in *Proc. 1993 IEEE Int. Conf. on Robotics and Automation*, pp. 3/418-3/423, 1993.

Increasing the Tracking Region of an Eye-in-Hand System by Singularity and Joint Limit Avoidance

Brad Nelson Pradeep K. Khosla

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3891

Abstract

A new control strategy is presented which visually tracks objects using a manipulator/camera system while simultaneously avoiding kinematic singularities and joint limits by moving in directions along which the tracking task space is unconstrained or redundant. A manipulability measure is introduced into the visual servoing objective function in order to derive the control law. The algorithms developed have been experimentally verified on an eye-in-hand system. Results demonstrate the effectiveness of the method by showing that the tracking region of a manipulator tracking objects with planar motion can be greatly increased.

1. Introduction

Robotic assembly typically requires precise calibration of the entire assembly workcell so that parts that are to be assembled can be placed at positions and orientations within thousandths of an inch of their desired position and orientation. An alternative to precise calibration is to use sensor feedback during the assembly process. Force feedback is sometimes incorporated into the assembly workcell, however force feedback can be difficult to use due to instabilities that arise when contact is made and due to the poor signal-to-noise ratio that force sensors tend to provide. In addition, the use of force feedback requires that parts be brought near one another before final parts mating occurs, thus the system must still be calibrated to a relatively high degree. Visual feedback can help overcome these problems, because a vision sensor is non-contact and can provide information on a much larger area of the workcell than a force sensor provides. The effective use of visual feedback combined with force feedback during the assembly process can be used to create robotic assembly workcells that have the ability to perform precision assemblies in imprecisely calibrated workcells. This can dramatically reduce the cost and setup time of robotic assembly systems.

An important component of a visually servoed assembly system is the visual tracking of objects during assembly. Visual tracking has several other areas of application as well, including the inspection, grasping, and assembly of parts on moving conveyors or in environments that are difficult to calibrate, and satellite docking in outer space. Although several researchers have studied visual tracking, for example, [2], [3], [4], [6], [8], [9], and [16], these visual servoing techniques are often impractical to use because of the limited workspace within which tracking can occur. These limits occur because of the presence of kinematic singularities throughout the workspace and the possibility of exceeding physical joint limits during manipulator motion.

When tracking a moving object with an eye-in-hand system or when visually servoing a manipulator using a static camera, it is necessary that robot motion commands be given in cartesian space. A well-known problem with controlling a manipulator in cartesian space occurs when the manipulator passes through or near a kinematic singularity, because cartesian based control algorithms employing the Jacobian inverse become numerically unstable and unrealizable at or near singularities. When visually tracking an object it is also important that the manipulator maintains a configuration that allows motion in all directions of possible object motion without requiring extremely large joint velocities from any actuator, because the future motion of the object is unknown or imprecisely known. This requires that the manipulator should not be near singularities, as well.

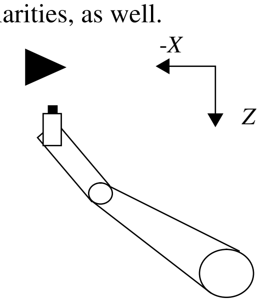

Figure 1. Overhead view of a visually tracking Scara manipulator near a singularity.

Consider the visual tracking system shown in Figure 1. As the object continues moving in the *-X* direction, the manipulator reaches an external singularity and the object is lost. Had the hand-eye system been initially started from a different configuration, the external singularity would have been reached either sooner, if the camera was initially placed nearer the object, or later, if the camera started farther from the object. However, if the system employs knowledge of the manipulator's configuration in order to improve the configuration using any redundancies that may exist, the tracking region can be greatly extended.

We propose a control strategy that uses a cartesian manipulability gradient which allows an eye-in-hand system to track objects with planar motion while avoiding singularities and joint limits continuously. This method improves the manipulator's configuration while tracking by using the redundancies that a six DOF manipulator provides when tracking in a lower dimensional task space. Our results indicate that the tracking region of an eye-in-hand system can be increased significantly. Section 2 of this paper describes modeling, control and feature tracking of the 2-D visual tracking eye-in-hand problem. Singularity and joint limit avoidance are described in Section 3, along with a discussion of the cartesian manipulability gradient. The implementation of the visual tracking system and experimental results which demonstrate the effectiveness of this method are described in Sections 4 and 5, followed by the conclusion.

