# Visual Registration and Navigation using Planar Features

Gabriel A. D. Lopes
EECS Department, College of Engineering
University of Michigan, Ann Arbor, MI
Email: glopes@umich.edu

Daniel E. Koditschek
EECS Department, College of Engineering
University of Michigan, Ann Arbor, MI
Email: kod@umich.edu

*Abstract*— This paper addresses the problem of registering the hexapedal robot, RHex, relative to a known set of beacons, by real-time visual servoing. A suitably constructed navigation function represents the task, in the sense that for a completely actuated machine in the horizontal plane, the gradient dynamics guarantee convergence to the visually cued goal without ever losing sight of the beacons that define it. Since the horizontal plane behavior of RHex can be represented as a unicycle, feeding back the navigation function gradient avoids loss of beacons, but does not yield an asymptotically stable goal. We address new problems arising from the configuration of the beacons and present preliminary experimental results that illustrate the discrepancies between the idealized and physical robot actuation capabilities.

## I. INTRODUCTION

This paper reports on our progress in adapting the fixed camera, moving beacon visual servoing algorithms of Cowan et al [1] to the beacon "inside out" version of the problem — a moving camera reacting to a fixed beacon — that arises when attempting to register a mobile robot vehicle relative to some effective landmark in its visual field. Specifically, we are interested in applying these ideas to the hexapedal robot, RHex [2], [3] considered as operating in the (three degree of freedom) horizontal plane. The lower level controls presently operative in our legged machine result in horizontal plane behavior nicely modeled by a unicycle [4]. The reduced affordance of this nonholonomically constrained model precludes the possibility of point stabilization by any smooth feedback law [5], and the navigation function will eventually play the role of a control Lyapunov function [6] in this research. In the present paper, we illustrate the interplay between the navigation function (our task model), its realization in physical hardware, and the preliminary navigation results that we have obtained to date both in extensive simulation studies and on the physical RHex platform.

### A. Background Literature

Several authors have dealt with the problem of vision based navigation [7], [8], almost exclusively, to date, in indoor environments. Ostrowski [9], [10] uses a blimp equipped with a camera that implements a diffeomorphism between image plane features and robot pose to maintain a constant distance from the beacon. Ezio and Chaumette [11] decouple the rotation and translation degrees of freedom to position a fully actuated camera arm in relation to a collection of features. Cowan [1], [12], [13] servos a 6 dof arm to a predefined pose, by introducing a Navigation function that guarantees the features stay in the field of view (FOV) of the camera at all times.

The problem to be solved in this paper entails navigation of an autonomous hexapod robot in an environment with known beacons using vision. The paper is an extension of Cowan's work [12] in the sense that it generalizes the configuration of the beacons (landmarks) in the planar version of the problem and implements the controller on a legged platform, maintaining the same emphasis on convergence to the goal with no FOV violations, modulo the reduced control affordance introduced by the kinematic constraints of the mobile platform.

The mobile platform of present interest is RHex, [2], [3], (illustrated in figure 1), a hexapedal machine with passive compliant legs that afford impressive mobility. Much of the theoretical inquiry into this machine has been confined to its behavior in the sagittal plane [3], [14], leaving a significant gap in the characterization of its operation in the horizontal plane. For present purposes, when only small accelerations of the body are required, we will find it acceptable to characterize RHex's horizontal plane mechanics via the standard quasi-static "unicyle" model — a nonholonomically constrained machine whose velocity can be commanded in the fore-aft and heading directions relative to the body. The efficacy of this highly simplified model for the present quasi-static operating regime is documented by a comparison of simulation and experimental results, below. No doubt, extending these techniques to the full dynamical regime of which RHex is capable will require a plant model far more accurately informed about its complex Lagrangian mechanics.

