Visual Servoing via Navigation Functions

Noah J. Cowan, *Member, IEEE*, Joel D. Weingarten, *Student Member, IEEE* and
Daniel E. Koditschek, *Senior Member, IEEE*

Abstract— **This paper presents a framework for visual servoing that guarantees convergence to a visible goal from almost every initially visible configuration while maintaining full view of all the feature points along the way. The method applies to first and second order fully actuated plant models. The solution entails three components: a model for the "occlusion-free" configurations; a change of coordinates from image to model coordinates; and a navigation function for the model space. We present three example applications of the framework, along with experimental validation of its practical efficacy.**

Keywords— **Visual servoing, navigation functions, dynamics, vision-based control, obstacle avoidance, occlusions, finite field-of-view (FOV).**

I. INTRODUCTION

INCREASINGLY, engineers employ computer vision to sense the projection of features of a rigid body as it moves in some scene. Closing a visual servo loop around image plane measurements requires a reliable machine vision system — incorporating image processing, feature extraction and feature correspondence — to supply a controller with the image plane coordinates of the features of the body. Traditionally, visual servoing algorithms impose motion upon a body's actuated configuration space variables so as to align its image-plane features with a previously stored desired view. When the mapping from configuration variables to the image-plane is one-to-one in some vicinity of the goal image, then traditional visual servoing generally results in (at least local) closed loop convergence to the desired position. Hutchinson *et. al.* [1] provide a general introduction and extensive bibliography to this approach.

Visual position regulation leaves few challenges: high performance, dynamic (albeit local) vision-based controllers which fit into the framework of linear control have existed for some years [2]. This complete characterization

N. J. Cowan is with the Department of Integrative Biology, University of California, Berkeley, CA 94720-3140 (e-mail: ncowan@eecs.berkeley.edu).

J. D. Weingarten and D. E. Koditschek are with the Advanced Technology Laboratory, University of Michigan, Ann Arbor, MI 48109-2110 (e-mail: jweingar@umich.edu; kod@umich.edu)

Reprint

is owed primarily to the local nature of such regulation tasks. However, the creation of a richer set of behaviors whose *global* (and hence highly nonlinear) properties are well characterized has remained a significant challenge to visual servoing.

The classical approach to visual servoing attempts to impose straight-line trajectories on image feature points. To illustrate, suppose a camera observes four feature points affixed to a rigid body controlled by a 6 degree of freedom (DOF) all revolute joint robot. The configuration space of the robot is parameterized by six joint angles, $q \in \mathbb{R}^6$, and the image space is defined in terms of the four image plane pairs, $y \in \mathbb{R}^8$. Locally, then, the camera's image may be modeled by a map $c : \mathbb{R}^6 \to \mathbb{R}^8$

$$y = c(q) = \begin{bmatrix} u_1 & v_1 & \cdots & u_4 & v_4 \end{bmatrix}^T,$$
$$\text{where} \quad \begin{bmatrix} u_i \\ v_i \end{bmatrix}, \quad i = 1, \ldots, 4,$$

are the image plane feature locations of the four features being observed. The traditional (kinematic) visual servoing law is then

$$\dot{q} = -J^\dagger(y - y^*), \quad \text{where} \quad J^\dagger = (J^T J)^{-1} J^T \in \mathbb{R}^{6 \times 8} \quad (1)$$

is a pseudo inverse of the Jacobian matrix, $[J]_{i,j} = \frac{\partial y_i}{\partial q_j}$.

Visual servoing systems based on the above approach have many well established merits, but may be improved in several key ways. First, they result in a *local* basin of attraction whose extent is poorly or not at all characterized. For example, the incursion of spurious (attracting) critical points may arise when $y - y^*$ aligns with the null space of $J^\dagger$ in (1). Consequently, the local basin of attraction around q^* may exclude seemingly reasonable initial conditions [3]. The second challenge to visual servoing involves the vulnerability to transient loss of features — either through self-occlusions or departure from the field of view (FOV). To the best of our knowledge, no prior work guarantees that these obstacles will be avoided (while ensuring convergence). However, as we will show, both of these challenges — local convergence and transient loss of features — can be overcome quite readily.

Another major problem is that most visual servoing algorithms do not specifically address dynamics. Of course, given the multitude of successful inverse dynamics based control strategies [4], trajectories generated from (1) (or any other kinematic controller) could be tracked very precisely with a high performance robot control system. However, such control techniques require precise parametric knowledge of the robot's kinematics and dynamics, the extra complexity of which seems superfluous given the sim-

ple end-point convergence objective of most visual servoing algorithms. Moreover, such heavy reliance on robot kinematic and dynamic parameters may be undesirable, especially when manipulating objects of imprecisely known size and weight. In other words, the specific reference trajectories generated by kinematic controllers are merely a means to an end; tracking those trajectories exactly may not be necessary. By contrast, our approach generates controllers capable of extremely high performance, which exhibit global convergence to the end-point goal, without the burden of precisely tracking a (clearly somewhat arbitrary) reference trajectory. Without prescribing specific reference trajectories, the proposed methodology nevertheless affords certain guarantees on the trajectories that result. For example, features are guaranteed to remain visible throughout transients (even in the presence of Newtonian dynamics).

A. Image-based navigation

In a naive attempt to improve (1), note that it may be conceived as the gradient of the potential function

$$\widetilde{\varphi}(y) = \frac{1}{2} \sum_{i=1}^{4} \|y_i - y_i^*\|^2, \qquad (2)$$

by letting $\dot{q} = -(J^T J)^{-1} D_q^T (\widetilde{\varphi} \circ c)$. Suppose the four features are coplanar, e.g. they are all on the same face of a polyhedral body. A self-occlusion occurs when the plane containing the features intersects the camera pinhole, causing the four points to project onto the same line on the image plane. To avoid this scenario, consider a naive improvement of (2) that avoids self-occlusions by "blowing them up", namely,

$$\widetilde{\varphi}(y) := \frac{\sum_{i=1}^{4} \|y_i - y_i^*\|^2}{\prod_{\{i,j,k\} \in \Gamma} \left| \det \begin{bmatrix} y_i & y_j & y_k \\ 1 & 1 & 1 \end{bmatrix} \right|^{1/2}}, \qquad (3)$$

where $\Gamma = \{\{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}\}$. The denominator will go to zero as the projected features become collinear, and thus the gradient will point away from the self-occlusion obstacle. However, as can be seen from Figure 1, even though self-occlusions are avoided, convergence is not necessarily achieved. Other such naive approaches may suffer similar peril.

Although the naive potential function approach above neither adequately addresses occlusions nor guarantees convergence, one suspects that some appropriately designed potential might overcome these serious limitations of traditional visual servoing. Indeed, we will show in this paper that the obstacles presented by self occlusion and a finite FOV can be obviated by addressing in a methodical fashion the relationships between the domain and range of the camera map, c, from which the obstacles arise.

Specifically, we introduce a framework for visual servoing yielding feedback controllers which are
1. *dynamic:* applicable to second order (Lagrangian) as well as first order (kinematic) actuation models;
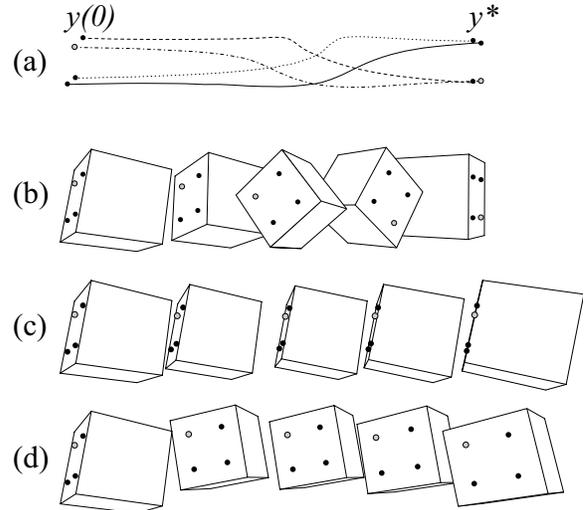

Fig. 1. Simulation results for three different artificial potential function based visual servo controllers using the same initial position, $y(0)$ and goal position y^*. **(a)** Image-plane feature trajectories resulting from the proposed navigation function (NF) in Appendix A. In this case, all of the features reach their respective goal. **(b,c,d)** Snap-shot of the body at five times during the trajectory for three different strategies; *(b)* based on the proposed NF; *(c)* based upon the potential function in equation (2) resulting in failure due to self occlusion; *(d)* based upon the potential function in equation (3) resulting in failure due to a spurious local minimum.

2. *global:* guaranteeing a basin of attraction encompassing almost every[1] initial configuration that presents full feature visibility;
3. *visibility-obstacle free:* avoiding configurations that lose features due to either self occlusion or departure from the camera FOV.

B. Related work

Many implementations in the literature offer strong anecdotal evidence that suggests convergence for visual servoing systems is robust with respect to large parametric uncertainty, though researchers rarely establish formally large basins of attraction for visual servoing systems (parametric uncertainty aside). Malis *et. al.* [5] and Taylor and Ostrowski [6] introduce visual servo controllers incorporating partial pose reconstruction to guarantee convergence even in the presence of large parametric uncertainty. Rizzi *et. al.* design a globally convergent nonlinear dynamical observer for a falling point mass viewed by a binocular camera pair [7], by mapping the observer feedback into the image plane. Similar machinery applies to binocular quasi-static visual servoing [8] to yield a controller with a large basin of attraction.

In addition to our preliminary results [9], [10], [11], there have been some recent efforts to address the FOV problem. For 6DOF visual servoing, Malis *et. al.* [5] guarantee that a single feature point remains within the FOV while, as noted above, guaranteeing convergence for a large basin

---

[1]That is, all but a set of measure zero.

of attraction. Morel *et. al.* [12] extend this idea by decoupling the image-plane motion of a cleverly chosen feature vector — a circle containing all the feature points — from the rotational motion of the camera; by keeping this conservative feature vector within the image plane, one is guaranteed that all feature points remain within the FOV (though self-occlusion avoidance is not guaranteed). Corke and Hutchinson [13] present a novel "partitioned" kinematic visual servo strategy for which simulations suggest a large basin of attraction while maintaining all features within the FOV boundary.