2. Modeling and control

2.1. Modeling

The controlled active vision framework is used to model the eye-in-hand system [11]. In formulating the visual tracking model, a full 3D tracking model is derived, and simplifications to this model are then made for 2D tracking in order to compare different 2D tracking strategies. To model the 3-D visual tracking problem, we assume a pinhole model for the camera with a frame {C} placed at the focal point of the lens as shown in Figure 2. A feature on an object at $\boldsymbol{P}$ with coordinates (X_o, Y_o, Z_o) in the camera frame projects onto the camera's image plane at

$$x_i = \frac{fX_o}{Z_o s_x} + x_p \tag{1}$$

$$y_i = \frac{fY_o}{Z_o s_y} + y_p \tag{2}$$

where (x_i, y_i) are the image coordinates of the feature, f is the focal length of the lens, s_x and s_y are the horizontal and vertical dimensions of the pixels on the CCD array, and (x_p, y_p) is the piercing point of the optical axis on the CCD. In addition, it is assumed that $Z_o >> f$. This assumption holds because the maximum focal length of our camera is 16mm, while Z_o is larger than 1m.

Initially, let us assume that the camera moves in a static environment with a translational velocity $\boldsymbol{T}=[\dot{x}_c \;\; \dot{y}_c \;\; \dot{z}_c]^T$ and a rotational velocity $\boldsymbol{R}=[\omega_{xc} \;\; \omega_{yc} \;\; \omega_{zc}]^T$ with respect to the camera frame {C}. The velocity of point $\boldsymbol{P}$ in the camera frame {C} induced by camera motion is then

$$\frac{d\boldsymbol{P}}{dt} = -\boldsymbol{T} - \boldsymbol{R} \times \boldsymbol{P} \tag{3}$$

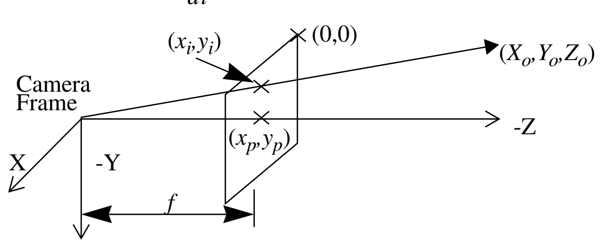

Figure 2. The pinhole camera model.

If we let $x_o=(x_i-x_p)$ and $y_o=(y_i-y_p)$ represent the projection of $\boldsymbol{P}$ on the image plane, then the velocity of the projection of $\boldsymbol{P}$ on the image plane ($\dot{x}_o, \dot{y}_o$) is equivalent to the camera induced optical flow of $\boldsymbol{P}$ which we represent by (u_c, v_c). By explicitly calculating $\dot{\boldsymbol{P}}$ from (3) and determining the projection of $\dot{\boldsymbol{P}}$ on the image plane using (1) and (2), the camera induced optical flow of point $\boldsymbol{P}$ is

$$u_c = -\frac{f\dot{x}_c}{Z_o s_x} + \frac{x_o \dot{z}_c}{Z_o} + \frac{x_o y_o s_y \omega_{xc}}{f} - \left(\frac{f}{s_x} + \frac{x_o^2 s_x}{f}\right)\omega_{yc} + \frac{y_o s_y}{s_x}\omega_{zc} \tag{4}$$

$$v_c = -\frac{f\dot{y}_c}{Z_o s_y} + \frac{y_o \dot{z}_c}{Z_o} + \left(\frac{f}{s_y} + \frac{y_o^2 s_y}{f}\right)\omega_{xc} - \frac{x_o y_o s_x \omega_{yc}}{f} - \frac{x_o s_x}{s_y}\omega_{zc} \tag{5}$$

The optical flow that can actually be observed on the image plane, however, is due to both the optical flow induced by camera motion (u_c, v_c) and the optical flow induced by object motion (u_o, v_o). The observed optical flow, which we represent by (u, v), can only be determined for successive image frames which are separated in time by a sampling period T, and can be written as

$$u(kT) = u_o(kT) + u_c(kT) \tag{6}$$

$$v(kT) = v_o(kT) + v_c(kT) \tag{7}$$

It can easily be seen that the observed optical flow between successive images can also be represented by

$$u(k) = \frac{x(k+1) - x(k)}{T} \tag{8}$$

$$v(k) = \frac{y(k+1) - y(k)}{T} \tag{9}$$

where we now write $x_o(k)$ and $y_o(k)$, which are the observed coordinates of the feature on the image plane with respect to the piercing point, as $x(k)$ and $y(k)$, respectively. Let $k=kT$ in order to simplify notation without any loss of generality.

