives the closed-loop derivative of $\phi_p(t)$ as

$$\dot{\phi}_p = -\lambda_p \phi_p + \varepsilon_p,$$

where $\varepsilon_p(t)$ is an unknown disturbance caused by the velocity of feature points. Despite the presence of the disturbance $\varepsilon_p(t)$, the use of integral feedback is not recommended for the perceptibility term. This is due to the fact that $\phi_p(t)$ is unlikely to ever become zero, leading to possible integrator windup and related stability problems.

E. Cascaded Control Law

As stated in Section III, the control objective of this paper is to design a camera controller that maintains a set of feature points within the FOV of several independent assets. The controller is decentralized, and each UAV operates under an identical control law. In addition to maintaining the view of the targets, each camera must remain within a maximum distance from its nearest neighbor in order to maintain a communication network.

Maintaining the connectivity of the network is the primary task. Regulating the mean to the image center is chosen as the secondary task in order to keep the feature points centered in the FOV. Regulating the variance to a constant is chosen as the tertiary task to restrict the distance between the feature points and the image center. These two tasks ensure features

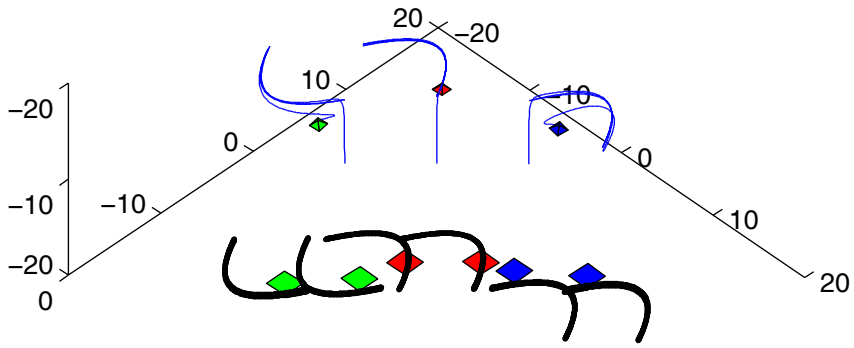

Fig. 5. Final 3D locations of the three cameras and the three sets of targets when inter-camera distance is not regulated.

remain in the FOV. High perceptibility will allow these two tasks to work more efficiently by ensuring larger available feature velocities at lower camera velocities. For this reason, increasing perceptibility is chosen as the quaternary task.

The designed feature velocities given in (21), (24), (27) and (29) are used in the null-space projection camera velocity (19) to give the overall controller for each camera as

$$v_{c5d} = v_{c\delta} + \left(I - J_\delta^\dagger J_\delta\right) L_5^+ \left[\dot{m}_m + \left(I - J_m^\dagger J_m\right)\left(\dot{m}_v + \left(I - J_v^\dagger J_v\right)\dot{m}_p\right) + L v_{cff}\right].$$

V. Simulation Results

Simulations of the proposed guidance and control system are presented. In these simulations, three camera-equiped UAV's observe six rigid, square objects. Each UAV is responsible for observing two of the square objects, and the corners of the squares give eight feature points for each camera to track. Each pair of objects moves with a sinusoidal linear velocity, independent of the other pairs. This simulation mimics the case of UAV's tracking sets of ground vehicles. Each camera has a resolution of 512×512. For each camera controller, the mean was regulated to $[256, 256]^T$, i.e., the mean of the points was regulated to the image center. The variance of the points was regulated to $[100^2,\ 100^2]^T$, i.e. a standard deviation of 100 pixels. The simulation was executed for 20 seconds at 30 frames per second.

The first simulation does not include any attempt to regulate the distance between the UAV's, i.e. the UAV's are free to move as necessary to track the targets, up to the constraints imposed by the rotorcraft model. Fig. 5 shows the final 3D positions of the three UAV's and the three pairs of targets, and the trajectory each took over time. The UAV's move away from each other to track their targets. Fig. 6 shows the views of the three cameras, including the final image of the tracked targets and the trajectory the corner points traced in the image over time. From the trajectory curves, it can be seen that the objects remain in view of each camera The dashed ellipse and square represent the final values of the variance and mean, while the solid ellipse and star represent the goal variance and mean. It can be seen that the targets remain in the field of view throughout the simulation.