### B. Organization of the paper

Section II describes analytically the generalization of the visual servoing algorithm developed by Cowan [13]. In section III it is shown that different configurations of beacons yield different pose measurement error for a given location and therefore, a proper choice of beacons may help improve the accuracy of the pose. Simulations of the globally convergent controller are made for a fully actuated rigid body and for a unicycle. Finally section IV describes the implementation on RHex.
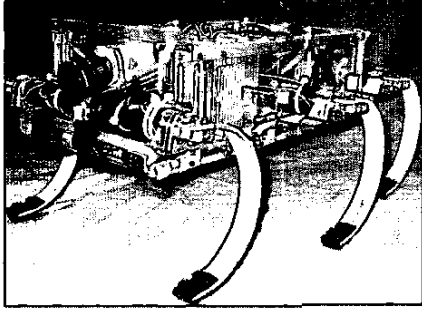
Fig. 1. RHex - Robot hexapod



Fig. 2. The parameters $(\alpha, \rho_1, \rho_2) \in \mathcal{B}$ define the beacon configuration. The parameters $(\phi, \psi, \beta)$ define the coordinates in $\mathcal{Q}$, the robot's configuration space. The dashed lines represent the FOV boundary with parameters $(\theta_{min}, \theta_{max})$. The line $\mathcal{I}$ represents the camera projection line and the points $(x_1, x_2, x_3)$ are the projection of each beacon $b_i$ into $\mathcal{I}$.

## II. 2D VISUAL SERVOING

Throughout this paper we assume a perfectly calibrated 1 dimensional pinhole camera, and our robot model assumes motion in only the horizontal plane. We further assume that the geometry of the beacons is known and the correspondence problem is solved. The algorithm presented here uses 3 beacons to extract and regulate full relative pose. In [12] it is shown that an algebraic inverse of the camera map is obtained for 3 collinear points in $SE(2)$. Although this result may be sufficient for some applications, for real environments it will become important for the robot to be able to handle more general configurations of beacons. With appropriate image pre-processing one can assign signatures to natural elements like trees, stones, etc, but natural beacons are, in general, non-collinear. We now generalize the methods of [12], [13] to accommodate arbitrary beacon configurations.

### A. Pose computation

The first step is to define a parameterization of the beacons and find the camera map that relates the projected beacon coordinates in the camera image line $\mathcal{I}$ to the robot pose in $SE(2)$ (the camera projection plane is reduced to a line in $SE(2)$). For any configuration of beacons in a plane, define the beacon parameter space $\mathcal{B} \subset \mathbb{R}^2 \times S$ by fixing the world frame so that the second point is at the origin and the remaining points lie in lines going through the origin with congruent angles. Figure 2 illustrates the configuration of the beacons with parameters $(\alpha, \rho_1, \rho_2)$.

$$\mathcal{B} := \{(\rho_1, \rho_2, \alpha) \in SE(2) \mid \rho_1 > 0, \rho_2 > 0, 0 \leq \alpha < \pi\}$$

The coordinates of each beacon $b_i$ in the world frame are:

$$\begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} = \begin{bmatrix} \rho_1 R_\alpha \hat{e}_y & 0 & \rho_2 R_\alpha^T \hat{e}_y \end{bmatrix}$$

To build the camera map it is convenient to use polar coordinates, in effect passing to a new space $\mathcal{Q}$, diffeomorphic to the robot configuration space. $\mathcal{Q}$ expresses in a computationally tractable form the fact that the robot configuration space has the topology of a solid torus (after removing a disk enclosing the beacons from the robot's available workspace). The motivation to introduce such a coordinate system arises from the fact that in $\mathcal{Q}$ the set of self-occlusions appears as a
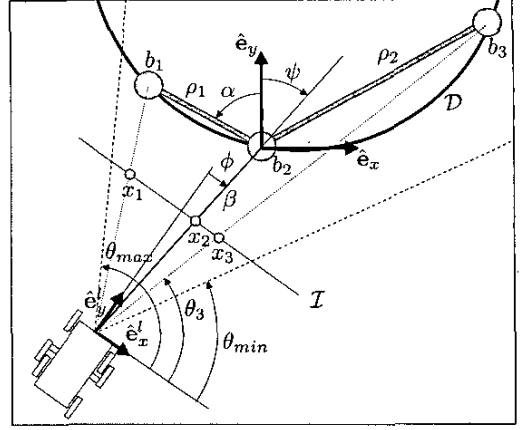
literal (2 dimensional) torus, providing significant geometrical insight into the self-occlusion problem. Figure 2 illustrates the parameterization of the space $\mathcal{Q}$ with parameters $(\phi, \psi, \beta)$.