If we substitute $u(k)$ and $v(k)$ in (8) and (9) with their equivalent expressions from (6) and (7), then (8) and (9) can be rewritten as

$$x(k+1) = x(k) + Tu_c(k) + Tu_o(k) \tag{10}$$

$$y(k+1) = y(k) + Tv_c(k) + Tv_o(k) \tag{11}$$

Equations (10) and (11) can be written in state-space form as

$$\boldsymbol{x_F}(k+1) = \boldsymbol{A_F}\boldsymbol{x_F}(k) + \boldsymbol{B_F}(k)\boldsymbol{u}(k) + \boldsymbol{E_F}\boldsymbol{d_F}(k) \tag{12}$$

where $\boldsymbol{A_F}=\mathbf{I}_2$, $\boldsymbol{E_F}=T\mathbf{I}_2$, $\boldsymbol{x_F}(k)\in R^2$, $\boldsymbol{d_F}(k)\in R^2$, and $\boldsymbol{u}(k)\in R^6$. The matrix $\boldsymbol{B_F}(k)\in R^{2x6}$ is

$$\boldsymbol{B_F}(k) = \begin{bmatrix} -\frac{f}{Z_o(k)s_x} & 0 & \frac{x(k)}{Z_o(k)} & \frac{x(k)y(k)s_y}{f} & -\left(\frac{f}{s_x} + \frac{x^2(k)s_x}{f}\right) & \frac{y(k)s_y}{s_x} \\ 0 & -\frac{f}{Z_o(k)s_y} & \frac{y(k)}{Z_o(k)} & \left(\frac{f}{s_y} + \frac{y^2(k)s_y}{f}\right) & -\frac{x(k)y(k)s_x}{f} & -\frac{x(k)s_x}{s_y} \end{bmatrix} \tag{13}$$

The vector $\boldsymbol{x_F}(k)=[x(k) \;\; y(k)]^T$ is the state vector, $\boldsymbol{u}(k)=[\dot{x}_c \;\; \dot{y}_c \;\; \dot{z}_c \;\; \omega_{xc}\omega_{yc}\omega_{zc}]^T$ is the vector representing possible control inputs, and $\boldsymbol{d_F}(k)=[u_o(k) \; v_o(k)]^T$ is the exogenous deterministic disturbances vector. The state vector $\boldsymbol{x_F}(k)$ is computed using the SSD algorithm to be described in Section 2.3.

Depending on the constraints placed on target motion and the objective of the visual tracking system, more than one feature may be required in order to achieve the system's goals. For example, for full 3D tracking in which it is desired to maintain a constant six degree of freedom transformation between the camera and the target, at least three features are required. To track an object constrained to move with motion in three dimensions, such as planar motion with rotations or 3D translational motion, at least two features are needed. For objects moving with only two degrees of freedom, such as planar motion without rotations, a single feature is sufficient. A generalized state equation for a variable number of features can be written as

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}(k)\boldsymbol{u}(k) + \boldsymbol{E}\boldsymbol{d}(k) \tag{14}$$

where M is the number of features required, $\boldsymbol{A}=\mathbf{I}_{2M}$, $\boldsymbol{E}=T\mathbf{I}_{2M}$, $\boldsymbol{x}(k)\in R^{2M}$, $\boldsymbol{d}(k)\in R^{2M}$, and $\boldsymbol{u}(k)\in R^{i}$ ($i\in\{1,2,3,4,5,6\}$, the number of axes along which tracking occurs). The matrix $\boldsymbol{B}(k)\in R^{2M\mathrm{x}i}$ is

$$\boldsymbol{B}(k) = \begin{bmatrix} \boldsymbol{B}_{\boldsymbol{F}}^{(1)}(k) \\ \cdots \\ \boldsymbol{B}_{\boldsymbol{F}}^{(M)}(k) \end{bmatrix} \tag{15}$$

The superscript (j) denotes each of the feature points ($j\in\{1,2,...,M\}$). Thus, the size of $\boldsymbol{B}$ is dependent on the number of non-zero cartesian control inputs and the number of features required, which the system designer determines based on task requirements. The vector $\boldsymbol{x}(k)=[x^{(1)}(k)\ y^{(1)}(k)\ldots x^{(M)}(k)\ y^{(M)}(k)]^{\mathrm{T}}$ is the new state vector, and $\boldsymbol{d}(k)=[u_o^{(1)}(k)\ v_o^{(1)}(k)\ldots u_o^{(M)}(k)\ v_o^{(M)}(k)]^{\mathrm{T}}$ is the new exogenous deterministic disturbances vector.