Two recent papers use path planning to address the FOV problem. Mezouar *et. al.* [14] adopt the approach of image-based path planning and local visual servoing along those paths to avoid mechanical limits and visibility obstacles. For nonholonomic plants, Zhang and Ostrowski [15] implement path planning to find kinematic trajectories that keep features within the FOV.

Zhang and Ostrowski [16] develop a dynamical controller for an Unpiloted Aerial Vehicle (UAV) with an on-board camera. Hamel and Mahony [17] present a dynamical visual servo for stabilizing a scale helicopter over a landing pad (implemented in simulation) which treats image measurements as unit vectors, thus preserving rotational symmetry that a flat image plane appears to break.

Potential field methods are employed for a wide variety of robot navigation problems (for a survey, see [18], Chapter 7). Traditionally, gradient fields are used to generate reference trajectories which are then tracked by a lower-level robot controller. Potential functions often encode obstacles as local maxima, or at least ensure that the gradient flow runs parallel to obstacles. The refinement to navigation functions (NF's), first articulated by Koditschek and Rimon [19], [20], [21], provides us machinery to "lift" potential functions to second order plants, while still ensuring obstacle avoidance with convergence guarantees, and no need for intermediate trajectories.

C. Organization

First we review NF-based robot control in Section II and then introduce our sensor model in Section III. The central contribution of the paper is found in Section IV where we propose a novel framework for dynamic, occlusion-free global visual servoing. We show how to apply this framework using three illustrative examples that provide insight into the specific geometries of some prototypical visual servoing systems. In Section V we present our experimental results for two of the example setups, and in Section VI we provide some concluding remarks. Refer to Table I for a list of symbols introduced in subsequent sections.

II. ROBOT CONTROL VIA NAVIGATION FUNCTIONS

For many visual servoing tasks, the objective is to bring a robot to rest at a desired configuration, q^*, known only through its image, $y^* = c(q^*)$. The task of moving to a goal while avoiding obstacles along the way can be achieved via a nonlinear generalization of proportional-derivative (PD) control deriving from Lord Kelvin's century old observation

TABLE I
LIST OF SYMBOLS AND SECTIONS IN WHICH THEY ARE INTRODUCED.

	Description	Section
$\mathcal{Q}$	configuration space	II
q	coordinates for $\mathcal{Q}$, $q \in \mathbb{R}^n$	II
$\mathcal{Y}$	image space	III
y	coordinates for $\mathcal{Y}$	III
$\mathbf{F}_b$	body frame	III
$\mathbf{F}_c$	camera frame	III
H	rigid transformation from $\mathbf{F}_b$ to $\mathbf{F}_c$	III
d	translation effected by H	III
R	rotation part of H, with columns r_i	III
π	perspective projection, $\pi : \mathbb{E}^3 \to \mathbb{R}^2$	III
$\mathcal{J}$	limited image plane, $\mathcal{J} = [-1,1]^2 \subset \mathbb{R}^2$	III
c	camera output model, $c : \mathcal{Q} \to \mathcal{Y}$	III
$\mathcal{O}$	obstacle set, $\mathcal{O} \subset \mathcal{Q}$	II, IV
$\mathcal{V}$	obstacle-free (visible) set, $\mathcal{V} = \mathcal{Q} - \mathcal{O}$	II, IV
$\mathcal{D}$	safe configurations, $\mathcal{D} \subset \mathcal{V}$	II, IV
$\mathcal{F}$	configurations facing the camera	IV
$\mathcal{W}$	configurations completely within FOV	IV
$\mathcal{I}$	camera image of safe domain, $\mathcal{I} = c(\mathcal{D})$	IV
$\mathcal{Z}$	simple model space, diffeomorphic to $\mathcal{D}$	IV
g	diffeomorphism to model space	IV
$\widetilde{\varphi}$	model space navigation function (NF)	IV
φ	configuration space NF, $\varphi = \widetilde{\varphi} \circ g \circ c$	II,IV

that total energy always decreases in damped mechanical systems [19]. Formally, this entails the introduction of a gradient vector field from a *navigation function* (hereafter, NF – a refined notion of an artificial potential function [20], [21]), together with damping to flush out any unwanted kinetic energy. As with any controller, tuning the parameters is still required for good performance. However, the NF method affords certain guarantees of reliability even in the absence of good tuning, as we will show. Moreover, a well constructed NF provides ample freedom to provide extremely high performance.

Of course, if a well-tuned robot control system is already in place, the NF may be used in the more traditional manner to generate trajectories via the gradient flow (in the case of a position controller) or field itself (in the case of a velocity controller). We will refer to this alternative as the "first-order" approach (i.e., based upon integrating the gradient dynamics directly) in distinction to the "second-order" approach.

A. Plant Model

Assume we have a holonomically constrained, fully actuated robot with known kinematics, affording a suitable set of local coordinates, $q \in \mathbb{R}^n$, and denote the $n-$dimensional free configuration space as $\mathcal{Q}$. The system dynamics

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + F_{\text{ext}}(q,\dot{q}),$$

may be found using Lagrange's equations (see, for example, [22], [23]) where F_{ext} are external forces (such as friction) which do not arise from Hamilton's variational principle

and τ are the input torques. We assume exact knowledge of the gravitational term G as well as the external forces F_{ext}.[2] Letting the input torque be $\tau = u - F_{\text{ext}} + G(q)$, where u is our control input, the plant equations are

$$\ddot{q} = M(q)^{-1}(u - C(q,\dot{q})\dot{q}). \qquad (4)$$

B. Task specification

The state space is constrained by the presence of forbidden configurations, the *obstacle set* $\mathcal{O} \subset \mathcal{Q}$. The *free space* is defined as the obstacle-free configuration space $\mathcal{V} = \mathcal{Q} - \mathcal{O}$, and we will be concerned only with a subset of *safe configurations* $\mathcal{D} \subseteq \mathcal{V}$ comprising a smooth compact connected manifold with boundary.[3] The positioning objective is described in terms of a *goal* $q^* \in \overset{\circ}{\mathcal{D}}$. The task is to drive q to q^* asymptotically subject to (4) by an appropriate choice of u while avoiding obstacles. Moreover, the basin of attraction $\mathcal{E}$ must include a dense subset of the zero velocity section of $T\mathcal{D}$, the tangent space of positions over $\mathcal{D}$, so that we may guarantee convergence from almost every initial zero velocity state $(q_0, 0)$ whose position component lies in $\mathcal{D}$. For obstacle avoidance, we require that the trajectories avoid crossing the boundary $\partial\mathcal{D}$, i.e. $q(t) \in \mathcal{D}$, for all $t \geq 0$.

C. First order gradient systems

Let $\mathcal{D}$ be an n-dimensional Riemannian manifold. The gradient in local coordinates is given by the vector of partial derivatives weighted by the inverse of the Riemannian metric. Suppose $q \in \mathbb{R}^n$ are local coordinates on $\mathcal{Q}$. Then, in local coordinates

$$\nabla\varphi(q) = M^{-1}(q)\,D_q^T\varphi(q),$$

where $(D_q\varphi)_i = \frac{\partial\varphi}{\partial q_i}$, and M is the local representation of the Riemannian metric. Gradient descent can now be achieved in local coordinates via

$$\dot{q} = -M^{-1}(q)\,D_q^T\varphi(q). \qquad (5)$$

A smooth scalar valued function whose Hessian matrix is non-singular at every critical point is called a *Morse* function [24]. Artificial potential controllers arising from Morse functions impose global steady state properties that are particularly easy to characterize, as summarized in the following proposition.

Proposition 1: (Koditschek, [19]) Let φ be a twice continuously differentiable Morse function on a compact Riemannian manifold, $\mathcal{D}$. Suppose that $\nabla\varphi$ is transverse and directed away from the interior of $\mathcal{D}$ on any boundary of that set. Then the negative gradient flow has the following properties:

1. $\mathcal{D}$ is a positive invariant set;
2. the positive limit set of $\mathcal{D}$ consists of the critical points of φ
3. there is a dense open set $\widetilde{\mathcal{D}} \subset \mathcal{D}$ whose limit set consists of the local minima of φ.

These first order dynamical convergence results do not apply for Lagrangian systems. We now briefly review machinery to "lift" the gradient vector field controller to one appropriate for second order plants of the kind introduced in (4).

D. Second order, damped gradient systems

Introducing a linear damping term, yields a nonlinear "PD" style feedback, in local coordinates,

$$u = -\alpha\, D_q\varphi(q)^T - K_d\,\dot{q}, \qquad (6)$$

that is appropriate for second order plants. Lord Kelvin's observation is now relevant and it follows that the total energy,

$$\eta = \alpha\varphi + \kappa \quad \text{where} \quad \kappa = \tfrac{1}{2}\dot{q}^T M(q)\dot{q}, \qquad (7)$$

is non-increasing.

Note that if the total initial energy exceeds the potential energy at some point on the boundary $\partial\mathcal{D}$, trajectories may intersect the boundary. Fortunately, further refining the class of potential functions will enable us to construct controllers for which the basin of attraction contains a dense subset of the zero velocity section of $T\mathcal{D}$. The following definition has been adapted from [25], [19].

Definition 1: Let $\mathcal{D}$ be a smooth compact connected manifold with boundary, and $q^* \in \overset{\circ}{\mathcal{D}}$ be a point in its interior. A Morse function, $\varphi \in C^2[\mathcal{D}, [0,1]]$ is called an *NF* if
1. φ takes its unique minimum at $\varphi(q^*) = 0$;
2. φ achieves its maximum of unity uniformly on the boundary, i.e. $\partial\mathcal{D} = \varphi^{-1}(1)$.

This notion, together with Lord Kelvin's observation, now yield the desired convergence result for the Lagrangian system (4).