The second simulation included the regulation of distance between the rotorcrafts as presented in Section IV-A. Fig. 7 shows the final 3D positions of the three targets and the three pairs of targets and their trajectories over time. While the targets all end at the same locations, the cameras end up closer to each other. Fig. 8 shows the views of the three cameras, including the final image of the tracked targets and the trajectory the corner points traced in the image over time. From the

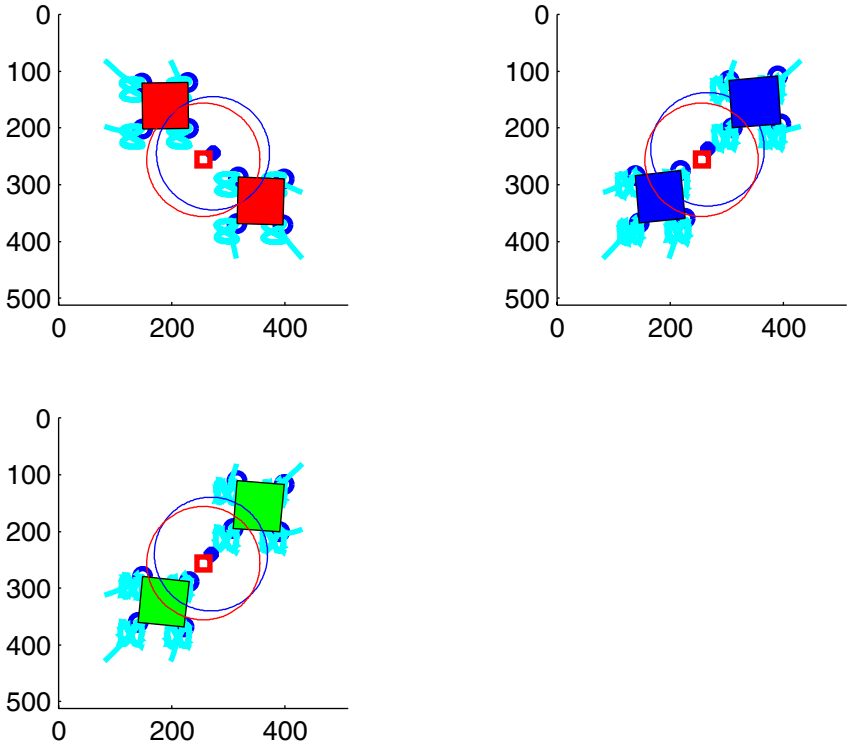

Fig. 6. The three camera views and image trajectories over time when inter-camera distance is not regulated.

trajectory curves, it can be seen that regulating the distance between UAV's makes the tracking task harder, and two of the cameras briefly lose sight of part of a target. The targets are brought back in to the field of view, but it is clear that the high-priority network maintenance task will prohibit the tracking task from working at peak performance.

Fig. 9 shows values of the task functions over time for the case when distance between UAV's is not regulated, and Fig. 10 shows values of the task functions when distance is regulated. As expected, when distance is regulated, the mean and variance tracking error grows larger, but the final distance between the cameras is smaller.

VI. Conclusions and Future Work

This paper presents the extension of a previous work in balancing two competing surveillance tasks, using multiple UAV's to monitor multiple, moving targets while maintaining a network communications between the UAV's. This problem is visualized as multiple air vehicles equipped with cameras tracking several groups of moving vehicles, with a network modeled as a proximity graph. Previous efforts did not include a vehicle model or velocity constraints on the camera, and this work introduces the use of rotorcrafts with gimbaled cameras. To achieve the tasks, a series of prioritized, underdetermined task functions were developed. Maintaining network connectivity can be ensured by regulating the distance of each UAV to its nearest neighbor. Targets can be kept in field of view through regulating mean and variance of the targets' features in the image. A third task function seeks to maximize motion perceptibility. There is no specific goal image or goal pose for the cameras, rather the underdetermined nature of the task functions allows the camera to move as necessary to regulate the task functions and keep objects in the FOV while maintaining network connectivity.