2.2. Control

The control objective of an eye-in-hand visual tracking system is to control camera motion in order to place the image plane coordinates of features on the target at some desired position, despite object motion. The desired image plane coordinates could be changing with time, or they could simply be the original coordinates at which the features appear when tracking begins. The control strategy used to achieve the control objective is based on the minimization of an objective function at each time instant. The objective function places a cost on differences in feature positions from desired positions, as well as a cost on providing control input, and is of the form

$$F(k+1) = [\boldsymbol{x}(k+1) - \boldsymbol{x}_D(k+1)]^{T}\boldsymbol{Q}\,[\boldsymbol{x}(k+1) - \boldsymbol{x}_D(k+1)] + \\ + \boldsymbol{u}^{T}(k)\boldsymbol{R}\boldsymbol{u}(k) \tag{16}$$

This expression is minimized with respect to the current control input $\boldsymbol{u}(k)$. In solving for $\boldsymbol{u}(k)$, (14) is used to replace $\boldsymbol{x}(k+1)$ in (16). The term $\boldsymbol{d}(k)$ in (14) is the optical flow induced by object motion and is considered a disturbance, thus it is ignored in the formulation. The derivative of F with respect to $\boldsymbol{u}(k)$ is then taken, set equal to zero, and solved for $\boldsymbol{u}(k)$. The end result yields the following expression for the control input

$$\boldsymbol{u}(k) = -\left(\boldsymbol{B}^{T}(k)\boldsymbol{Q}\boldsymbol{B}(k) + \boldsymbol{R}\right)^{-1}\boldsymbol{B}^{T}(k)\boldsymbol{Q}\,[\boldsymbol{x}(k) - \boldsymbol{x}_D(k+1)] \tag{18}$$

The weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ allow the user to place more or less emphasis on the feature error and the control input. Their selection effects the response and stability of the tracking system. The $\boldsymbol{Q}$ matrix must be positive definite, and $\boldsymbol{R}$ must be positive semi-definite for a bounded response. Although no standard procedure exists for choosing the elements of $\boldsymbol{Q}$ and $\boldsymbol{R}$, general guidelines can be found in [12].

2.3. Measurement of feature positions

The measurement of the motion of the features on the image plane must be done continuously and quickly. The method used to measure this motion is based on optical flow techniques and is a modification of the method proposed in [1]. This technique is known as Sum-of-Squares-Differences (SSD) optical flow, and is based on the assumption that the intensities around a feature point remain constant as that point moves across the image plane. A more complete description of the algorithm and its implementation can be found in [12].

For singularity/joint limit avoidance, the tracking manipulator may move such that the depth of the object from the camera changes. This slowly changes the size of the feature due to the camera-lens projective geometry. In order to account for this change, the feature template can be periodically updated by using the matched feature window from a recent image as the new feature template.

3. Singularity and joint limit avoidance

Although many researchers have proposed solutions to the singularity avoidance problem, all of these solutions have important drawbacks for the visual tracking algorithm presented in this paper. This is because, (a) the path of the object being tracked is not known a priori, so the manipulator's path may not be planned in advance to avoid singularities as proposed in [14], and (b) pseudo-inverse and singularity-robust inverse [10] or damped least-squares methods [15] of cartesian control allow motion to continue near and even at singularities by approximation of the cartesian path, but higher than usual joint velocities can still occur, and the loss of movement in one or more directions still exists. An efficient and effective method has been developed that keeps the manipulator as far from singularities as possible during visual tracking, while also avoiding motions which will bring the manipulator near joint limits.

In order to avoid singularities and joint limits, a measure must be determined in order to compare different manipulator configurations so that the best configuration can be found. Several researchers have proposed several different manipulability measures. The measure most commonly used is proposed by Yoshikawa [17] and is

$$w(\boldsymbol{q}) = \sqrt{det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^{T})} \tag{19}$$

where $\boldsymbol{J}(\boldsymbol{q})$ is the manipulator Jacobian matrix.

Because a measure is desired which indicates nearness to joint limits, as well as nearness to kinematic singularities, a variation of the measure first proposed in [13] is used. This measure combines a penalty function which goes to zero in the vicinity of joint limits, with Yoshikawa's measure which indicates nearness to singular configurations. The original form of this measure is

$$w(\boldsymbol{q}) = \left(1 - e^{-k\prod_{i=1}^{n}\frac{(q_i - q_{imin})\,(q_{imax} - q_i)}{(q_{imax} - q_{imin})^2}}\right)\sqrt{det\,(\boldsymbol{J}\,(\boldsymbol{q})\,\boldsymbol{J}^T\,(\boldsymbol{q})} \qquad (20)$$

where k is a user defined constant, n is the number of joints, q_i is the ith joint angle, and q_{imin} and q_{imax} are the minimum and maximum allowable joint values, respectively, for the ith joint. The experimental eye-in-hand system uses a non-redundant manipulator, therefore,

$$\sqrt{det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)} = \sqrt{(det(\boldsymbol{J}(\boldsymbol{q})))^2} = |det(\boldsymbol{J}(\boldsymbol{q}))| \qquad (21)$$