Proposition 2: (Koditschek [19]) Given the system described by (4) subject to the control (6), almost every initial condition q_0 within the set

$$\mathcal{E} = \{(q,\dot{q}) \in T\mathcal{D} : \eta(q,\dot{q}) \leq \alpha\} \qquad (8)$$

converges to q^* asymptotically. Furthermore, transients remain within $\mathcal{D}$ such that $q(t) \in \mathcal{D}$ for all $t \geq 0$.

Proposition 2 generalizes the kinematic global convergence of Proposition 1. Note that for the second order system, $\mathcal{E}$ imposes a "speed limit" as well as a positional limit, since the total energy must be initially bounded [25].

E. Invariance under diffeomorphism

One last key ingredient in the mix of geometry and dynamics underlying the results we present involves the realization that an NF in one coordinate system is an NF in another, if the two coordinate systems are related by a

---

[2]The generalized PD approach to control, detailed below, will not require knowledge or computation of the mass-inertia matrix M or the Coriolis term C. In principle, some of the kinematic and dynamic parameters may be required for G and F_{ext}, though in our experiments we neglected these terms.

[3]Of course, in the general obstacle avoidance literature, the connectedness of the configuration space is very much at issue, but the issue does not arise in our present application.

diffeomorphism [19]. This affords the introduction of geometrically simple model spaces and their correspondingly simple model NF's.

III. SENSOR MODEL

To sense a robot's configuration with a camera, we seek a map, c, from the robot configuration space $\mathcal{Q}$ to an appropriate output space $\mathcal{Y}$. The exact form of c depends on several factors, for example the type and number of features being viewed, the number of cameras, the robot kinematics, and so forth. The examples presented in this paper are restricted to a monocular view of a fully actuated rigid body endowed with a set of point or "vector" features. However, the general methods presented in Section IV may in principle be applied to a wide variety of settings, including other feature types (e.g. conic sections or area features), so-called "eye-in-hand" servoing where the camera is moved by the robot relative to a fixed environment or systems incorporating binocular vision [26].

A Euclidean point $\mathbf{p} \in \mathbb{E}^3$ is represented in homogeneous coordinates by the array $[p_1, p_2, p_3, 1]^T$. Assume that there is a camera fixed coordinate frame, $\mathbf{F}_c$, such that the (x, y)-plane is parallel to the image plane and coincident with the optical center or "pinhole." A robot is assumed to move a rigid body with distinguishable features and body-fixed frame $\mathbf{F}_b$. A superscript preceding a point denotes the frame with respect to which it is written, e.g. $^b\mathbf{p}$ is expressed with respect to $\mathbf{F}_b$. We interpret the rigid transformation

$$H = \begin{bmatrix} R & d \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3), \quad \text{where}$$

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} \in \mathrm{SO}(3) \quad \text{and} \quad d \in \mathbb{R}^3,$$

as a change of coordinates from the body frame to the camera frame, *i.e.* $^c\mathbf{p} = H\,^b\mathbf{p}$. In general, H is a function of q through the robot kinematics, and so we write $H(q)$.

Because we are interested in potentially large displacements over the camera FOV, we employ a full perspective camera model. The methods in this paper require that the camera intrinsic parameters (e.g. focal length, pixel scale, and so forth) have been calibrated. Fortunately, we gain some robustness to parametric uncertainty since the feedback loop is closed around the image-plane measurement.[4] As such, we model a perspective projection camera, $\pi : \{\mathbf{p} \in \mathbb{E}^3 : p_3 > 0\} \to \mathbb{R}^2$, by

$$\pi(\mathbf{p}) := A\frac{1}{p_3}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + b, \quad \text{where} \quad A \in \mathrm{GL}(2), b \in \mathbb{R}^2. \quad (9)$$

Typically, camera images are rectangular, so we choose A and b to map the image plane pixel values (for example $[0, 640] \times [0, 480]$) to the box $\mathcal{J} = [-1, 1] \times [-1, 1]$. The algorithms presented below are designed to keep the feature projections within $\mathcal{J}$, to account for the finite FOV.

As an example, suppose we have a set of point features, $\{\mathbf{p}_i\}_{i=1}^N$. Then the composition of the camera model with the kinematics as applied to all of the features generates the camera map $c : \mathcal{Q} \to \mathcal{Y}$

$$y = c(q) := \begin{bmatrix} \pi(^c\mathbf{p}_1) \\ \vdots \\ \pi(^c\mathbf{p}_N) \end{bmatrix}, \quad \text{where} \quad \begin{matrix} ^c\mathbf{p}_i = H(q)^b\mathbf{p}_i, \\ i = 1, \dots, N. \end{matrix} \quad (10)$$

The map c is parameterized by the (constant) feature locations in body coordinates, the robot kinematic parameters and the camera parameters. The output space in this example is simply $\mathcal{Y} = \mathbb{R}^{2N}$.

IV. NAVIGATION FUNCTION BASED VISUAL SERVOING

We wish to create visual servoing algorithms that offer high performance, and enjoy global guarantees respecting both convergence and the avoidance of FOV boundaries and self-occlusions. To achieve our objective we compute the *visible set* for a particular problem. This is the set of all configurations $\mathcal{V} = \mathcal{Q} - \mathcal{O}$ in which all features are visible to the camera and on which c is well defined. We then design a safe, possibly conservative, subset $\mathcal{D} \subseteq \mathcal{V}$ to provide additional safety with respect to visibility obstacles. The *image space* is defined $\mathcal{I} = c(\mathcal{D}) \subset \mathcal{Y}$, and the camera map must be a diffeomorphism $c : \mathcal{D} \approx \mathcal{I}$. For each problem, $\mathcal{D}$ is analyzed to construct a model space $\mathcal{Z}$ and a diffeomorphism $g : \mathcal{I} \approx \mathcal{Z}$. Given a configuration space goal $q^* \in \overset{\circ}{\mathcal{D}}$ we define the *goal image* $y^* = c(q^*)$.

We propose a new framework for visual servoing that incorporates three ingredients:
1. a *model space*, $\mathcal{Z}$, for the "safe" configurations, $\mathcal{D}$;
2. an *NF*, $\widetilde{\varphi} : \mathcal{Z} \to [0, 1]$, for the model space;
3. a *diffeomorphism*, $g : \mathcal{I} \to \mathcal{Z}$, from the image space to the model space.

Recalling that NF's are invariant with respect to diffeomorphism, an NF with respect to the configuration variables is given by $\varphi(q) = \widetilde{\varphi} \circ g \circ c(q)$ with gradient given by

$$D\varphi^T = Dc^T\, Dg^T\, D\widetilde{\varphi}^T. \quad (11)$$

Hence, the three ingredients above are assembled with the feedback control strategy (6). With this approach, almost all initial configurations[5] within $\mathcal{D}$ *dynamically converge* to the goal while ensuring *occlusion-free* transients.

By recourse to the general framework outlined above we develop controllers for several specific configurations of a robot and monocular camera in the subsections that follow. Section IV-A considers a planar body on which three collinear feature points project to a planar camera. In Section IV-B, we present a solution to the problem of visually servoing a high performance custom robot. Finally in Section IV-C we present a solution for visually servoing a 6DOF body. Interestingly, the very different visual servoing problems of Sections IV-B and IV-C share a common

---

[4]As mentioned in Section I, prior contributions establish this formally in some settings [27], [6], and we believe (though have not formally shown) that similar results are possible here.

[5]That is, all but a set of measure zero. Due to topological constraints, smooth global visual servos are technically impossible in some cases [26].

model space $\mathcal{Z} = [-1,1]^n \times \mathrm{S}^1$ for $n = 2$ and $n = 5$ respectively, so Appendix A presents a new, highly tunable NF for the more general model space $\mathcal{Z} = [-1,1]^n \times \mathrm{T}^m$ for all n and m in $\mathbb{N}$.

A. Example 1: Planar rigid body servoing

Consider the problem of visually servoing a planar rigid body, viewed by a planar pinhole camera, as depicted in Figure 2, first presented in [9]. The intuition for this example is very simple: the projection of three collinear feature points onto the image plane provides a set of generalized coordinates for the three degrees of rigid body freedom. To maintain visibility, the three feature point projections must remain in order left-to-right (so that the body faces the camera), and must remain within a bounded interval on the image plane (due to the finite FOV).

Suppose we have a planar rigid body, $\mathcal{Q} = \mathrm{SE}(2)$, with three distinguishable collinear feature points, $\{\mathbf{p}_i\}_{i=1}^3$. We use for local coordinates on $\mathcal{Q}$ the rigid body position and orientation $q = [d_1, d_2, \theta]^T$, i.e.

$$H = \begin{bmatrix} R & d \\ 0^T & 1 \end{bmatrix}, \quad \text{where} \quad R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

We conveniently collocate the x-axis of the body frame with the edge containing the feature points so that in body coordinates

$$^b\mathbf{p}_i = \begin{bmatrix} l_i & 0 & 1 \end{bmatrix}^T, \quad \text{where} \quad l_1 < l_2 < l_3,$$

and in camera coordinates

$$^c\mathbf{p}_i = H\,^b\mathbf{p}_i = \begin{bmatrix} d_1 + l_i\cos\theta & d_2 + l_i\sin\theta & 1 \end{bmatrix}^T.$$

The body frame y-axis is oriented "into" the body, as depicted in Figure 2.

For this simplified example, we reduce the calibrated camera of (9) to $\pi : \left\{ \mathbf{p} \in \mathbb{E}^2 : p_2 > 0 \right\} \to \mathbb{R}$

$$\pi(\mathbf{p}) := \frac{p_1}{p_2} \tag{12}$$

where $\mathbf{p}$ is a point expressed with respect to the camera frame. The bounded image plane reduces to $\mathcal{J} = [-1, 1]$.

For all the feature points to be visible, the rigid body must be facing the camera, and all the features must be in the FOV. We define the workspace as those configurations for which all features are within the FOV, namely

$$\mathcal{W} = \{H \in \mathrm{SE}(2) : d_2 > 0, \pi(H\,^b\mathbf{p}_i) \in \mathcal{J},\ i = 1, 2, 3\}.$$

Those configurations that face the camera are given by

$$\mathcal{F} = \{H \in \mathrm{SE}(2) : \mathbf{v}(H) > 0\},$$
$$\text{where} \quad \mathbf{v}(H) = (d_2\cos\theta - d_1\sin\theta).$$

and the visible set is the intersection $\mathcal{V} = \mathcal{F} \cap \mathcal{W} \subset \mathrm{SE}(2)$.