Having adopted a representation for the beacon configuration and the robot configuration space it is now necessary to determine for a given beacon the set of robot configurations for which occlusion-free servoing can be accomplished. Define the facing set $\mathcal{F}$ as the set of configurations for which the robot lies "in front" of the set of beacons, i.e. the beacons appear to face the robot sensor. Intuitively the beacons must keep a certain order in the camera projection line $\mathcal{I}$. Define the function $f_i$ that returns a vector that goes through beacon $b_i$ for a given configuration $q = (\phi, \psi, \beta)$.

$$f_i(q) := R_\phi R_\psi b_i + \beta R_\phi \hat{e}_y$$

The facing set if then defined by (1) where $J$ is a skew symmetric matrix:

$$\mathcal{F} := \{q \in \mathcal{Q} \mid f_i(q)^T J f_j(q) > 0; i < j\} \qquad (1)$$

Define the visible set $\mathcal{V}$ as the set of configurations for which the beacons are in the FOV of the camera sensor, where $\theta_{min}, \theta_{max}$ are the FOV camera parameters illustrated in figure 2 and function "$\angle$" returns the angle of a vector.

$$\mathcal{V} := \{q \in \mathcal{Q} \mid \theta_{min} < \angle(f_i(q)) < \theta_{max}, i = 1, 2, 3\}$$

The previous sets arise from geometrical insight, necessary for the vision implementation, but are not sufficient to fully characterize the set of configurations for which pose computation can be accomplished. In fact, as shown next, the camera map may not always be injective in $\mathcal{F} \cap \mathcal{V}$. This is due to the generalization of [13] by allowing any beacon configuration. It is shown here that the injectivity is lost at worst on the zero set of the function $\Theta_b$ (a factor in the determinant of the jacobian of $c_b$) in which set the inverse image can have
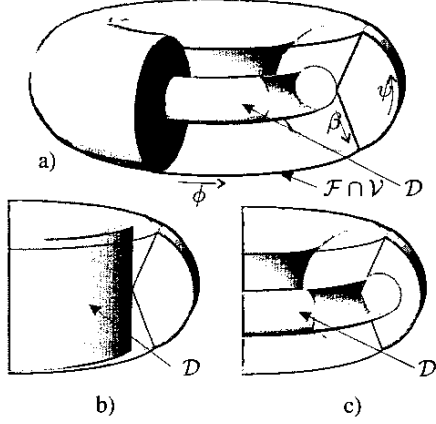
Fig. 3. Illustration of $\mathcal{Q}$. The thick black lines represent the intersection of the visible set and facing set slicing $\mathcal{Q}$. a) convex configuration. b) linear configuration. c) concave configuration. For the concave configuration c) $\mathcal{D}$ disconnects $\mathcal{F} \cap \mathcal{V}$.

cardinality 2. We then introduce the degenerate set $\mathcal{D}$ using the function $\Theta_b$:

$$\Theta_b(q) \quad : \quad \mathcal{B} \times \mathcal{Q} \to \mathbb{R}$$
$$q \mapsto \rho_1 \sin(\alpha - \psi) + \rho_2 \sin(\alpha + \psi) + \beta \sin(2\alpha)$$

$$\mathcal{D} := \{b \in \mathcal{B}, q \in \mathcal{Q} \mid \Theta_b(q) = 0\}$$

If the configuration space is understood topologically as a thickened torus, then the degenerate set will in general be a thin torus that disconnects $\mathcal{Q}$. If the three beacons are collinear then $\mathcal{D}$ becomes a cylinder. Figure 3 illustrates the solid torus sliced by the FOV of the camera sensor. One should notice that when the FOV slices the configuration space it may be the case that the degenerate set does not disconnect the facing set $\mathcal{F} \cap \mathcal{V}$ as in figures 3a) and 3b). In fact, only if the set of beacons is configured in a concave shape will $\mathcal{D}$ disconnect $\mathcal{F}$ (figure 3c)). Define the free configuration space $\mathcal{W} \subset \mathcal{Q}$ by:

$$\mathcal{W} := (\mathcal{F} \cap \mathcal{V}) - \mathcal{D}$$

**Proposition 1.** *There exists a smooth and smoothly invertible map from the free configuration space $\mathcal{W}$ into a subset of $T^3$.*

*Proof:* Consider the map:

$$c_b(q) \quad : \quad \mathcal{W} \to T^3 \tag{2}$$
$$q \mapsto \begin{bmatrix} \angle(f_1(q)) & \angle(f_2(q)) & \angle(f_3(q)) \end{bmatrix}^T$$

Smoothness is easily verifiable. It is sufficient to show that the map $c_b$ is a local diffeomorphism in the neighborhood of a point in $\mathcal{W}$ and that the cardinality of the inverse image in the co-domain is unity. This is necessary since $\mathcal{D}$ disconnects $\mathcal{F} \cap \mathcal{V}$.

