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

Bradley J. Nelson

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15213-3891

Pradeep K. Khosla

Department of Electrical and Computer Engineering and
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15213-3891

Abstract

An eye-in-hand system visually tracking objects can fail when the manipulator encounters a kinematic singularity or a joint limit. A solution to this problem is presented in which objects are visually tracked while the manipulator simultaneously avoids kinematic singularities and manipulator joint limits by moving in directions along which the tracking task space is unconstrained or redundant. A manipulability measure is introduced into the visual tracking objective function, providing an elegant and robust technique for deriving a control law that visually tracks objects while accounting for the configuration of the manipulator. Two different tracking strategies, one using a standard visual tracking strategy and the other using the newly proposed strategy, are experimentally compared on an actual hand/eye system. The experimental results demonstrate the effectiveness of the new method by showing that the tracking region of a manipulator tracking objects with planar motion can be greatly increased.

To appear in the *International Journal of Robotics Research*

1. Introduction

Robotic assembly 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 manipulation. Force feedback is sometimes incorporated into assembly workcells, 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 part mating occurs, thus the system must still be well calibrated. 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 manipulation can be used to create robotic manipulation systems that have the ability to perform precise manipulation tasks within imprecisely calibrated and dynamically varying environments.

An important component of a visually guided manipulation system is the visual tracking of objects with an eye-in-hand system during manipulation. 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; satellite docking in outer space, autonomous vehicle navigation, etc. Although several researchers have studied eye-in-hand visual servoing, for example (Weiss, Sanderson and Neumann 1987; Allen 1989; Corke and Paul 1989; Feddema and Lee 1990; Papanikolopoulos, Khosla and Kanade 1991; Espiau, Chaumette and Rives 1992; Hashimoto and Kimura 1993; Wilson 1993), these proposed visual servoing techniques can be quite difficult to use in practice because of the limited workspace with which objects can be tracked. These limits occur because of the presence of kinematic singularities throughout the workspace and the possibility of exceeding physical joint limits during manipulator motion. For the manipulators with which we have performed eye-in-hand visual servoing, the CMU DDArmII, a Puma 560, and an Adept2, these regions are extremely limited and can be difficult to determine.

When a manipulator encounters a kinematic singularity during cartesian motion, or when the cartesian controller commands a motion which will exceed a joint limit, the visual servoing system will fail. When visually tracking an object with an eye-in-hand system, 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 either unknown or imprecisely known. This also requires that the manipu-

lator should not be near singularities. Typically, when visually tracking objects with an eye-in-hand system it is necessary to constrain the allowable tracking region to regions of the workspace where there is no danger that the manipulator passes near kinematic singularities or joint limits. This often places extreme limitations on the trackable workspace. Consider the visual tracking system shown in Figure 1. If a constant transformation between the camera and object is desired and the object continues to move in the -X direction, the manipulator will soon reach a singularity and the system will fail. 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 and a joint limit penalty function that allows an eye-in-hand system to track objects while simultaneously avoiding kinematic singularities and joint limits continuously. This method improves the manipulator's configuration while tracking by using the redundancies with respect to the task 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 the modeling, control, and feature tracking of the visual tracking eye-in-hand system. Singularity and joint limit avoidance are described in Section 3, along with a discussion of the cartesian manipulability gradient. A description of the implementation of the visual tracking system followed by experimental results which demonstrate the effectiveness of this method are described in Sections 4 and 5, followed by the conclusion.

2. Visual Tracking

2.1. Modeling

The controlled active vision framework is used to model the eye-in-hand system (Papanikolopoulos, Khosla and Kanade 1991). 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 illustrate and 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 (Gremban, Thorpe and Kanade 1988). A feature on an object at $\boldsymbol{P}$[1] with coordinates (X_o,Y_o,Z_o) in the camera frame projects onto the camera's image plane by the equations for

1. Bold symbols denote vectors or matrices.

perspective projection at

$$x = \frac{fX_o}{Z_o s_x} \tag{1}$$

$$y = \frac{fY_o}{Z_o s_y} \tag{2}$$

in which it is assumed that $Z_o >> f$, where f is the focal length of the lens. This assumption holds because the focal length of our camera is 16mm, while Z_o is typically larger than 1m. In (1) and (2), s_x and s_y are the horizontal and vertical dimensions of the pixels on the CCD array, respectively.