Simulations point towards positive, but incomplete results at this time. When distance between the crafts is not enforced, target tracking shows the successful extension our multi-target tracking method to include realistic vehicle constraints. However, when the distance between crafts is maintained, the targets are able to temporarily escape, as enforcing the maximum distance prevents the rotorcrafts from following the targets. Future work will focus on overcoming this by improving the guidance law. The prioritization scheme may be too rigid, and alternatives could be investigated. Another option is the use of time-varying desired values for the task functions, possibly incorporating motion planning, which may give more freedom of movement while meeting the relative distance requirements.

There are several other avenues of future work. Simulations are very promising, but a proper analysis can determine

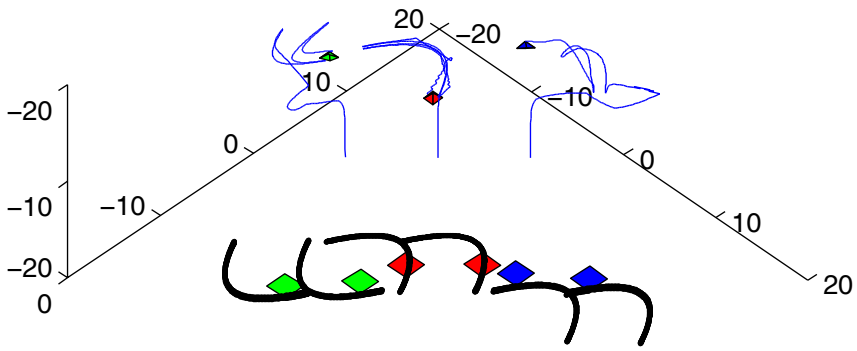

Fig. 7. Final 3D locations of the three cameras and the three sets of targets when inter-camera distance is not regulated.

whether, or under what conditions, stability can be achieved for all tasks. There are also numerous other task functions that could be used. For instance, it may be desirable to maintain a certain distance or orientation with respect to the tracked objects. The network was limited to three nodes, which allowed for the use of a simple distance regulation to maintain the network. This methodology must be extended to a larger, possibly unlimited, number of assets. This will require a better metric for measuring and regulating network connectivity. Finally, we hope to conduct flight experiments with the support of the Air Vehicles Directorate of AFRL, where recently an indoor flight facility for micro air vehicles has been developed.

REFERENCES

[1] Malis, E., Chaumette, F., and Boudet, S., "2-1/2D visual servoing," *IEEE Trans. Robot. Automat.*, Vol. 15, No. 2, 1999, pp. 238–250.
[2] Corke, P. and Hutchinson, S., "A New Partitioned Approach to Image-Based Visual Servo Control," *IEEE Trans. Robot. Automat.*, Vol. 17, No. 4, 2001, pp. 507–515.
[3] Mezouar, Y. and Chaumette, F., "Path Planning for Robust Image-Based Control," *IEEE Trans. Robot. Automat.*, Vol. 18, No. 4, 2002, pp. 534–549.
[4] Gans, N. and Hutchinson, S., "Stable Visual Servoing through Hybrid Switched-System Control," *IEEE Trans. Robotics*, Vol. 23, No. 3, 2007, pp. 530–540.
[5] Gaspar, J., Winters, N., and Santos-Victor, J., "Vision-Based Navigation and Environmental Representations with an Omnidirectional Camera," *IEEE Trans. Robotics*, Vol. 16, No. 6, 2000, pp. 890–898.
[6] Flandin, G., Chaumette, F., and Marchand, E., "Eye-in-hand/eye-to-hand cooperation for visual servoing," *Proc. IEEE International Conference on Robotics and Automation*, Vol. 3, 2000, pp. 2741–2746.
[7] Dixon, W., Zergeroglu, E., Fang, Y., and Dawson, D., "Object tracking by a robot manipulator: a robust cooperative visual servoing approach," *Proc. IEEE International Conference on Robotics and Automation*, Vol. 1, 2002, pp. 211–216.
[8] Gans, N. R., Shea, J. M., Barooah, P., and Dixon, W. E., "Ensuring network connectivity of UAV's performaing video reconnaissance," *Proc. IEEE Military Communications Conf.*, 2008.
[9] Shim, D. H., Kim, H. J., and Shankar, S., "Hierarchical control system synthesis for rotorcraft-based unmanned aerial vehicles," *Proc. AIAA Conf. on Guidance, Navigation, and Control*, 2000.
[10] Nakamura, Y., Hanafusa, H., and Yoshikawa, T., "Task-priority based redundancy control of robot manipulators," *Int. J. Robotics Research*, Vol. 9, 1987, pp. 3–15.
[11] Chiaverini, S., "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," Vol. 13, No. 3, 1997, pp. 398–410.
[12] Chaumette, F. and Hutchinson, S., "Visual servo control Part I: Basic approaches," *IEEE Robotics and Automation Mag.*, Vol. 13, No. 4, 2006, pp. 82–90.
[13] Bishop, B., Stilwell, D., Eng, S., Acad, U., and Annapolis, M., "On the application of redundant manipulator techniques to the control of platoons of autonomous vehicles," *Proc. IEEE Int. Conf. Control Applications*, 2001, pp. 823–828.