A continuously differentiable function is desired so a modified form of (20), which also incorporates (21), is used. The final form of the manipulability measure used to aid in visual tracking is

$$w'(\boldsymbol{q}) = \left(1 - e^{-k\prod_{i=1}^{n}\frac{(q_i - q_{imin})\,(q_{imax} - q_i)}{(q_{imax} - q_{imin})^2}}\right)(det\,(\boldsymbol{J}\,(\boldsymbol{q})))^2 \qquad (22)$$

A variety of methods can be used for integrating the manipulability measure into the visual tracking strategy. An effective technique is to introduce the manipulability measure into the visual tracking objective function given in (16), resulting in the new objective function

$$F(k+1) = [\boldsymbol{x}(k+1) - \boldsymbol{x}_D(k+1)]^t\boldsymbol{Q}\,[\boldsymbol{x}(k+1) - \boldsymbol{x}_D(k+1)] + $$
$$+ \boldsymbol{u}^T(k)\boldsymbol{R}\boldsymbol{u}(k) + \frac{S}{w'(\boldsymbol{q}(k))} \qquad (23)$$

where S is a constant representing the relative cost of maintaining a manipulator configuration with a low manipulability. Minimizing this objective function with respect to the control input $\boldsymbol{u}(k)$, results in a control law of the form

$$\boldsymbol{u}\,(k) = -\left(\boldsymbol{B}^T\,(k)\,\boldsymbol{Q}\boldsymbol{B}\,(k) + \boldsymbol{R}\right)^{-1}\times$$
$$\times\left[\boldsymbol{B}^T\,(k)\,\boldsymbol{Q}\,[\boldsymbol{x}\,(k) - \boldsymbol{x}_D\,(k+1)\,] + \frac{S}{2}\frac{\partial}{\partial\boldsymbol{u}(k)}\left(\frac{1}{w'(\boldsymbol{q})}\right)\right] \qquad (24)$$

Rewriting the partial derivative of the reciprocal of the manipulability measure in terms of the gradient of the manipulability measure, the following can be calculated

$$\frac{\partial}{\partial\boldsymbol{u}(k)}\left(\frac{1}{w'(\boldsymbol{q})}\right) = \left(\nabla_{\boldsymbol{u}}\frac{1}{w'(\boldsymbol{q})}\right)^T = \frac{-1}{w'(\boldsymbol{q})^2}\,(\nabla_{\boldsymbol{u}}w'(\boldsymbol{q}))^T \qquad (25)$$

which results in the control law

$$\boldsymbol{u}\,(k) = -\left(\boldsymbol{B}^T\,(k)\,\boldsymbol{Q}\boldsymbol{B}\,(k) + \boldsymbol{R}\right)^{-1}\times$$
$$\times\left[\boldsymbol{B}^T\,(k)\,\boldsymbol{Q}\,[\boldsymbol{x}\,(k) - \boldsymbol{x}_D\,(k+1)\,] - \frac{S}{2w'(\boldsymbol{q})^2}\nabla^T_{\boldsymbol{u}}w'(\boldsymbol{q})\right] \qquad (26)$$

This provides an elegant and robust technique for allowing the system to achieve its primary goal of visual tracking, while simultaneously avoiding the threat of failure due to kinematic singularities or the violation of joint limits.

The cartesian manipulability gradient can be approximated numerically by determining the difference between the current manipulability and the manipulability of the robot if it is moved incrementally along each of the six cartesian axes. For example, the x component of the manipulability gradient can be calculated by the following

$$\frac{\partial}{\partial x}w' \approx \frac{w'(\boldsymbol{q} + \boldsymbol{J}^{-1}(\boldsymbol{q})\delta_{\boldsymbol{x}}) - w'(\boldsymbol{q})}{|\delta_{\boldsymbol{x}}|} \qquad (27)$$

where $\delta_{\boldsymbol{x}}$=[0.001m 0 0 0 0 0]^T. The other five components of the gradient can be calculated in a similar fashion.

4. Implementation

The visual tracking algorithm described previously has been implemented on a robotic assembly system consisting of three Puma560's called the Rapid Assembly System. One of the Pumas has a Sony XC-77RR camera mounted at its end-effector. The camera is connected to an Innovision IDAS/150 Vision System. The Pumas are controlled from a VME bus with two Ironics IV-3230 (68030 CPU) processors, an IV-3220 (68020 CPU) which also communicates with a trackball, a Mercury floating point processor, and a Xycom parallel I/O board communicating with three Lord force sensors mounted on the Pumas' wrists. All processors on the controller VME run the Chimera3 real-time operating system. The system controller architecture is shown in Figure 3.