Knowing which configurations can be "seen" allows us to define the camera output map, $c : \mathcal{V} \to \mathbb{R}^3$, as the projection of the three feature points on the body, namely

$$y = c(H) := \begin{bmatrix} \pi(H\,^b\mathbf{p}_1) & \pi(H\,^b\mathbf{p}_2) & \pi(H\,^b\mathbf{p}_3) \end{bmatrix}^T. \tag{13}$$

(In an abuse of notation, we will write $c(q)$ when using local coordinates, $q = [d_1, d_2, \theta]^T$.)

The map c is a diffeomorphism from $\mathcal{V}$ to its image

$$\mathcal{I}' = \left\{ y \in \mathbb{R}^3 : -1 \le y_1 < y_2 < y_3 \le 1 \right\},$$

the proof of which follows from the fact that c is a homeomorphism from $\mathcal{V}$ to $\mathcal{I}'$, and at each point in $\mathcal{V}$, c is a local diffeomorphism. To verify the first fact, note that for $H \in \mathcal{V}$, the points (y_1, y_2, y_3) and (l_1, l_2, l_3) are related by the unique homography,[6]

$$\alpha_i \begin{bmatrix} y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & d_1 \\ \sin\theta & d_2 \end{bmatrix} \begin{bmatrix} l_i \\ 1 \end{bmatrix}, \quad i = 1, 2, 3,$$

for some $\alpha_i \ne 0$, which can be used to construct the unique inverse of c for each $y \in \mathcal{I}'$. To show that c is a local diffeomorphism, note that the pinhole camera π is differentiable everywhere in front of the camera, and hence c is differentiable on $\mathcal{V}$. Moreover, direct computation reveals that

$$\begin{aligned} |Dc(q)| &= \frac{(l_1 - l_2)(l_2 - l_3)(l_3 - l_1)(d_2\cos\theta - d_1\sin\theta)}{(d_2 + l_1\sin\theta)^2(d_2 + l_2\sin\theta)^2(d_2 + l_3\sin\theta)^2} \\ &= \mathbf{v}(H)\frac{(l_1 - l_2)(l_2 - l_3)(l_3 - l_1)}{(d_2 + l_1\sin\theta)^2(d_2 + l_2\sin\theta)^2(d_2 + l_3\sin\theta)^2} \end{aligned}$$

which is different from zero at every point in $\mathcal{V}$, and hence c is a local diffeomorphism at every point in $\mathcal{V}$ (inverse function theorem).

We seek a compact manifold with boundary on which to impose an NF. Note that $\mathcal{V}$ is not a closed set. However it is practical to impose impose a "collar" around the points on the image plane by enforcing the requirement that they maintain a minimum distance from one another, as well as maintaining their distance from the edge of the image plane. This will also have the practical benefit of keeping the body within a certain distance of the camera. Letting $y_0 = -1$ and $y_4 = +1$ denote the left and right edge of the image plane, we define the compact set

$$\mathcal{I} = \left\{ y \in \mathbb{R}^3 : y_{i+1} - y_i \ge \rho,\ i = 0, \ldots, 3 \right\},$$

where $\rho > 0$ denotes the (arbitrarily) small "safety" collar around each feature point. The set $\mathcal{I} \subset \mathcal{I}'$ is compact, and hence we define the "safe" compact domain

$$\mathcal{D} = c^{-1}(\mathcal{I}) \subset \mathcal{V}.$$

As required by our general framework, the camera map provides a change of coordinates $c : \mathcal{D} \to \mathcal{I}$. By choosing $\mathcal{Z} = \mathcal{I}$, the mapping g of our formal methodology is the identity mapping. For this model space we refine a class of NF's designed in a separate context [29]. For $k > (2N + 3)/2$, the objective function

$$\overline{\varphi}(z) = \frac{\|z - z^*\|^{2k}}{\prod_{i=0}^{N}(z_{i+1} - z_i)^2 - \rho^2} \tag{14}$$

---

[6] A homography or projective transformation, $Z \in \mathrm{PL}(1)$ is uniquely determined by the correspondence of three distinct points [28].
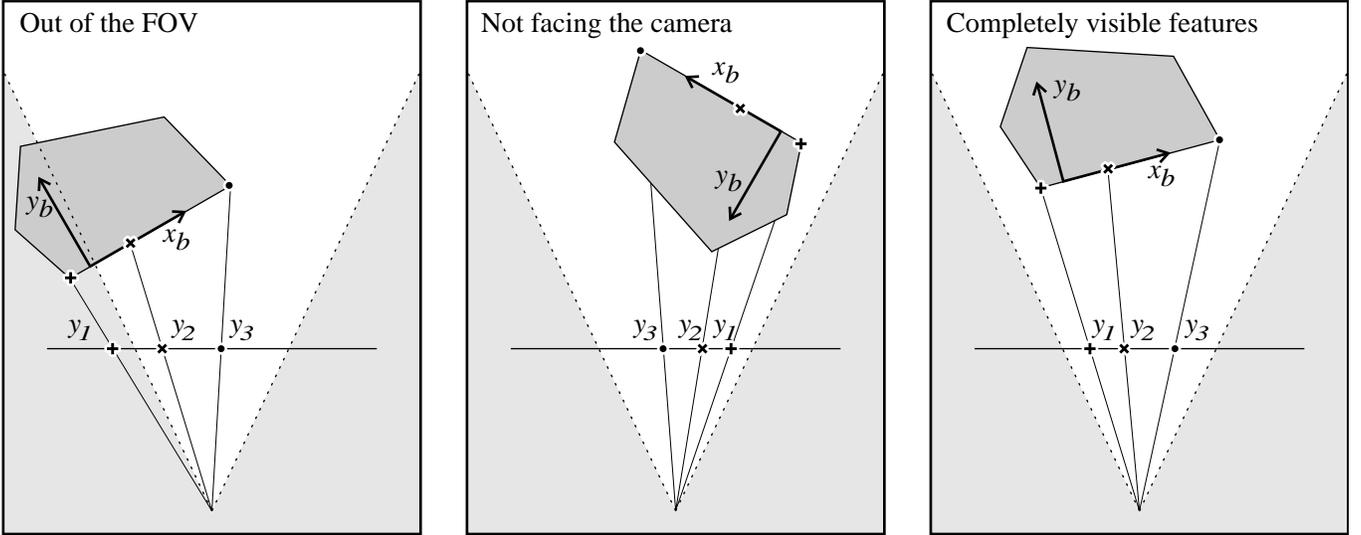
Fig. 2. The setup for a planar pinhole camera and planar rigid body with collinear feature points showing three typical configurations of a rigid body with respect to the camera, to illustrate the notion of visibility. **Left:** The features face the camera, but the leftmost point lies out of view. **Center:** Although within the camera workspace, the body occludes the features. **Right:** The features are all visible.

is convex on

$$\mathcal{Z} = \{z \in \mathbb{R}^N : z_{i+1} - z_i \geq \rho, \, i = 0 \ldots, N\},$$

where $z_0 = -1$, $z_{N+1} = +1$ are constants. Moreover,

$$\widetilde{\varphi} := \frac{\overline{\varphi}^{1/k}}{(1 + \overline{\varphi})^{1/k}} \qquad (15)$$

is an NF on $\mathcal{Z}$ [29].

For planar servoing, $N = 3$ and so we require $k > 9/2$ to ensure that $\overline{\varphi}$ is convex. As such, $\widetilde{\varphi}$ is an NF on $\mathcal{I}$ and $\varphi := \widetilde{\varphi} \circ g \circ c$ is an NF on $\mathcal{D}$.

B. Example 2: Buehgler arm servoing

The Buehgler arm, a direct drive robot built previously in our laboratory [7], [4], has allowed us to explore experimentally a dynamical implementation of our visual servoing methodology. Moreover, it turns out that the construction of a good NF for this problem will, with appropriate generalization (Appendix A), apply to 6DOF visual servoing (Section IV-C).

The Buehgler arm, depicted in Figure 3, has three actuated revolute degrees of freedom parameterized in local coordinates by angles $q = [\, q_1, \, q_2, \, q_3 \,]^T$ and its configuration space is $\mathcal{Q} = \mathrm{S}^1 \times \mathrm{S}^1 \times \mathrm{S}^1 = \mathrm{T}^3$. We affix a body frame to the "tip" of the paddle, as depicted in Figure 3. The Buehgler kinematics, detailed in [7], parameterize the rigid transformation, H, from the body frame to the camera frame.

We affixed a "pointer" to the tip of the arm, which we model as the position and unit orientation of a vector in space, namely[7] $({}^b p, {}^b w) \in T_1 \mathbb{E}^3 \approx \mathbb{E}^3 \times \mathrm{S}^2$. The feature point is centered at the body frame origin, ${}^b p = [\, 0, \, 0, \, 0, \, 1 \,]^T$, and "pointing" along the body y−axis, ${}^b w = [\, 0, \, 1, \, 0 \,]^T$.