The determinant of the Jacobian of the camera map degenerates only on the degenerate set and outside the visible and

facing sets. Therefore $c_b$ is a local diffeomorphism in $\mathcal{W}$:

$$|D_q c_b| = \frac{\rho_1 \rho_2 \Theta_b}{\|f_1\|^2 \|f_2\|^2}$$

Now suppose that there exist two configurations $q'$ and $q''$ such that $c_b(q') = c_b(q'')$. This is equivalent to saying that each of the points $f_i(q'')$ and $f_i(q'')$ are in a line that goes through the origin, i.e.:

$$f_i(q'') \times f_i(q'') = 0 \tag{3}$$

For $i = 2$ we get $(\beta' R_{\phi'} \hat{e}_y)^T J (\beta'' R_{\phi''} \hat{e}_y) = 0$ which simplifies to $\beta' \beta'' \sin(\phi' - \phi'') = 0$. Since $\beta'$ and $\beta''$ cannot be null this results in $\phi' = \phi'' + k\pi$ with $k \in \mathbb{N}$. For $i = 1, 3$ equation (3) simplifies to:

$$\begin{cases} \Theta_b(q'') \sin(\psi' + \alpha) - \Theta_b(q') \sin(\psi'' + \alpha) = 0 \\ \Theta_b(q'') \sin(\psi' - \alpha) - \Theta_b(q') \sin(\psi'' - \alpha) = 0 \end{cases} \tag{4}$$

Eliminating $\Theta_b(q')$ and $\Theta_b(q'')$ from the previous equations we get:

$$\sin(2\alpha) \sin(\psi' - \psi'') = 0 \Rightarrow \psi' = \psi'' + k\pi, k \in \mathbb{N}$$

Finally using equation (4) with $\psi'' = \psi' = \psi$ completes the result:

$$\begin{cases} \rho_1(\beta' - \beta'') \sin(\alpha + \psi) = 0 \\ \rho_2(\beta' - \beta'') \sin(\alpha - \psi) = 0 \end{cases} \Rightarrow \beta' = \beta''$$

$\square$

Call $c_b$ the camera map and define the set $\mathcal{Y} = c_b(\mathcal{W})$. To find the inverse camera map the same constructive method is used as in [12]. Let the projection of the beacons in the camera projection line be $(\theta_1, \theta_2, \theta_3) \in T^3$ (i.e. $\theta_i = \arctan(x_i) + \pi/2$ as illustrated in figure 2) and let $Y$ and $Y'$ be:

$$Y = \begin{bmatrix} \cos(\theta_1) & \cos(\theta_2) & \cos(\theta_3) \\ \sin(\theta_1) & \sin(\theta_2) & \sin(\theta_3) \end{bmatrix}$$

$$Y' = \begin{bmatrix} \rho_1 \cos(\theta_1 - \alpha) & 0 & \rho_2 \cos(\theta_3 + \alpha) \\ \rho_1 \cos(\theta_1 - \alpha) & 0 & \rho_2 \cos(\theta_3 + \alpha) \end{bmatrix}$$

The robot's pose is computed by the following expressions, where $Y^\dagger$ is the pseudo-inverse of $Y^T$ and $Y_\perp$ is the orthogonal complement of the subspace generated by the lines of $Y^\dagger$:

$$\phi = \theta_2 + \frac{\pi}{2}$$
$$\psi = \angle(\delta R_\phi^T J Y' Y_\perp) \tag{5}$$
$$\beta = \frac{\|Y^\dagger Y'^T J Y' Y_\perp\|}{\|Y' Y_\perp\|}$$

Having an explicit closed form expression for the camera map and its inverse parameterized by the beacon configuration, one may now address the question: how does the beacon configuration affect the pose computation error? For convex and collinear beacons it is expected that the pose computation error grows with distance, but for concave beacon configuration a more complex error structure is expected, due to the degenerate set $\mathcal{D}$. Section III approaches this question through a numerical study.