Any displacement of a rigid object can be described by a rotation about an axis and a translation. If the angle of this rotation is small, the rotation can be characterized by three independent rotations about the X, Y and Z axes. Initially, assume that the camera moves in a static environment with a translational velocity $\boldsymbol{T}=[\dot{x}_c\ \dot{y}_c\ \dot{z}_c]^{\mathrm{T}}$ and a rotational velocity $\boldsymbol{R}=[\omega_{xc}\ \omega_{yc}\ \omega_{zc}]^{\mathrm{T}}$ with respect to the camera frame {C}. The velocity of point $\boldsymbol{P}$ in the camera frame {C} induced by camera motion is then

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

The velocity of the projection of $\boldsymbol{P}$ on the image plane $(\dot{x}, \dot{y})$ 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 the point $\boldsymbol{P}$ can be determined to be, assuming for the moment that the object is stationary, (Horn 1986)

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

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

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 is represented 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 $k=kT$ without loss of generality. If $u(k)$ and $v(k)$ are substituted in (6) and (7) with their equivalent expressions from (8) and (9), then (6) and (7) 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) represent the state equations for the eye-in-hand tracking system and 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[2] $\boldsymbol{A_F}=\mathbf{I}_2$, $\boldsymbol{E_F}=T\mathbf{I}_2$, $\boldsymbol{x_F}(k)\in R^2$, $\boldsymbol{d_F}(k)\in R^2$, and $\boldsymbol{u}(k)\in R^6$. The matrix $\boldsymbol{B_F}(k)\in R^{2x6}$ is

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

The vector $\boldsymbol{x_F}(k)=[x(k)\ y(k)]^{\mathrm{T}}$ is the state vector and is computed using the SSD algorithm to be described in Section 2.3. The vector $\boldsymbol{u}(k)=[\dot{x}_c\ \dot{y}_c\ \dot{z}_c\omega_{xc}\omega_{yc}\omega_{zc}]^{\mathrm{T}}$ represents possible control inputs, and $\boldsymbol{d_F}(k)=[u_o(k)\ v_o(k)]^{\mathrm{T}}$ is the exogenous deterministic disturbances vector. The elements of $\boldsymbol{d_F}$ are due to optical flow terms that are induced by unknown object motion. It is possible to use predictive techniques based on recently observed object motion to estimate $\boldsymbol{d_F}$, as in (Kalata 1992). It is common in regulator theory, however, to simply assume that unknown signals to be tracked are considered disturbances to be suppressed. Past work has shown that this is a reasonable assumption for the visual tracking problem (Papanikolopoulos, Khosla and Kanade 1991).

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 (Papanikolopoulos and

2. The symbol $\mathbf{I}_n$ denotes the identity matrix of order n.

Khosla 1993). To track an object constrained to move with motion in only 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 $\boldsymbol{B}(k)\in R^{2M\text{x}i}$ 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 ($j\in\{1,2,...,M\}$). The vector $\boldsymbol{x}(k)=[x^{(1)}(k)\ y^{(1)}(k)...\ x^{(M)}(k)\ y^{(M)}(k)]^{\mathrm{T}}$ is the new state vector, and $\boldsymbol{d}(k)=[u_o{}^{(1)}(k)\ v_o{}^{(1)}(k)...\ u_o{}^{(M)}(k)\ v_o{}^{(M)}(k)]^{\mathrm{T}}$ is the new exogenous deterministic disturbances vector.

The system model just presented can be extended to account for system delays, modeling and control inaccuracies, and measurement noise. See (Papanikolopoulos, Nelson and Khosla 1992) for a detailed explanation of how this can be accomplished.