[7]$T_1 \mathcal{X} \subset T\mathcal{X}$ is the unit tangent bundle of $\mathcal{X}$ [24].
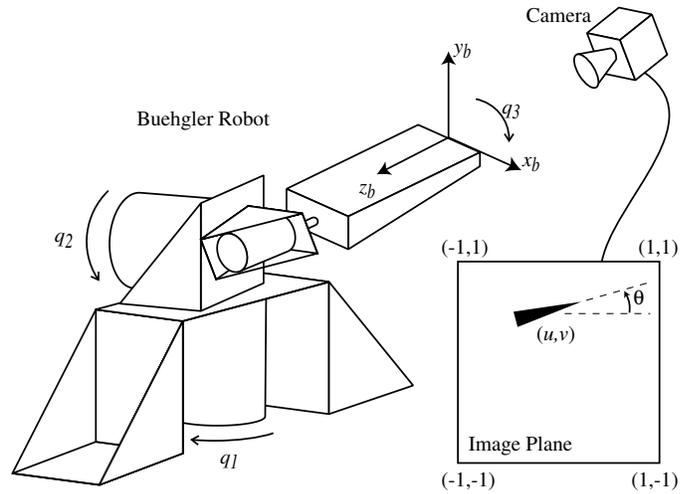

Fig. 3. The Buehgler arm [4], [7] has been modified with an "arrow" feature on the tip of the arm, which is observed by a perspective projection camera. The body frame is coincident with the tip of the arm, and the z-axis is pointing into the arm. The camera image is segmented to extract the position, (u, v), and orientation, θ, of the arrow. Roughly speaking, the waist motor and shoulder motors, (q_1, q_2), move the image feature left-and-right and up-and-down, respectively, and the wrist, q_3, rotates the image feature. The camera is positioned so the feature may reach the entire image plane.

The pointer moves with respect to the camera via $h : \mathcal{Q} \to T_1 \mathbb{E}^3$

$$h(q) := \begin{bmatrix} H(q){}^b p \\ R(q){}^b w \end{bmatrix} =: \begin{bmatrix} p(q) \\ w(q) \end{bmatrix}.$$

Note that in the camera frame, $p(q) = [\, d(q)^T, \, 1 \,]^T$.

A camera (9) is positioned to view the robot end effector as depicted in Figure 3. The workspace is the set of configurations that keep the projected feature point within $\mathcal{J}$,

namely

$$\mathcal{W} = \{q \in \mathcal{Q} : d_3(q) > 0, \pi(p(q)) \in \mathcal{J}\}, \qquad (16)$$

which may have two connected components. Considering only those configurations facing the camera,

$$\mathcal{F} = \left\{q \in \mathcal{Q} : d(q)^T r_3(q) > 0\right\}, \qquad (17)$$

we show in [30] that, for reasonable camera positions, the visible set $\mathcal{V} = \mathcal{F} \cap \mathcal{W}$ has a single connected component.

The camera map is given by the projection of the feature, and its orientation on the image plane, namely

$$y = \begin{bmatrix} u \\ v \\ \theta \end{bmatrix} = c(q) = \begin{bmatrix} \pi(p(q)) \\ \angle \{[D_p\pi(p(q))]\ w(q)\} \end{bmatrix}, \qquad (18)$$

where $\angle$ normalizes the vector on the image plane. Hence, the function c yields the position and orientation of our projected feature on the image plane, *i.e.* $c : \mathcal{V} \to \mathcal{Y}$, where $\mathcal{Y} = T_1\mathbb{R}^2 \approx \mathbb{R}^2 \times S^1$.

For reasonable camera positions relative to the robot, we show in [30] that c is a diffeomorphism $c : \mathcal{V} \approx T_1\mathcal{J}$. In this case $\mathcal{V}$ is compact, so we set $\mathcal{D} = \mathcal{V}$ and $\mathcal{I} = T_1\mathcal{J}$, which is diffeomorphic to $\mathcal{Z} = [-1, 1]^2 \times S^1$. (Additional safety may be added with respect to the FOV quite readily if desired.) The change of coordinates, $g : \mathcal{I} \approx \mathcal{Z}$ is just the natural identification of the two spaces. Appendix A defines a new NF, $\widetilde{\varphi}$, for this space (see Equations (24) and (25)).

C. Example 3: 6DOF rigid body servoing

In this example, we consider a free convex polyhedral rigid body, with configuration space $\mathcal{Q} = \mathrm{SE}(3)$, and let $\{\mathbf{p}_i\}_{i=1}^N$, $\mathbf{p}_i \in \mathbb{E}^3$, be a set of coplanar distinguishable points on a face of the body. For convenience, we place the body frame $\mathbf{F}_b$ so that the x, y-plane contains our coplanar features, the z-axis points "into" the body, and the body origin coincides with the center of the smallest sphere containing all the feature points. Denote the radius of this sphere ρ. Our camera map, c, is now given by

$$y = \begin{bmatrix} u_1 & v_1 & \cdots & u_N & v_N \end{bmatrix}^T = c(H),$$
$$\text{where} \quad \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \pi(H\ ^b\mathbf{p}_i), \quad i = 1, \ldots, N.$$

It is now routine to define the visible set, $\mathcal{V}$, as in previous examples. Denote the set of configurations facing the camera

$$\mathcal{F} = \{H \in \mathrm{SE}(3) : \mathbf{v}(H) > 0\} \quad \text{where} \quad \mathbf{v}(H) = r_3^T d$$

and denote the workspace

$$\mathcal{W} = \{H \in \mathrm{SE}(3) : d_3 > 0, \pi(H\ ^b\mathbf{p}_i) \in \mathcal{J}, i = 1, \ldots, N\}$$

The visible set, as before, is given by $\mathcal{V} = \mathcal{W} \cap \mathcal{F}$.

Construction of a suitable image-based coordinate system for this problem represents work in progress. Therefore, we employ 3D or task-space visual servoing for this

scenario, wherein we use feature-point locations on the image plane to estimate H and H^* from their respective images. For $N \geq 4$, this can be done quite readily by noting that in body coordinates

$$^b\mathbf{p}_i = \begin{bmatrix} l_{1i} & l_{2i} & 0 & 1 \end{bmatrix}^T,$$

and hence the feature locations and body coordinates are related by the homography

$$\alpha_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} A & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & d \end{bmatrix} \begin{bmatrix} l_{1i} \\ l_{2i} \\ 1 \end{bmatrix}, \qquad (19)$$
$$\text{for some } \alpha_i \neq 0, \quad i = 1, \ldots, N,$$

where A and b are the camera calibration parameters given in (9). Note that the matrix $[r_1, r_2, d]$ is nonsingular if and only if $\mathbf{v}(H) > 0$, which our visual servoing algorithm will ensure. Hence, given y, y^* and $\{\mathbf{p}_i\}_{i=1}^N$, then for $H, H^* \in \mathcal{V}$, one may solve (19) for H and H^*.

Task-based visual servoing presents only a minor twist to our overall program. As before, we will construct a model space for visible configurations and define an NF on that model space. However, in this case, our diffeomorphism g maps from the configuration space directly to the model space, without the camera map intervening. To construct g, we adopt a very specific parameterization of the rigid body configuration to aid in our construction of the safe domain. Denote the translation of the rigid body origin by d and Euler angles by (ϕ, ψ, θ). The Euler angles have been selected to parameterize the motion in such a way that for $\phi = \psi = 0$, the z-axis of the body is parallel to the translation, *i.e.*

$$H(d, \phi, \psi, \theta) := H_1(d)H_2(\phi, \psi, \theta), \quad \text{where} \qquad (20)$$

$$H_1(d) := \begin{bmatrix} \frac{e_2 \times d}{\|e_2 \times d\|} & \frac{d \times (e_2 \times d)}{\|d \times (e_2 \times d)\|} & \frac{d}{\|d\|} & d \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (21)$$

$$H_2(\phi, \psi, \theta) := \begin{bmatrix} R_x(\phi)R_y(\psi)R_z(\theta) & 0 \\ 0^T & 1 \end{bmatrix}, \qquad (22)$$

and where R_x, R_y and R_z are the standard x, y, z Euler angle rotation matrices (see, for example, Craig [22]).

Consider a "safe" subset of configurations that face the camera

$$\mathcal{F}' = \{H \in \mathcal{F} : -\vartheta \leq \phi, \psi \leq \vartheta\},$$

where H is parameterized by (20-22) and $\vartheta < \pi/2$. Also, consider a "safe" workspace

$$\mathcal{W}' = \{H \in \mathrm{SE}(3) : d_3 \in [\delta_{\min}, \delta_{\max}], \pi(\mathcal{B}_\rho(d)) \subset \mathcal{J}\},$$

where $\mathcal{B}_\rho(d)$ denotes a ball of radius ρ around the point d. In other words $\mathcal{W}'$ is the set of all configurations which keep a ball of radius ρ completely within the FOV and impose upper and lower limits, $\delta_{\min}$ and $\delta_{\max}$, on the distance of the body from the camera. Recall that the ball $\mathcal{B}_\rho(d)$ contains all the feature points, and hence $\mathcal{W}' \subset \mathcal{W}$. We now define the "safe" domain $\mathcal{D} = \mathcal{W}' \cap \mathcal{F}' \subset \mathcal{V}$. For
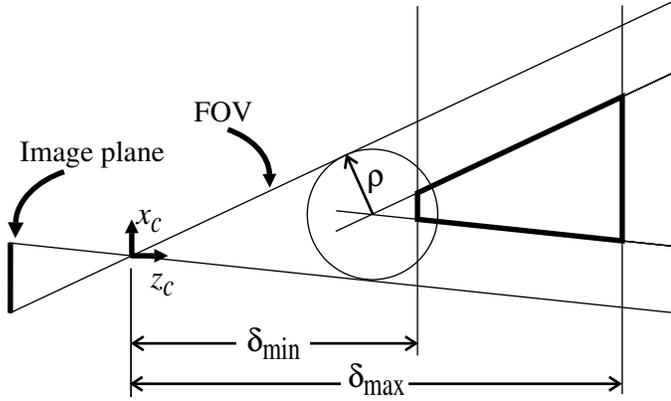
Fig. 4. In the safe domain, translations are constrained to move in a trapezoidal box (an orthographic projection of which is shown by the thick lines), so that the body may not move too far from or close to the camera, and so that the smallest sphere containing all the feature points remains within the FOV. The minimum and maximum distances, $\delta_{\min}$ and $\delta_{\max}$, from the camera are free design parameters.

$\rho < \delta_{\min} < \delta_{\max}$ the translations are constrained as seen in Figure 4.