3937

For concave beacons the free configuration space $\mathcal{W}$ is disconnected. A new question arises: If the robot's initial and goal pose lie in distinct connected components of $\mathcal{W}$ can occlusion-free navigation be accomplished? In other words, is it possible to "puncture" the disconnecting degenerate set $\mathcal{D}$? In section III-B numerical simulation suggest that in general it is possible to accomplish global convergent occlusion-free navigation even in the presence of a disconnecting degenerate set $\mathcal{D}$.

### B. Navigation function

Since the camera map is a diffeomorphism between the free configuration space $\mathcal{W}$ and the projected beacons on the camera projection line $\mathcal{I}$, one can build a potential function $\varphi$ so that the system $\dot{\mathbf{y}} = -\nabla \varphi(\mathbf{y})$ is globally asymptotically stable in $\mathcal{Y}$. Next, we use the camera map $c_b$ to pull back the velocities from a known globally convergent system into $\mathcal{Q}$. Let $\varphi$ be a potential function:

$$\varphi(\theta) \quad : \quad T^3 \to [0,1]$$

$$\theta \mapsto \frac{\bar{\varphi}(\theta)^k}{\epsilon + \bar{\varphi}(\theta)^k} \tag{6}$$

$$\bar{\varphi}(\theta) := \frac{\left(\sum_{i=1}^{3}(\theta_i - \theta_i^*)^2\right)^m}{(\theta_{max} - \theta_1)(\theta_1 - \theta_2)(\theta_2 - \theta_3)(\theta_3 - \theta_{min})}$$

By construction the function $\varphi$ equals unity on the boundary of $\mathcal{Y}$ and has a global minima at the goal configuration. $\varphi$ is also continuous and differentiable and therefore it is a navigation function in $\mathcal{Y}$ as defined in [15]. The parameters $\epsilon, k$ and $m$ shape the function $\varphi$ to allow fine tuning of the resultant velocity vector field, $(\theta_1^*, \theta_2^*, \theta_3^*)$ represent the robot goal configuration in $T^3$ and $(\theta_{min}, \theta_{max})$ are the FOV parameters described in figure 2.

The final ingredient is to pullback the gradient vector field $\nabla \varphi$ into the world space. Two new maps are introduced to accomplish that: $\Phi$ maps coordinates in $\mathcal{Q}$ into $SE(2)$ in the local robot frame. $\Upsilon$ maps local robot coordinates into world coordinates. See the appendix for details on these maps. Define the full camera map $\bar{c}_b = c_b \circ \Phi \circ \Upsilon(x_w)$. Writing the gradient system in the world space we get 7, then apply the chain rule on $\bar{c}_b$.

$$\dot{\mathbf{x}}_w = \mathbf{u} = -\nabla\left(\varphi \circ \bar{c}_b\right)(x_w) = -D\bar{c}_b^{\ T} \cdot \nabla \varphi(x_w) \tag{7}$$

### C. Unicycle model

In the previous sections it is assumed that the robot is fully actuated. In reality the dynamical model of RHex, used in the experiments, is not yet fully modeled in all gaits and terrains of interest. Several assumptions are made in order to implement the algorithms previously described. Most importantly, on the strength of empirical experience and the longer term theoretical perspective of [4], we adopt for RHex's horizontal plane behavior the model of a quasi-static unicycle. The motion control software written for RHex implements a tripod gate for a normal walk. At any time 3 legs always touch the ground. The "walk mode" used in the experiments has the
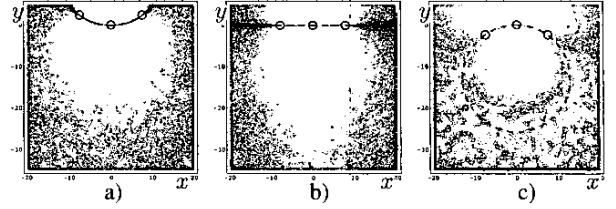


Fig. 4. Illustration of simulated pose computation error. a) Convex beacons; b) collinear beacons; c) Concave beacons. Darker values mean larger error. Beacons are represented by the small circles and the image of the degenerate set $\mathcal{D}$ is represented by the dashed curves.

control inputs of forward velocity and turn velocity, therefore the unicycle model is the natural choice to implement. The input velocity vector of the robot is computed by projecting the desired velocity vector given by equation (7) into the $y$ axis of the local body frame $(\hat{e}_y^l)$. The turn velocity is applied directly. Let $[\mathbf{u}\ u_\theta]^T = -\nabla (\varphi \circ \bar{c}_b)$. Then the unicycle model equation becomes:

$$\begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\theta}_w \end{bmatrix} = \begin{bmatrix} -\sin(\theta_w) & 0 \\ \cos(\theta_w) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \langle \mathbf{u}, \hat{e}_y^l \rangle \\ u_\theta \end{bmatrix} \tag{8}$$

Local minima are introduced when the projection of the velocity vector yields a zero vector and the turn velocity is null. Numerical simulations verify this fact in section III-C.