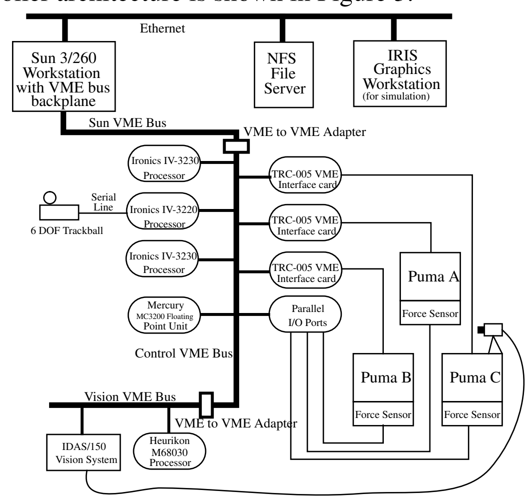

Figure 3. The Rapid Assembly System controller architecture.

The vision system VME communicates with the controller VME using BIT3 VME-to-VME adapters. The Innovision IDAS/150 Vision System calculates the optical flow of the features and the manipulator cartesian control inputs for object tracking using the control law given by (26). A special floating point array processor on the IDAS/150 is used to calculate the optical flow of the feature, and a Heurikon 68030 board, also on the vision system, computes the control inputs. An image can be grabbed, displacements found for two features on an object, and control inputs along the tracking axes calculated at 7Hz for X-Y-Θ object motion and 10Hz for X-Y object motion.

5. Experimental results

Experimental results demonstrate the effectiveness of these algorithms. In this section, the capabilities of translational tracking of an object moving in X and Y without using any singularity or joint/limit avoidance is first shown. Next, the tracking capabilities of a redundant four DOF tracker (tracking along and about X and Y) of two DOF object motion that

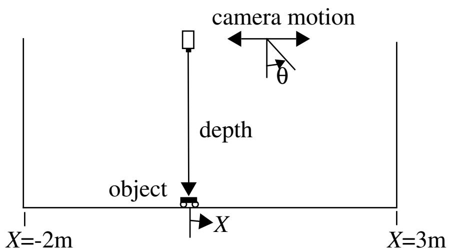

Figure 4. Experimental setup.

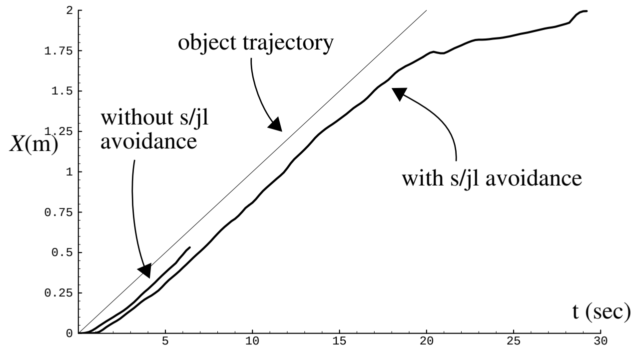

Figure 5. Camera and object translation along +*X* versus time.

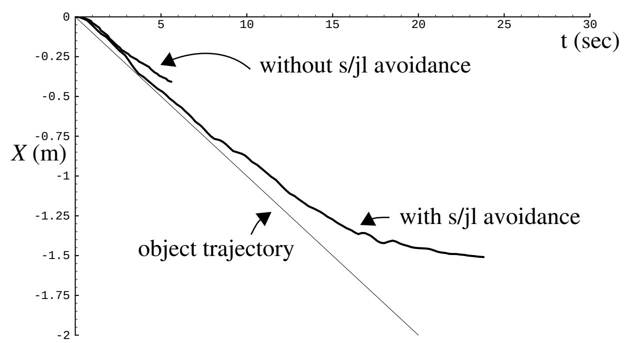

Figure 6. Camera and object translation along -*X* versus time.

uses two of its tracking axes for singularity/joint limit avoidance is demonstrated. Figure 4 illustrates the experimental setup used to collect the results.

5.1. Two axis tracking of 2D motion without singularity/joint limit avoidance

Figures 5 and 6 show the translation of the object and the camera versus time. The first tracking strategy described is two-axis tracking without avoidance of singularities and joint limits. Although the object has a total range of linear motion of 5m, the eye-in-hand system is able to track the object for only 1.04m. The limitations on the tracking region for this case are due to both singularities and joint limits. Figure 7 is a plot of the manipulability measure $w'(\boldsymbol{q})$ (22) of the Puma versus X and shows that at X=.532m a singularity is reached. This occurs when the second and third joints of the Puma become aligned. The dashed line in the figure represents the square of Yoshikawa's manipulability measure (21), and shows that this term dominates the modified measure in the +X direction. In the -X direction, the distance to the nearest singularity actu-