Within $\mathcal{D}$, the body translation is confined to move within a diffeomorphic copy of $[-1, 1]^3$. Recall that the body coordinate frame $\mathbf{F}_b$ is attached such that the z-axis is orthogonal to the face (facing into the body) and the (x, y)-plane contains the feature points. Consider the fact that SO(3) is an SO(2) bundle over S^2, and identify the orientation of the z-axis with the base point in S^2. The requirement that the body faces the camera is a constraint on the z-axis, namely that it always has a positive projection of onto the line-of-site. This yields an open hemisphere; *i.e.* a diffeomorphic copy of $\mathbb{R}^2$. An SO(2) bundle over $\mathbb{R}^2$ is diffeomorphic to $\mathbb{R}^2 \times \mathrm{SO}(2)$. Constraining the Euler angles further $-\vartheta \leq \phi, \psi \leq \vartheta$ yields a diffeomorphic copy of $[-1, 1]^2 \times \mathrm{SO}(2)$. Therefore

$$\mathcal{D} \approx \underbrace{[-1, 1]^2 \times \mathrm{SO}(2)}_{\text{rotations}} \times \underbrace{[-1, 1]^3}_{\text{translations}} \approx [-1, 1]^5 \times \mathrm{S}^1.$$

The above coordinates (20-22) may now be used as local coordinates on $\mathcal{D}$, with no singularities. Hence, constructing g is now straight-forward – for the three translational DOF's, simply map the trapezoidal box depicted in Figure 4 to the box $[-1, 1]^3$, and map the two constrained Euler angles to $[-1, 1]^2$. Hence, $g : \mathcal{D} \to [-1, 1]^5 \times \mathrm{S}^1$ (the details can be found in [31], [30]). As for the Buehgler, we employ the more general model space and NF described in Appendix A, and hence $\varphi = \widetilde{\varphi} \circ g$, where $\widetilde{\varphi}$ is given by (25) for $n = 5$, $m = 1$, is an NF for spatial visual servoing.

V. Empirical Validation

To validate the practical efficacy of our proposed framework we constructed two experimental platforms. On the first system, the custom 3DOF direct drive Buehgler arm described in Section IV-B, we implemented a fully dynamical controller (6) based on the NF given in Appendix A, for

$n = 2$ and $m = 1$. Our second set of experiments employ an industrial 6DOF RTX robot from Universal Machine Intelligence on which we tested a kinematic controller (5) using the NF in Appendix A, for $n = 5$ and $m = 1$.

In both experimental setups discussed below, we used simple linear camera calibration (see, for example, [32]). For the RTX kinematic parameters, we used the manufacturer specified Denavit-Hartenberg parameters. For the Buehgler setup we measured the kinematic parameters roughly by hand. To refine this estimate, we affixed a point to the tip of the paddle and moved it to a known grid of joint space locations, and observed the corresponding grid of image plane points. Using these correspondences, gradient decent based on a simple pixel disparity cost function refined our parameter estimates for the thirteen parameters – two kinematic, six extrinsic, and five intrinsic camera parameters.

A. Buehgler Arm results

The Buehgler Arm is controlled by a network of two Pentium II computers running LynxOS, a commercial real-time operating system. The first captures 8bit 528x512 pixel images at 100Hz using an Epix Pixci D frame grabber connected to a DALSA CAD6 high-speed digital camera. The second computer implements servo control at a servo rate of 1kHz, based on the dynamical controller in (6).

Two sets of experiments were implemented with two different gain settings (i.e., assignments for the gains K_d and α from (6) and K from (25)) chosen to contrast performance resulting from a well tuned critically damped closed loop using relatively high gains, as against a "detuned" low gain and underdamped circumstance. Each trial consisted of driving the feature position and orientation to a model space goal (z^*, ζ^*) from some initial condition in joint space $(q_0, \dot{q}_0)$. Initial configurations were chosen by hand to be near the edge of the FOV, with initial velocity vectors chosen so as to drive the robot out of the FOV (i.e to "defeat" the controller). The initial conditions were prepared with a simple joint-space trajectory planner and joint-space PD controller that drove the robot to the starting state at which time the control switched to the NF based controller. In other words, we forced the robot to literally "fling" itself toward an obstacle before turning on our visual servoing controller. Both the goal positions and initial conditions where chosen to span the visible robot workspace.

For the "tuned" gain experiments, there were eight goals and forty initial conditions, for a total of 320 trials. We chose relatively high gains that provided a critically damped response and settling times on the order of a second.[8]

For the "detuned" gain experiments, a smaller set of more aggressive initial conditions and goal locations was

---

[8]Of course, the allusion to linear notions of damping is merely an intuitive designer's convenience. We chose gains K, K_d using standard PD tuning rules to ensure the local linearized system was nearly critically damped at the equilibrium state, and then turned up the "boundary" gain, α (6), to force reasonably snappy descent into the domain wherein the linearized approximation was dominant.
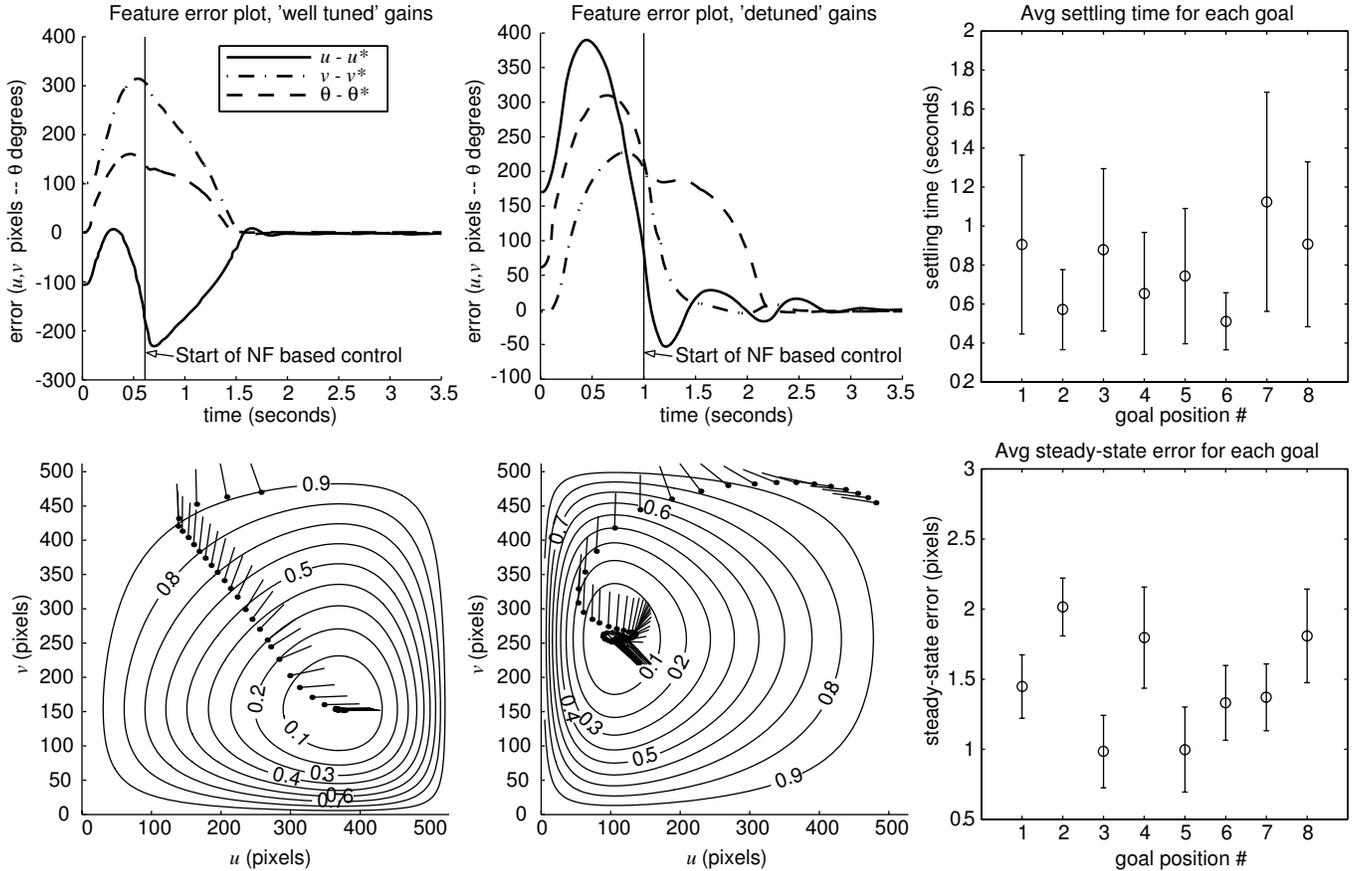
Fig. 5. For the dynamical, image-based visual servoing implementation on the Buehgler, we executed two batches of experiments: "tuned" gains and "detuned" gains, as described in the text. The first portion of motion correspond to joint-space limb motion setting up the initial condition, at which time (indicated by the vertical line in the error-vs-time plots) the NF based controller was switched on, in order to bring the paddle back to the desired image-plane position and orientation. **Left:** The pixel error and corresponding image plane feature trajectory for a typical 'well tuned' trial on the Buehgler robot. 2D level sets of the NF are superimposed on the image plane trajectory. **Middle:** A typical 'detuned' trial, with a different initial and goal locations. **Right:** Buehgler convergence results. *Top:* The mean steady-state pixel error for each of the eight goal positions. *Bottom:* Five percent settling time for each of the eight high-gain goal positions.

used, and the damping gain was reduced to provide "underdamped" performance. There were four goals and eight initial conditions, for a total of 32 trials.

Figure 5 shows the the error coordinates of a typical run for both "tuned" and "detuned" gains. With well tuned gains, the controller drove the feature to the goal location with a rate of success of 97%. Of the eleven errors one was due to the robot exceeding a software velocity limit, one to a software driver error, and one to a feature leaving the FOV of the camera during initialization. The remaining eight failures were caused by not allowing enough time for convergence as each experiment ran for a maximum of six seconds. These errors generally arose when the robot was close to a saddle of the NF so the controller was slow to overcome the robot's unmodeled friction. However, with "detuned" gains and high initial velocity the feature left the FOV 25% of the time. These failures were probably because the initial kinetic energy caused the arm to escape the artificial potential well – by using a lower "detuned" gain on the potential energy feedback term, the potential barrier is reduced. (It would not be difficult to compute

the invariant domain, as in [25].) These experiments give some sense of the relatively graceful performance degradation consequent upon imperfectly tuned gains. Figure 5 shows image-based error plots and the image-plane trajectory for two typical runs.