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

$$\boldsymbol{u}(k) = -\left(\boldsymbol{B}^T(k)\boldsymbol{Q}\boldsymbol{B}(k) + \boldsymbol{R} \right)^{-1} \boldsymbol{B}^T(k)\boldsymbol{Q} \left[\boldsymbol{x}(k) - \boldsymbol{x}_D(k+1) \right] \tag{17}$$

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 stability and response of the tracking system. The $\boldsymbol{Q}$ matrix must be positive semi-definite, and $\boldsymbol{R}$ must be positive 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 (Papanikolopoulos, Nelson and Khosla 1992).

A coupling exists between the number of degrees of freedom with which the object moves in the camera frame, the number of axes of camera motion that must be commanded by the control input, and the number of features that must be tracked on the moving object. In general, the number of features needed per object must be at least half the number of degrees of freedom with which the object is allowed to move in the camera frame. The number of axes along or about which the camera is allowed to move, i.e. the number of potentially non-zero control inputs, must be at least as great as the number of degrees of freedom of object motion. Analysis of the control law (17) shows why this is true. For the error in the state vector to map to a single control input vector that will drive the error to zero, the number of states must be at least as great as the number of desired control inputs. In order to track an object with n degrees of freedom, the camera must move with n degrees of freedom if a constant transformation between object and camera is desired. Thus, in general, the number of states must be at least as great as the number of desired control inputs. It is also necessary that the matrix expression

$$\left(\boldsymbol{B}^T(k)\boldsymbol{Q}\boldsymbol{B}(k) + \boldsymbol{R} \right)^{-1} \boldsymbol{B}^T(k)\boldsymbol{Q} \tag{18}$$

is of full rank. In (Papanikolopoulos and Khosla 1993), conditions which can cause (18) to lose rank due to particular feature locations on the image plane are discussed.

The axes of camera motion which are controlled in order to track the object can vary depending on the requirements of the task. If it is important that the transformation between the object and the camera remains constant, then the camera must move along and about axes parallel to the degrees of freedom of allowed object motion. In order to reduce the error in feature positions due to object motion, it is advantageous to control axes of the camera which are well conditioned with respect to the axes of object motion. An analysis of the eigenvalues and eigenvectors of $\boldsymbol{B}^T\boldsymbol{B}$ indicate these directions. This is because the eigenvectors of $\boldsymbol{B}^T\boldsymbol{B}$ result in a set of basis vectors for the row space of $\boldsymbol{B}$, which describe the vector space of $\Re(\boldsymbol{B}^T)$, the range of $\boldsymbol{B}^T$. Directions which

contain large eigenvalues are directions in which camera motion induces a relatively large optical flow on the image plane. These are good directions to use for tracking object motion, since camera motion can more easily compensate for object motion in these directions. For example, for an object that can move in the Y direction (in the camera frame), camera motion along Y or about X is well conditioned to drive the state error to zero when the object moves along Y. Motion along or about Z could also be used to track object motion along Y due to the projective transformation exhibited by the camera, but inspection of the eigenvalues and eigenvectors of $\boldsymbol{B}^T\boldsymbol{B}$ shows that camera motion along these axes is poorly conditioned with respect to object motion along Y. Thus, convergence of feature errors would be slow if these axes of camera motion were used for tracking. The eigenvalues of $\boldsymbol{B}^T\boldsymbol{B}$ along Z or about any of the three axes are mainly dependent on the coordinates of the features on the image plane. For robust tracking, axes of motion whose eigenvalues are mainly dependent on image coordinates are poor choices for the tracking strategy. In (Nelson and Khosla 1994), a tracking measure called *resolvability* is introduced which provides a technique for measuring the relative ability of different camera-lens-manipulator configurations to precisely visually servo objects held by a manipulator. This same measure can be used for eye-in-hand systems.

Previous eye-in-hand systems that track objects with planar motion have not used redundant tracking motion. For example, an object with 2D planar motion can be tracked by camera motion along X-Y, along X-Y-Θ_X-Θ_Y, or some other combination of axes, provided that the $\boldsymbol{B}$ matrix representing the strategy is well conditioned along X and Y. The former strategy keeps the transformation from the camera to the object constant, while the later strategy results in a varying depth of the object in the camera frame during tracking. The advantage of the latter strategy is that the burden of tracking is more evenly distributed among the manipulator's joints, thus allowing for potentially faster object motion before exceeding manipulator joint velocity limits. Another advantage of this strategy is that the tracking axes are redundant, which allows motion along these axes to be used for more than one objective.