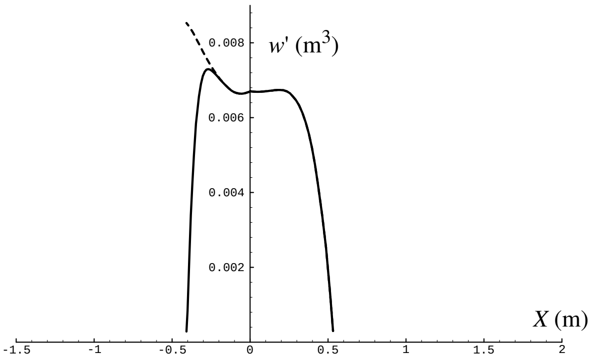

Figure 7. Manipulability versus *X* for two-axis visual tracking without singularity or joint limit avoidance.

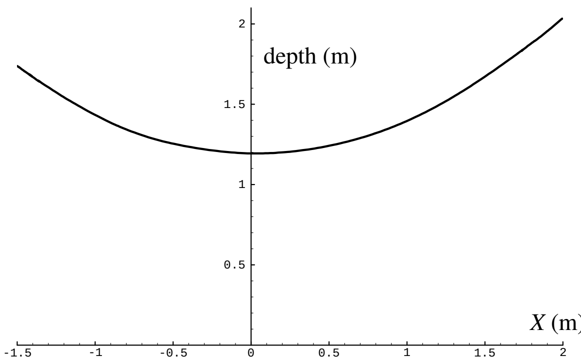

Figure 8. Depth of the object versus *X* for four-axis visual tracking with singularity and joint limit avoidance.

ally increases, however, the first joint reaches a limit on its allowable range after the object travels only 0.403m. The penalty function dominates the modified manipulability measure $w'(\boldsymbol{q})$ and causes this measure to quickly descend to zero.

5.2. Four axis tracking of 2D motion with singularity/ joint limit avoidance

In this example, redundant tracking motion is allowed in order to decrease the translational motion requirements of the camera. Tracking is allowed along and *about* X and Y. Singularity avoidance is also allowed along X and Y, so that as the manipulator nears joint limits or singularities, tracking along X and Y is inhibited and tracking about these axes increases in magnitude in order to keep the feature at its desired coordinate on the image plane. Figures 5 and 6 show the translation of the object and the estimated translation of the object based on camera motion according to the equation

$$X_{est} = X + \frac{Z}{\cos\theta} \tag{28}$$

where X is the translation of the camera, Z is the height of the camera above the plane of object motion, and θ is the rotation of the camera about its Y axis. The two figures show that the estimate of the position of the object becomes less accurate as the object moves farther away from the manipulator. This is due to the increasing depth of the object from the camera at these object positions, as illustrated by Figure 8. At greater depths, errors in camera calibration become more pronounced, the location of the features on the object become less precise, and, hence, the estimation of the object's location

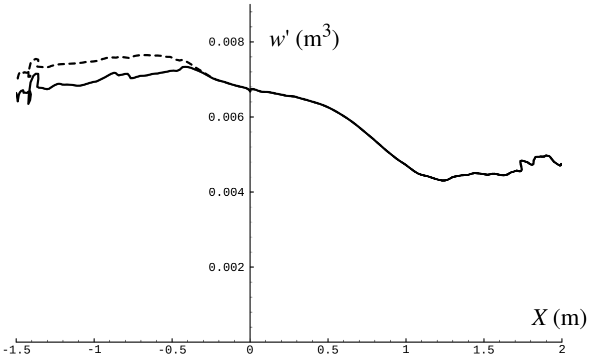

Figure 9. Manipulability versus *X* for four-axis visual tracking with singularity and joint limit avoidance.

becomes less reliable. Since the objective of the system is to track the object, the system still operates successfully at these extreme distances, and, as the object moves back towards the manipulator, the accuracy of the estimated world position of the object increases.

Figure 9 shows that the manipulability of the Puma remains high as the object is tracked over the entire 5m length of object motion. As with the example given in Section 5.1, the manipulator has the potential to enter singular regions of the workspace or to exceed joint limits. By employing the cartesian manipulability gradient to guide the arm away from these regions, redundant camera motion can be used to avoid these regions while simultaneously tracking the object, and the modified manipulability measure $w'(\boldsymbol{q})$ (22) remains relatively high. The joint limit penalty function tends to decrease in the same region as it did during two-axis tracking without singularity or joint limit avoidance, however, redundant tracking motion is used to keep this function from going to zero.

Figures 5 and 6 show that the object is tracked by the eye-in-hand system over the object's entire linear range of motion. Upon experimentation, it was determined that the failure modes of the system occur when the object moves such that the base of the manipulator occludes the object, or the depth of the object becomes so great that the feature cannot be reliably detected. Both of these cases are extreme.