We designed our controller to have a very rapid and dexterous response. The Buehgler arm has a mass in excess of 100Kg making precise, quick and efficient movement quite challenging. Figure 5 (top right) shows our navigation based controller produced a one second or less five percent settling time for seven of the eight primary goal positions. Figure 5 (bottom right) presents the mean pixel error averaged over the final second of each trial. As can be seen the errors are in the neighborhood of 1 to 2 pixels over each of the eight goal positions.

B. RTX Arm results

The RTX is commanded through the serial port of a single Pentium PC running a Linux 2.0 kernel. The PC is equipped with a Data Translations DT3155 frame grabber connected to a standard 30Hz NTSC video camera.
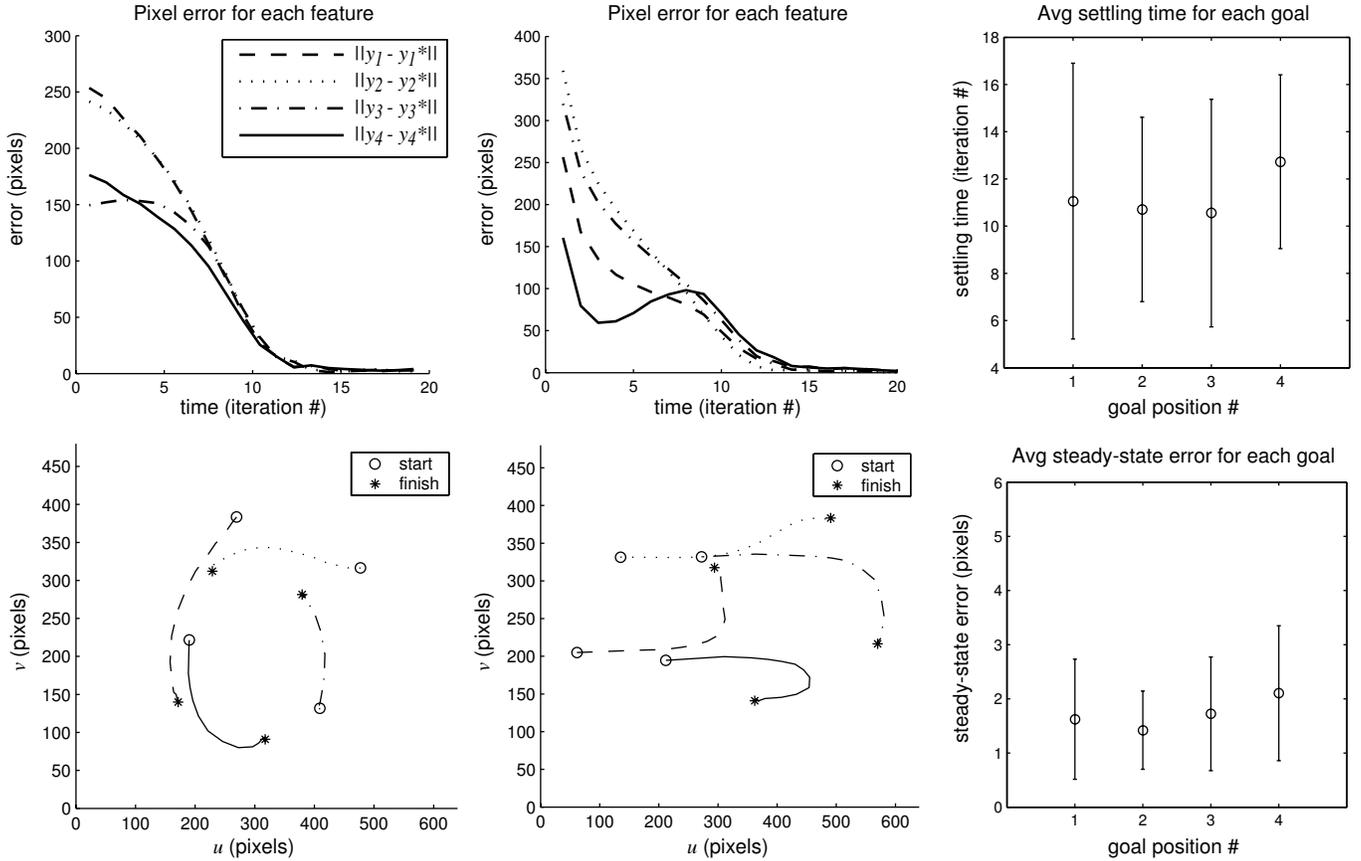
Fig. 6. For the quasi-static implementation on the RTX arm, one batch of 203 trials were selected. **Left:** The pixel error and corresponding image plane feature trajectory for a typical trial. **Middle:** Another typical trial, with a different initial condition and goal location. **Right:** RTX convergence results. *Top:* The mean pixel error for each of the four goal positions. *Bottom:* Five percent settling time for each of the four goal positions.

The theory presented in Section IV-C presumes the configuration space to be $\mathcal{Q} = \mathrm{SE}(3)$. However, unlike the Buehgler arm for which the robot joint space *was* the configuration space, in this case $\mathcal{Q}$ is parameterized only *locally* by the robot joint angles $q \in \mathbb{R}^6$ through the forward kinematics, namely $h : \mathbb{R}^6 \to \mathcal{Q}$. Of course, inevitably, all such kinematic parameterizations introduce singularities that may, in turn, inject spurious critical points to the gradient fields, necessarily actuated in the robot's joint space rather than in the task space, as our theory presumes. Similarly, since our formal theory "knows" only about visibility bounds, the robot's unmodeled joint space angles limits are not in principle protected against.[9] However, the weight of experimental evidence we present below suggests that these discrepancies between presumed model and physical reality do not seriously imperil the practicability of this scheme. Regarding the first discrepancy, the absence of stalled initial conditions suggests that any critical points so introduced were not attractors. Regarding the second, we found that choosing initial and goal locations away from the joint space boundaries was sufficient to avoid running into the end-stops.

The RTX controller employs first order gradient descent on the NF presented in Appendix A. Because the RTX arm accepts only position commands, given goal and current images with feature points extracted, the gradient update was implemented iteratively, as follows:

$$
\begin{aligned}
u_k &\Leftarrow -D_q^T \varphi = -D_q^T (g \circ h)(q_k)\, D_z^T \widetilde{\varphi}_{z^*}(z_k), \\
q_{k+1} &\Leftarrow q_k + \beta u_k \quad \text{(where } \beta \text{ is the step size)}.
\end{aligned}
$$

To explore our algorithm, we conducted a set experiments in which 58 initial conditions and four goal locations were chosen randomly from a grid of 4096 points in model space (configurations near kinematic singularities, not within the robot workspace, or which resulted in feature projections outside the FOV were removed, resulting in a total of 203 trials). Initially, the robot was moved to each goal location to capture an image of the robot, respecting which the vision system stored the desired location of feature points, y^*. Figure 6 shows the pixel errors feature trajectories of two typical runs. As shown, we used four coplanar feature points for the camera map,

---

[9]Addressing the further practical realities of kinematic singularities and robot joint space limitations falls outside the scope of the present paper (and, indeed, is not even addressed at all in the traditional visual servoing literature). In principle, the NF framework would be relevant to these problems as well: joint space limits are analogous to the FOV obstacles, while the kinematic singularities are akin to self-occlusion.

$c : \mathcal{Q} \to \mathcal{Y}$. Of 203 trial runs, 5 were found to have failed. In each case the failure was due to the robot not converging within our software imposed limit of thirty iterations (visual inspection of each such run revealed that convergence was eminent).

Using the mean over the four feature points, we found an average final pixel error on the order of 1-2 pixels upon convergence. Figure 6 (upper right) shows the mean pixel error and standard deviation for each of the four unique goal positions. The average five percent setting time, shown in Figure 6 (lower right), was approximately 10-14 iterations for each of the four goal locations, averaged over all successful trials.

VI. Conclusions

This paper addresses the problem of driving image plane features to some goal constellation while guaranteeing their visibility at all times along the way. We cast the problem as an instance of generalized dynamical obstacle avoidance, thereby affording the use of navigation functions in a nonlinear PD-style feedback controller.

Ideally, one creates an image based coordinate system using direct feature information as in the two settings of Section IV-A and IV-B. In those cases the occlusion obstacles manifest themselves in *image coordinates*, and hence the NF gradient in model space coordinates will not depend on the specific parameters of our camera or robot. Moderate calibration uncertainty will lead to errors in our estimate of the image Jacobian, but we believe that this will not imperil successful obstacle avoidance. However, construction of a 6 DOF image-based coordinate system for 3D rigid body motion represents work in progress, thus we have resorted to a task-space visual servoing algorithm in Section IV-C.

The two experimental systems – the custom 3DOF Buehgler Arm and the 6DOF commercial RTX arm – confirmed the practicability of the theoretical framework. For the Buehgler, our experiments suggest that the navigation function based controller indeed achieves extremely high performance, though in a few cases we were able to "defeat" the controller with deliberately adversarial initial conditions and poor gain tuning. The kinematic experiments with the RTX validated our 6DOF task-space servo architecture. In both cases our results show systems with large basins of attraction that both avoid self occlusion and respect FOV constraints.