The control law given by (17) applies to any eye-in-hand tracking strategy, though different strategies use different matrices in the equation. For 2D tracking using camera motion along X and Y, $\boldsymbol{B}$ is reduced to a 2x2 matrix. $\boldsymbol{Q}$ is also 2x2, since only a single feature, thus two states, is needed, and $\boldsymbol{R}$ is a 2x2 diagonal matrix representing the cost of cartesian control input along both X and Y. For 2D tracking using X-Y-Θ_X-Θ_Y camera motion, $\boldsymbol{B}$ is 2x4, $\boldsymbol{Q}$ for the single feature is still 2x2, however, $\boldsymbol{R}$ must be a 4x4 diagonal matrix and allows the user to place a cost effecting relative motion along all four axes of possible camera motion. If the elements of $\boldsymbol{R}$ which govern the rotational axes are chosen to be large with respect to the elements representing translation cost, then the four-axis tracking system performs the same as the two-axis system performs.

2.3. Feature Tracking

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 (Anandan 1987). This technique is known as a 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. The displacement of a point $\mathbf{p}_a=(x,y)$ at the next time increment to $\mathbf{p}_{a'}=(x+\Delta x, y+\Delta y)$, is determined by finding the displacement $\Delta\boldsymbol{x}=(\Delta x,\Delta y)$ which minimizes the SSD measure

$$e(\boldsymbol{p}_a, \Delta\boldsymbol{x}) = \sum_W [I_a(x+i, y+j) - I_{a'}(x+i+\Delta x, y+j+\Delta y)]^2 \quad (19)$$

where I_a and $I_{a'}$ are the intensity functions from two successive images and W is the window centered about the feature point which makes up the feature template. For the algorithm implemented, W is 16x16 pixels, and possible displacements of up to $\Delta x=\Delta y=\pm 32$ pixels are considered. Features on the object that are to be tracked can be selected by the user, or a feature selecting algorithm can be invoked. Features with strong intensity gradients in perpendicular directions, such as corners, are typically the best features to select.

In order to decrease the search space, a pyramidal search scheme (Figure 3) has been implemented which first searches a coarse resolution of the image that has 1/16 the area of the original image, using a feature template in which a W that is originally 32x32 is averaged to 8x8. After determining where the feature is in the coarse image, a finer resolution image that is 1/4 the original spatial resolution is searched with an original W of 16x16 which is averaged to 8x8 in an area centered about the location of the minimum SSD measure found in the coarse image. Finally, the full resolution image and the 16x16 feature template are used to pinpoint the location of the displaced feature.

The pyramidal scheme reduces the time required for the computation of the SSD algorithm by a factor of five for a single feature over the method of computing the feature locations at the full resolution alone. However, reliability can be sacrificed when the selected feature loses its tracking properties (strong perpendicular intensity gradients) at the coarser image resolutions. Since the search scheme first estimates where the feature is located based on the coarse image, it is critical that good features at coarse resolutions are tracked. When a user selects features, it is often not obvious that a particular feature may lose its tracking characteristics at coarse resolutions. Because of this, an automatic feature selector has been implemented based on (Tomasi and Kanade 1991) which accounts for the different levels of resolution in the pyramidal search

scheme.

Depending on the tracking strategy chosen, the depth of the object from the camera may change in order to maximize the distance the manipulator is from singularities and joint limits. This slowly changes the size of the feature template based on the projection equations. 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

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. There are two different types of singularities that a manipulator may encounter. An internal singularity occurs when two axes of the manipulator become aligned within the boundaries of the manipulator's workspace, and an external singularity occurs when the manipulator reaches the boundaries of its workspace. When a manipulator encounters either of these singularities, or when the cartesian controller commands a motion which will exceed a joint limit, the visual tracking system will fail. 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 also requires that the manipulator should not be near singularities.