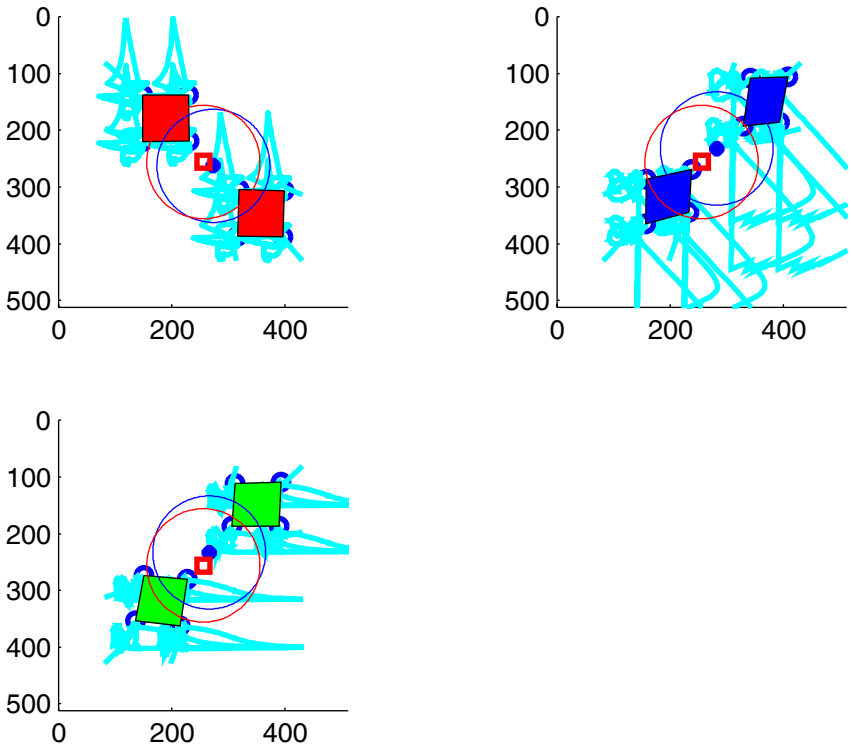

Fig. 8. The three camera views and image trajectories over time when inter-camera distance is regulated.

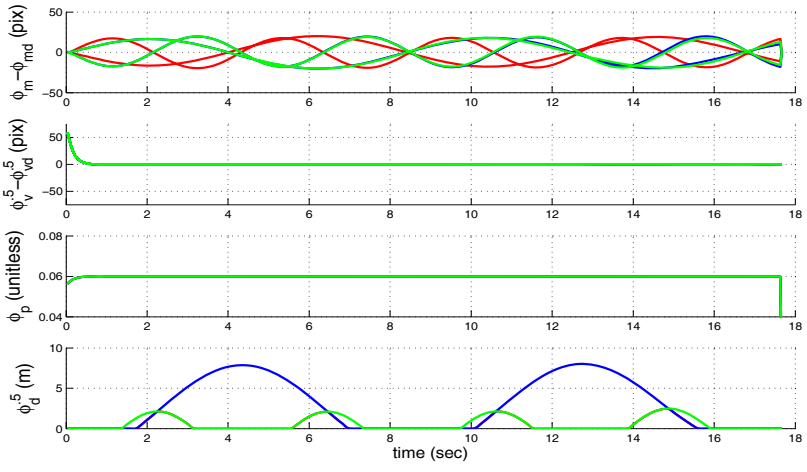

Fig. 9. Values of the task functions when inter-camera distance is not maintained