VII. Acknowledgments

Appendix

I. A navigation function for $[-1, 1]^n \times \mathrm{T}^m$

Let

$$\mathcal{Z} = [-1, 1]^n \times \mathrm{T}^m \tag{23}$$

for some $m, n \in \mathbb{N}$ and let $(z^*, \zeta^*) \in \overset{\circ}{\mathcal{Z}}$ denote a goal. Consider the function $f : (-1, 1)^n \to \mathbb{R}^n$

$$f(z) = \begin{bmatrix} \frac{z_1 - z_1^*}{(1 - z_1^2)^{\frac{1}{2}}} & \cdots & \frac{z_n - z_n^*}{(1 - z_n^2)^{\frac{1}{2}}} \end{bmatrix}^T.$$

Let $K \in \mathbb{R}^{n \times n}$ be a positive definite symmetric matrix and $\kappa_i > 0$, $i = 1, \ldots, m$. Define

$$\overline{\varphi}(z, \zeta) := \tfrac{1}{2} f(z)^T K f(z) + \sum_{i=1}^{m} \kappa_i (1 - \cos(\zeta_i - \zeta_i^*)). \tag{24}$$

Proposition 3: The objective function

$$\widetilde{\varphi} := \frac{\overline{\varphi}}{1 + \overline{\varphi}} \tag{25}$$

is a navigation function on $\mathcal{Z}$, where $\overline{\varphi}$ is given in (24).

Proof: According to Definition 1, $\widetilde{\varphi}$ must be a smooth Morse function which evaluates uniformly to unity on the boundary of $\mathcal{Z}$, and has (z^*, ζ^*) as the unique minimum.

The boundary of $\mathcal{Z}$ is given by

$$\partial \mathcal{Z} = \{(z, \zeta) \in \mathcal{Z} : z_i = \pm 1, i \in \{1, \ldots, n\}\}.$$

Clearly, $\widetilde{\varphi}$ evaluates to 1 on the boundary, i.e. as $z_i \to \pm 1$ then $\widetilde{\varphi} \to 1$. Furthermore, $\forall (z, \zeta) \in \mathcal{Z}$, $\overline{\varphi} >= 0$. Moreover, $\overline{\varphi} = 0$ iff $(z, \zeta) = (z^*, \zeta^* + \sum_{i \in \Gamma} 2\pi e_i) = (z^*, \zeta^*)$, i.e. (z^*, ζ^*) is the global minimum.

To study the critical points of $\widetilde{\varphi}$, we need only study those of $\overline{\varphi}$, because the function $\sigma : [0, \infty) \to [0, 1)$ given by $\sigma(x) = x/(1 + x)$ has derivative $\sigma'(x) = 1/(1 + x)^2$, which does not introduce any spurious critical points. The critical points of $\overline{\varphi}$ are found by solving

$$0 = D\overline{\varphi} =$$
$$\begin{bmatrix} f^T K Df, & \kappa_1 \sin(\zeta_1 - \zeta_1^*), \cdots, \kappa_m \sin(\zeta_m - \zeta_m^*) \end{bmatrix} \tag{26}$$

noting that

$$Df = \mathrm{diag}\{f'\}, \quad \text{where} \quad f_i'(z) := \frac{1 - z_i z_i^*}{(1 - z_i^2)^{3/2}},$$
$$i = 1, \ldots, n.$$

Since Df is nonsingular on $(-1, 1)^n$, $D\overline{\varphi} = 0$ iff $f = 0$ and $\sin(\zeta_i - \zeta_i^*) = 0$, $i = 1, \ldots, n$ which is true iff $(z, \zeta) = (z^*, \zeta^* + \sum_{i \in \Gamma} \pi e_i)$, $\Gamma \in \mathrm{powerset}\{1, \ldots, m\}$. There are 2^m index sets which enumerate all possible critical points. One readily verifies that the Hessian is nonsingular at every critical point and (z^*, ζ^*) is the only minimum. Hence $\widetilde{\varphi}$ is a Morse function which evaluates uniformly to unity on the boundary, has $2^m - 1$ saddles and the goal is the unique minimum. ∎

References

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, pp. 651–670, October 1996.

[2] P. I. Corke and M. C. Good, "Dynamic effects in visual closed-loop systems," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996.

[3] Francois Chaumette, *The Confluence of Vision and Control*, chapter Potential problems of stability and convergence in image-based and position-based visual servoing, Springer-Verlag, 1999.

[4] Louis L. Whitcomb, Alfred A. Rizzi, and Daniel E. Koditschek, "Comparative experiments with a new adaptive contoller for robot arms," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp. 59–70, Feb 1993.

[5] Ezio Malis and Francois Chaumette, "Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 176–186, April 2002.

[6] Camillo J. Taylor and James P. Ostrowski, "Robust visual servoing based on relative orientation," in *International Conf. on Robotics and Automation*, 1998.

[7] Alfred A. Rizzi and Daniel E. Koditschek, "An active visual estimator for dexterous manipulation," *IEEE Transactions on Robotics and Automation*, pp. 697–713, October 1996.

[8] D. Kim, Alfred A. Rizzi, Gregory D. Hager, and Daniel E. Koditschek, "A "robust" convergent visual servoing system," in *International Conf. on Intelligent Robots and Systems*, Pittsburgh, PA, 1995, IEEE/RSJ.

[9] Noah J. Cowan and Daniel E. Koditschek, "Planar image based visual servoing as a navigation problem," in *International Conference on Robotics and Automation*, Detroit, MI, 1999, IEEE, vol. 1, pp. 611–617.

[10] Noah J. Cowan, Gabriel A. D. Lopes, and Daniel E. Koditschek, "Rigid body visual servoing using navigation functions," in *Conference on Decision and Control*, Sydney, Australia, 2000, IEEE, pp. 3920–3926.

[11] Noah J. Cowan, Joel D. Weingarten, and Daniel E. Koditschek, "Empirical validation of a new visual servoing stragegy," in *Conference on Control Applications*, Mexico City, September 2001, IEEE, Omnipress.

[12] G. Morel, T. Leibezeit, J. Szewczyk, S. Boudet, and J. Pot., "Explicit incorporation of 2d constraints in vision based control of robot manipulators," in *Experimental Robotics VI*, Peter Corke and James Trevelyan, Eds., vol. 250 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2000, ISBN: 1 85233 210 7.

[13] Peter I. Corke and Seth A. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 507–515, 2001.

[14] Youcef Mezouar and Francois Chaumette, "Design and tracking of deserable trajectories in the image space by integrating mecahnical and visibility constraints," in *International Conference on Robotics and Automation*, Seoul, Korea, 2001, IEEE, vol. 1, pp. 731–6.

[15] Hong Zhang and Jim Ostrowski, "Visual motion planning for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 199–208, April 2002.

[16] Hong Zhang and James P. Ostrowski, "Visual servoing with dynamics: Control of an unmanned blimp," in *International Conf. on Robotics and Automation*, 1999.

[17] Tarek Hamel and Robert Mahony, "Visual servong of an under-actuated dynamic rigid-body system: An image-based approach," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 187–198, April 2002.

[18] Jean-Claude Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 1991.

[19] Daniel E. Koditschek, "The application of total energy as a Lyapunov function for mechanical control systems," in *Dynamics and control of multibody systems (Brunswick, ME, 1988)*, pp. 131–157. Amer. Math. Soc., Providence, RI, 1989.

[20] Daniel E. Koditschek and Elon Rimon, "Robot navigation functions on manifolds with boundary," *Advances in Applied Mathematics*, vol. 11, pp. 412–442, 1990.

[21] Elon Rimon and Daniel E. Koditschek, "Exact robot navigation using artificial potential fields," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, Oct 1992.

[22] J. Craig, *Introduction to Robotics*, Addison-Wesley, Reading, Mass., 1986.

[23] Richard M. Murray, Zexiang Li, and S. Shankar Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Reading, Mass., 1994.

[24] Morris W. Hirsch, *Differential Topology*, Springer-Verlag, 1976.

[25] Daniel E. Koditschek, "The control of natural motion in mechanical systems.," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 113, no. 4, pp. 547–551, Dec 1991.

[26] Noah Cowan, "Binocular visual servoing with a limited field of view," in *Mathematical Theory of Networks and Systems*, Notre Dame, Indiana, August 2002.

[27] Ezio Malis, Francois Chaumette, and Sylvie Boudet, "2-1/2-d visual servoing," *IEEE Transactions on Robotics and Automation*, pp. 238–250, 1999.

[28] Dan Pedoe, *Geometry, a comprehensive course*, Dover Publications, Inc., New York, 1970.

[29] Daniel E. Koditschek, "An approach to autonomous robot assembly," *Robotica*, vol. 12, pp. 137–155, 1994.

[30] Noah J. Cowan, Joel D. Weingarten, and Daniel E. Koditschek, "Visual servoing via navigation functions," Tech. Rep. CSE-TR-449-02, The University of Michigan, Dept of EECS, http://www.eecs.umich.edu/eecs/research/techreports/-compsci.html, 2002.

[31] Noah J. Cowan, *Vision-based Control via Navigation Functions*, Ph.D. thesis, University of Michigan, 2001.

[32] Olivier Faugeras, *Three-Dimensional Computer Vision*, MIT Press, London, England, 1993.

Noah J. Cowan (S'98–M'01) received the B.S. degree from The Ohio State University, Columbus, in 1995, and the Ph.D. degree from the University of Michigan, Ann Arbor, in 2001, both in electrical engineering.

He is currently completing postdoctoral study at the University of California at Berkeley. His research interests include vision-based control, machine learning, and biologically inspired robot design.

Dr. Cowan is a member of the IEEE Robotics and Automation, IEEE Controls Systems Technology, and IEEE Computer Societies.

Joel D. Weingarten (S'02) received the B.Sc. degree with distinction from York University, Toronto, ON, Canada. He is currently working toward the Ph.D. degree in the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor.

His current research efforts include work on legged robot gait optimization and generation.

Mr. Weingarten is a member of the IEEE Robotics and Automation and IEEE Computer Societies.

Daniel E. Koditschek (S'80–M'83–SM'01) received the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, in 1983. He was appointed to the Yale faculty in 1984 and moved to the University of Michigan, Ann Arbor, in 1993, where he is presently a Professor in the Department of Electrical Engineering and Computer Science. His research interests include robotics and, generally, the application of dynamical systems theory to intelligent mechanisms.

Prof. Koditschek is a member of the American Association for the Advancement of Science, American Mathematical Society, Association for Computing Machinery, Mathematical Association of America, Society for Industrial and Applied Mathematics, Society for Integrative and Comparative Biology, and Sigma Xi.