6. Conclusions

We have demonstrated that singularity and joint limit avoidance using a cartesian gradient of manipulability which accounts for nearness to joint limits as well as singularities is an effective method of significantly extending the tracking region of an eye-in-hand visual tracking system. By introducing a modified measure of manipulability into the control objective function of the visual tracker, an elegant control law is derived which allows a system designer to determine, based on design requirements, which axes are to be used for tracking and which are to be used for avoiding singularities and joint limits. Experimental results comparing two different tracking strategies are presented. These results demonstrate that a tracking system that accounts for the configuration of the manipulator during tracking allows object motion which is far less constrained than traditional eye-in-hand tracking systems have previously allowed.

7. Acknowledgments

This research was supported by the U.S. Army Research Office through Grant Number DAAL03-91-G-0272 and by Sandia National Laboratories through Contract Number AC-3752D. We would like to thank Nik Papanikolopoulos for his suggestions concerning this work.

8. References

[1] P. Anandan, "Measuring Visual Motion from Image Sequences," Technical Report COINS-TR-87-21, COINS Department, University of Massachusetts, 1987.

[2] F. Chaumette, P. Rives, and B. Espiau, "Positioning of a Robot with respect to an Object, Tracking it, and Estimating its Velocity by Visual Servoing," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 2248-2253, April, 1991.

[3] P.I. Corke, "Video-Rate Visual Servoing for Robots," in *Lecture Notes in Control and Information Science*, eds. V. Hayward and O. Khatib, pp. 429-451, Springer-Verlag, 1989.

[4] J.T. Feddema and C.S.G. Lee, "Adaptive Image Feature Prediction and Control for Visual Tracking with a Hand-Eye Coordinated Camera," *IEEE Trans. on Systems, Man, and Cybernetics*, 20(5), pp. 1172-1183,1990.

[5] K.D. Gremban, C.E. Thorpe, and T. Kanade, "Geometric Camera Calibration using Systems of Linear Equations," in *Proc. of Image Understanding Workshop*, pp. 820-825, April, 1988.

[6] K. Hashimoto, T. Ebine, and H. Kimura, "Dynamic Visual Feedback Control for a Hand-Eye Manipulator," in *Proc. of the 1992 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS-92)*, pp. 1863-1868, July, 1992.

[7] B.K.P. Horn, *Robot Vision*, MIT Press, Cambridge, MA, 1986.

[8] J. Ishikawa, K. Kosuge, and K. Furuta, "Intelligent Control of Assembling Robot Using Vision Sensor," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1904-1909, April, 1990.

[9] A.J. Koivo and N. Houshangi, "Real-Time Vision Feedback for Servoing of a Robotic Manipulator with Self-Tuning Controller," in *IEEE Trans. on Systems, Man, and Cybernetics,* 21(1), pp. 134-142,1991.

[10] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions with Singularity Robustness for Robot Manipulator Control," in *Journal of Dynamic Systems, Measurement and Control*, 108(3), pp. 163-171, 1986.

[11] N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade, "Adaptive Robotic Visual Tracking," in *Proc. of the American Control Conference*, pp. 962-967, June 1991.

[12] N.P. Papanikolopoulos, B. Nelson, and P.K. Khosla, "Full 3-D Tracking Using the Controlled Active Vision Paradigm," in *Proc. of the 1992 IEEE Int. Symp. on Intelligent Control (ISIC-92)*, August 11-13, 1992.

[13] M.J. Tsai, *Workspace Geometric Characterization of Industrial Robot*, Ph.D. Thesis, Ohio State University, Department of Mechanical Engineering, 1986.

[14] M. Uchiyama, K. Shimizu, and K. Hakomori, "Performance Evaluation of Manipulators using the Jacobian and its Application to Trajectory Planning," in *Robotics Research 2*, eds. H. Hanafusa and H. Inoue, pp. 447-454, Cambridge, MA: MIT Press, 1985.

[15] C. W. Wampler and L. J. Leifer, "Applications of Damped Least-Squares Methods to Resolved-Rate and Resolved Acceleration Control of Manipulators," in *Journal of Dynamic Systems, Measurement and Control*, 110(1), pp. 31-38, 1988.

[16] L.E. Weiss, A.C. Sanderson, and C.P. Neuman, "Dynamic Sensor-Based Control of Robots with Visual Feedback," in *IEEE Journal of Robotics and Automation* RA-3(5), pp. 404-417, October, 1987.

[17] T. Yoshikawa, "Manipulability of Robotic Mechanisms," in *Robotics Research 2*, eds. H. Hanafusa and H. Inoue, pp. 439-446, Cambridge, MA: MIT Press, 1985.