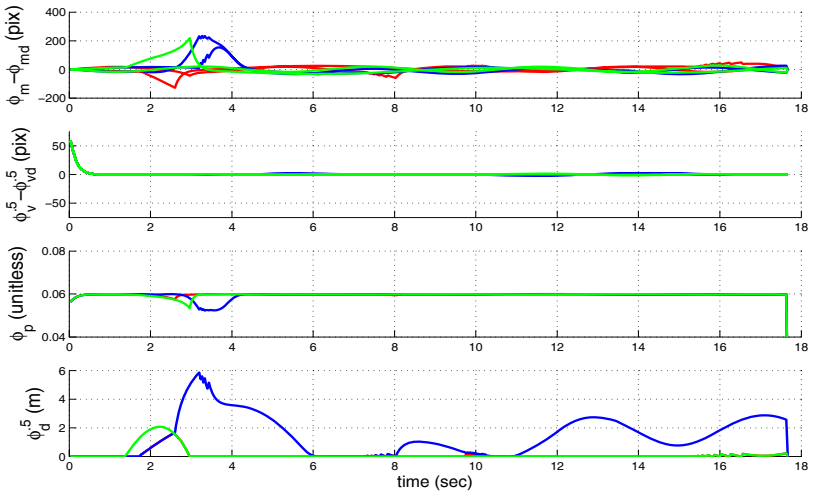

Fig. 10. Values of the task functions when inter-camera distance is maintained

[14] Antonelli, G. and Chiaverini, S., "Kinematic Control of Platoons of Autonomous Vehicles," Vol. 22, No. 22, Dec 2006, pp. 1285–1292.

[15] Espiau, B., Chaumette, F., and Rives, P., "A New Approach to Visual Servoing in Robotics," *IEEE Trans. Robot. Automat.*, Vol. 8, No. 3, June 1992, pp. 313–326.

[16] Gans, N. R., Hu, G., and Dixon, W. E., "Keeping Objects in the Field of View: An Underdetermined Task Function Approach to Visual Servoing," *Proc. IEEE Multi-Conf. Systems and Control*, 2008, pp. 432–437.

[17] Wagner, A. R. and Arkin, R. C., "Communication-sensitive multi-robot reconnaissance," *IEEE International Conference on Robotics and Automation (ICRA)*, 2004, pp. 2480–2487.

[18] Sweeney, J., Brunette, T. J., Yang, Y., and Grupen, R. A., "Coordinated teams of reactive mobile platforms," *International Conference on Robotics and Automation (ICRA)*, 2002, pp. 99–304.

[19] Pereira, G. A. S., Das, A. K., Kumar, V., and Campos, M. F. M., "Decentralized motion planning for multiple robots subject to sensing and communication constraints," *Second Multi-Robot Systems Workshop*, 2003, p. 267Ű278.

[20] Hsieh, M. A., Cowley, A., Kumar, V., and Taylor, C. J., "Maintaining network connectivity and performance in robot teams: Research Articles," *Journal of Field Robotics*, Vol. 25, No. 1-2, 2008, pp. 111–131.

[21] Notarstefano, G., Savla, K., Bullo, F., , and Jadbabaie, A., "Maintaining limited-range connectivity among second-order agents," *American Control Conference*, June 2006, pp. 2124–2129.

[22] Spanos, D. P. and Murray, R. M., "Motion planning with wireless network constraints," *American Control Conference*, June 2005, pp. 87–92.

[23] Do, K. D., "Formation Tracking Control of Unicycle-Type Mobile Robots With Limited Sensing Ranges," *IEEE Trans. Control Systems Technology*, Vol. 16, No. 3, 2008, pp. 527–538.

[24] Sharma, R. and Hutchinson, S., "Motion perceptibility and its application to active vision-based servo control," *IEEE Trans. Robot. Automat.*, Vol. 13, No. 4, 1997, pp. 607–617.

[25] Yoshikawa, T., "Manipulability of Robotic Mechanisms," *Int. J. of Robotics Research*, Vol. 4, No. 2, 1985, pp. 3–9.