Although many researchers have proposed solutions to the singularity avoidance problem, all of these solutions have drawbacks when applied to the visual tracking problem. 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 (Uchiyama, Shimizu and Hakomori 1985), and (b) pseudo-inverse and singularity-robust inverse (Nakamura and Hanafusa 1986) or damped least-squares methods (Wampler 1986) of cartesian control allow motion to continue near and even at singularities by approximation of the cartesian path, but the loss of movement in one or more directions still exists and higher than usual joint velocities may occur. In (Egeland, Ebdrup and Chiaverini 1990), the damped least-squares method proposed by both (Nakamura and Hanafusa 1986) and (Wampler 1986) is extended to create a cartesian controller for visual servoing that operates well in the presence of internal singularities. The authors use a damping factor

derived from an estimate of the smallest singular value of the manipulator Jacobian. Experimental results show that extremely good tracking accuracy can be maintained despite the presence of internal kinematic singularities, and joint velocities can be kept relatively low. Two important limitations of this method are that objects which move beyond external singularities cannot be tracked, and joint limits are not considered. The remainder of this section describes an efficient and effective method that uses kinematic redundancies to keep manipulators from approaching external as well as internal singularities while visually tracking. In addition, manipulator motions which will bring the manipulator near joint limits are also avoided.

3.1. Measures of Manipulability

In order to avoid singularities and joint limits, a measure must be determined in order to compare different manipulator configurations. Several researchers have proposed several different manipulability measures. One of the first measures proposed was used to evaluate the working region of a manipulator wrist (Paul and Stevenson 1983), which was considered to be all wrist configurations satisfying

$$w(\boldsymbol{q}) \;=\; |det(\boldsymbol{J}(\boldsymbol{q}))| \geq 0.5 \tag{20}$$

where $\boldsymbol{J}$ is the Jacobian of the wrist and $\boldsymbol{q}$ is the vector of joint angles. The most commonly referred to measure was proposed by (Yoshikawa 1985) and is

$$w(\boldsymbol{q}) \;=\; \sqrt{det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)} \tag{21}$$

A variance on this measure defines a “transmissibility measure” for force control (Ghosal and Roth 1988) as

$$w(\boldsymbol{q}) \;=\; \sqrt{det(\boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{J}(\boldsymbol{q}))} \tag{22}$$

A measure used to compare manipulators of different sizes, different numbers of joints, and which operate across differently dimensioned task spaces (Kim and Khosla 1991) is

$$w(\boldsymbol{q}) \;=\; \frac{\sqrt[m]{det\left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\right)}}{\sqrt{\sum_{i=1}^{n}\left(a_i^2 + d_i^2\right)}} \tag{23}$$

where m is the dimension of the task space, n is the number of links, and a_i and d_i are the Denavit-Hartenberg parameters of each link of the manipulator.

Because a measure is desired which indicates nearness to joint limits, as well as nearness to kine-

matic singularities, a variation of the measure first proposed in (Tsai 1986) is used. This measure combines a penalty function which approaches zero in the vicinity of joint limits, with Yoshikawa's measure which indicates nearness to singular configurations. The original form of this measure is

$$w(\boldsymbol{q}) = \left(1 - e^{-k \prod_{i=1}^{n} \frac{(q_i - q_{imin})\,(q_{imax} - q_i)}{(q_{imax} - q_{imin})^2}} \right) \sqrt{det\,(\boldsymbol{J}\,(\boldsymbol{q})\,\boldsymbol{J}^T\,(\boldsymbol{q}))} \tag{24}$$

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,

$$\sqrt{det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)} = \sqrt{(det(\boldsymbol{J}(\boldsymbol{q})))^2} = |det(\boldsymbol{J}(\boldsymbol{q}))| \tag{25}$$

A continuously differentiable function is desired so a modified form of (24), which also incorporates (25), is used. The final form of the manipulability measure used to aid in visual tracking is

$$w'(\boldsymbol{q}) = \left(1 - e^{-k \prod_{i=1}^{n} \frac{(q_i - q_{imin})\,(q_{imax} - q_i)}{(q_{imax} - q_{imin})^2}} \right) (det\,(\boldsymbol{J}\,(\boldsymbol{q}))\,)^2 \tag{26}$$