## III. SIMULATIONS

### A. Pose computation error

For concave beacon configurations the degenerate set disconnects the free configuration space $\mathcal{W}$. Therefore, it is expected that in a small neighborhood of $\mathcal{D}$ the camera map Jacobian is small potentially introducing large numerical errors. To visualize the extent and magnitude of this problem, pose computation error equation (9) was simulated for intervals of $x_w$ and $y_w$ with $\theta_w = 0$. A random noise vector $\delta$ with Gaussian distribution is added to the computation of the inverse camera map to simulate the noise from the camera. The simulated error is computed by:

$$\mathbf{e}_\delta = \|\mathbf{x}_w - \bar{c}_b^{-1}(\delta + \bar{c}_b(\mathbf{x}_w))\| \tag{9}$$

One can notice that the error increases with distance from the beacons as expected. In figure 4 c) the pose computation error increases when the robot is close to $\mathcal{D}$. This clearly suggests that concave beacon configurations are not desirable.

### B. Fully actuated rigid body

Figure 5 illustrates the simulation of a fully actuated body using equation (7). The initial conditions range from $x \in [-5, 5]$ meters, $y = -5$ meters and $\theta \in [0, \frac{\pi}{2}]$. The goal location is at $(0, -2, 0)$. In section II-A the possibility of puncturing the degenerate set is contemplated. Figure 5 shows that in simulation, using a concave beacon configuration, the algorithm converges successfully in all the trials. This suggests that in theory it is safe to puncture the degenerate set.
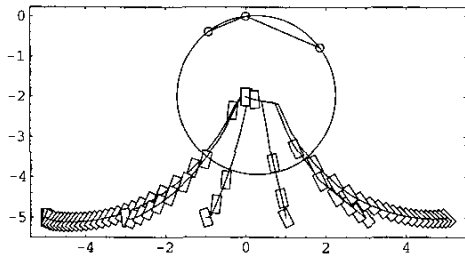
Fig. 5. Simulation of a fully actuated body with a concave beacon configuration. The degenerate set $\mathcal{D}$ is represented by the large circle. On top of the trajectories represented by the solid lines, the pose of the robot is plotted for fixed time intervals to give a crude idea of the robot's velocity.
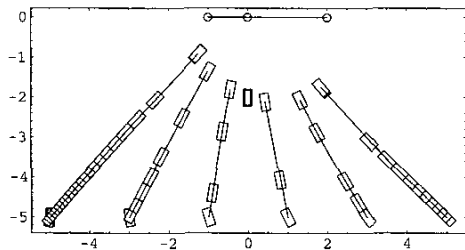


Fig. 6. Simulation of the non-holonomic constrained unicycle with a linear beacon configuration. The goal pose is represented by the gray rectangle.

Nevertheless, as shown previously, the pose computation error increases close to $\mathcal{D}$ and therefore, although puncturing is safe in theory, it should be avoided in practice.

*C. Unicycle model*

Figure 7 illustrates the same simulation of figure 5 but now using the unicycle model. Figure 6 illustrates a simulation for a linear beacon configuration. For both figures 6 and 7 the non-holonomic constraint predictably introduces local minima and, in general, as expected, the robot does not reach the goal. As figure 7 illustrates, the degenerate set $\mathcal{D}$ does not perturb the robot's motion.

## IV. EXPERIMENTS

This section describes the experiments performed with RHex in order to validate the algorithms developed in this
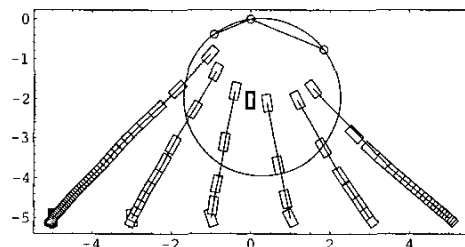


Fig. 7. Simulation of the non-holonomic constrained unicycle with a concave beacon configuration. The degenerate set $\mathcal{D}$ is represented by the large circle. The goal pose is represented by the gray rectangle.

paper. The robot's body frame contains 2 PC104 stacks, a motor drive board, a camera, accelerometers and a gyroscope. The first stack is equipped with i/o boards to read the motor encoder information, motor temperature, etc. It runs the robot's controller using a supervisor implemented with the RHexLib library [16], [17]. The 2nd stack, connected to a digital camera through a FireWire port, does all of the image processing. To implement the low level image processing functions a new fast vision library (SVision) was written. A discretized version of equation (7) is implemented. In order to accommodate for the input velocities allowed in the robot's walk mode the function $w_i$, defined in the appendix, is introduced. We then get the discretized equation of motion:

$$\xi_{k+1} = \xi_k + \Delta_k H_\xi \cdot W(H_\xi^T \cdot \mathbf{u}) \qquad (10)$$

where $W(x_1, x_2) = (w_1(x_1), w_2(x_2))$ with $w_i$ defined in the appendix, $H_\xi$ is the "non-holonomic projection matrix", $\Delta_k$ is a gain factor and $\mathbf{u}$ is the input velocity vector obtained in equation (7).

$$H_\xi = \begin{bmatrix} -\sin(\xi_\theta) & 0 \\ \cos(\xi_\theta) & 0 \\ 0 & 1 \end{bmatrix}$$

The robot is positioned approximately 2 meters away from a set of beacons and a "snapshot" is taken. The location of the beacons recorded in the snapshot's image plane is fed into the navigation function as the goal pose. The robot is then moved into different initial conditions and it is released as represented by the triangles in the right side of figures 8 and 9. In general it is not expected that the robot will get back to goal point, only to the apparent curve of equilibrium points suggested in the numerical simulations. Due to the differences between the presumed quasi-static unicycle model and RHex's true locomotion behaviour, some failures occurred as reported in the following table. A trial is considered a failure if the beacons leave the FOV of the robot's camera.

| experiment | failure rate |
|---|---|
| #1 | 5 out of 23 |
| #2 | 0 out 18 |

For experiment #1, illustrated in figure 8, a linear beacon configuration is used. Experiment #2 verifies the results obtained in simulation suggesting that it is safe to puncture the degenerate set $\mathcal{D}$: figure 9 shows that the robot successfully reaches a small neighborhood of the goal pose and it is not perturbed by the singularity $\mathcal{D}$ represented by the large circle. One can notice that in experiment #1 the number of failures is higher then experiment #2. This is due to the more careful selection of the scaling, saturation and dead zone parameters of function $w_i$, used in equation 10 and defined in the appendix.

## V. CONCLUSIONS

The experiments in section IV reveal that navigation using visual servoing can be accomplished for the 3 beacon algorithm. It is verified experimentally that the algorithm is fairly robust to parameter uncertainty. By taking a snapshot of
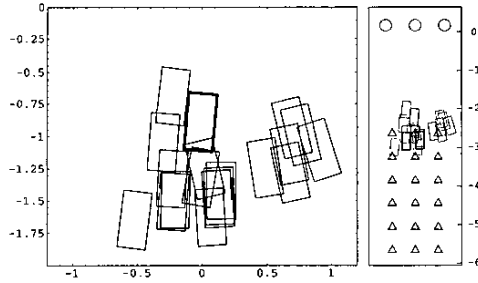
Fig. 8. Experiment #1. Linear beacon configuration. The picture on the right represents a view of the hall: the circles represent the beacons, the triangles represent initial conditions and the thick rectangle represents the goal pose. The picture on the left is a detailed view of the final pose for all the experiments. The units are in meters.
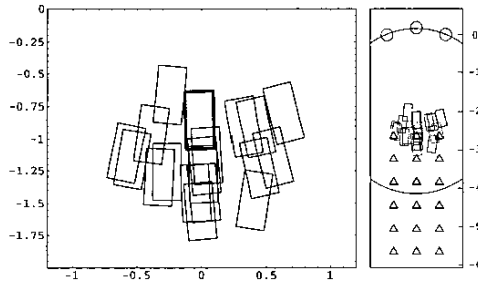


Fig. 9. Experiment #2. Linear beacon configuration. The picture on the right represents a view of the hall: the small circles represent the beacons, the large circle represents $\mathcal{D}$ the triangles represent initial conditions and the thick rectangle represents the goal pose. The picture on the left is a detailed view if the final pose for all the experiments. The units are in meters.

the goal location the parameter uncertainty error is implicitly subtracted during the trial run, as discussed at greater length in [1]. The simulations reported in section III qualitatively resemble the results obtained in the experiments and serve to validate the modeling assumptions used for the robot's dynamics. However, ultimately, the known limitations in the quasi-static model, built into the present version of these algorithms, limit the achievable accuracy.

### A. Future work

Naturally, the next step is to modify the 3 beacon algorithm to account for the non-holonomic model of the robot. Following that, a better dynamical model for RHex should be developed. In the longer term, we seek to replace the current bright red beacons with landmarks derived from natural elements of the scene, allowing the robot to use trees, rocks, and other objects to navigate the real world.

### VI. ACKNOWLEDGMENTS

### REFERENCES

[1] N. J. Cowan, J. D. Weingarten, and D..E. Koditschek, "Visual servoing via navigation functions," *Transactions on Robotics and Automation*, August 2002.

[2] U. Saranli, M. Buehler, and D. E. Koditschek, "Rhex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, July 2001.

[3] R. Altendorfer, N. Moore, H. Komsuoglu, M. Buehler, H. Brown Jr., D. McMordie, U. Saranli, R. J. Full, and D. Koditschek, "Rhex: A biologically inspired hexapod runner," *Autonomous Robots*, vol. 11, pp. 207–213, 2001.

[4] U. Saranli and D. E. Koditschek, "Template based control of hexapedal running," in *IEEE Int. Conf. on Robotics and Automation*, May 2003.

[5] R. W. Brockett, *Asymptotic stability and feedback stabilization*, ser. Differential Geometric Control Theory. Birkhauser, Boston: R. W. Brockett, R. S. Millman, and H. J. Sussmann, 1983, pp. 181–191.

[6] E. D. Sontag, "'a lyapunov-like" characterization of asymptotic controlability," *SIAM J. Contr. Opt*, vol. 21, pp. 462–471, 1983.

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

[8] P. I. Corke, "Visual control of robot manipulators - a review," 1994.

[9] C. J. Taylor and J. P. Ostrowski, "Robust visual servoing based on relative orientation," in *Int. Conf. on Robotics and Automation*, 1998.

[10] H. Zhang and J. P. Ostrowski, "Visual servoing with dynamics: Control of an unmanned blimp," in *Int. Conf. on Robotics and Automation*, 1999.

[11] E. Malis, F. Chaumette, and S. Boudet, "2-1/2-d visual servoing," *IEEE Transactions on Robotics and Automation*, pp. 238–250, 1999.

[12] N. J. Cowan, "Vision-based control via navigation functions," Ph.D. dissertation, University of Michigan, 2001.

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

[14] R. Ghigliazza, R. Altendorfer, P. Holmes, and D. Koditschek, "Passively stable conservative locomotion," *submitted to SIAM Journal of Applied Dynamical Systems*, 2002.

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

[16] U. Saranli and E. Klavins, "Rhexlib programmer's reference manual," University of Michigan, Tech. Rep.

[17] U. Saranli, "Simsect hybrid dynamical simulation environment," University of Michigan, Tech. Rep., 2000.

### APPENDIX

Define $\Phi$ as the map from local body coordinates in $SE(2)$ into $Q$:

$$\Phi(x_b, y_b, \theta_b) \quad : \quad SE(2) \to Q$$

$$(x_b, y_b, \theta_b) \mapsto \begin{bmatrix} \arctan(-x_b/y_b) \\ \theta_b - \arctan(-x_b/y_b) \\ \sqrt{x_b^2 + y_b^2} \end{bmatrix}$$

Define $\Upsilon$ as the map from world coordinates to body coordinates:

$$\Upsilon(x_w, y_w, \theta_w) \quad : \quad SE(2) \to SE(2)$$

$$(x_w, y_w, \theta_w) \mapsto - \begin{bmatrix} R_{\theta_w}^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ \theta_w \end{bmatrix}$$

Define the function $w_i(x)$ with saturation $\tau_i$, scaling $\kappa_i$ and dead zone $v_i$ as:

$$w_i(x) := \begin{cases} 0 & if \quad x \in [-v_i, v_i] \\ \max(\min(\kappa_i x, \tau_i), -\tau_i) & if \quad x \notin [-v_i, v_i] \end{cases}$$