Figure 4 illustrates the measure for a Puma 560 by varying the fifth joint between its joint limits while keeping the other five joints stationary. When the fifth joint angle becomes zero the fourth and sixth joint axes become aligned, and an internal singularity is reached in which a degree of freedom of end-effector motion is lost. Yoshikawa's measure correctly indicates the distance of the manipulator from this singular condition, but the measure also indicates that the manipulator is optimally configured at $\pm 90^o$, which is near the joint limits. By using the penalty function, the modified manipulability measure ensures that motion towards the joint limits will be penalized.

The envelope of the penalty function can be easily varied by changing the value of the constant k in the penalty function

$$P = \left(1 - e^{-k \prod_{i=1}^{n} \frac{(q_i - q_{imin})\,(q_{imax} - q_i)}{(q_{imax} - q_{imin})^2}} \right) \tag{27}$$

Figure 5 shows the envelope created by different values of k as the fifth joint of a Puma 560 varies between its joint limits.

3.2. Augmenting the Control Objective Function

A variety of methods could be used for integrating the manipulability measure into the visual tracking strategy. For example, (Nelson and Khosla 1992) presents a hybrid control structure which uses an unconstrained tracking axis for singularity avoidance. If one uses the task function approach for visual servoing (Espiau, Chaumette and Rives 1992), the manipulability measure presented in (26) could be introduced as a secondary task function, also resulting in a hybrid strategy. An effective and more general technique is to introduce the manipulability measure into the visual tracking objective function given in (16), resulting in the new objective function

$$F(k+1) = [\boldsymbol{x}(k+1) - \boldsymbol{x}_D(k+1)]^T \boldsymbol{Q} [\boldsymbol{x}(k+1) - \boldsymbol{x}_D(k+1)] + \boldsymbol{u}^T(k)\boldsymbol{R}\boldsymbol{u}(k) + \frac{S}{w'(\boldsymbol{q}(k))} \quad (28)$$

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 (see the appendix for the derivation of (29) from (28))

$$\boldsymbol{u}(k) = -\left(\boldsymbol{B}^T(k)\boldsymbol{Q}\boldsymbol{B}(k) + \boldsymbol{R} \right)^{-1} \left[\boldsymbol{B}^T(k)\boldsymbol{Q}[\boldsymbol{x}(k) - \boldsymbol{x}_D(k+1)] - \frac{S}{2w'(\boldsymbol{q})^2} \nabla_{\boldsymbol{u}} w'(\boldsymbol{q}) \right] \quad (29)$$

This provides an elegant and robust technique for allowing the system to achieve its primary goal of visual tracking, while simultaneously allowing the system to operate without the threat of system failure due to kinematic singularities or the violation of joint limits. This formulation allows the hand/eye system geometric characteristics to influence the magnitude of end-effector velocities commanded to improve the manipulator configuration. For example, if the depth of the object from the camera becomes large, control inputs of smaller magnitudes will be able to successfully track the object, due to the projective geometry of the camera-lens system. This is reflected in the term $\left(\boldsymbol{B}^T(k)\boldsymbol{Q}\boldsymbol{B}(k) + \boldsymbol{R} \right)^{-1}$, which will cause correspondingly small camera velocities for singularity/joint limit avoidance. Smaller velocities for singularity/joint limit avoidance will allow the manipulator to approach nearer to poor configurations. This is reasonable since an object moving far from the camera will not induce camera tracking velocities as large as if the object is close. Thus, the manipulator can come closer to poor configurations without the threat of entering a singular configuration or exceeding a joint limit, since the required tracking velocity of the manipulator is expected to have a smaller magnitude. If the object is near the camera, small object velocities require larger tracking velocities, making it important for the manipulator to maintain larger distances from singularities and joint limits during operation.

If axes of cartesian motion used for avoiding singularities and joint limits are not properly chosen, as discussed in Section 2.2, it may not be possible for the hand-eye system to successfully track

objects while avoiding singularities and joint limits. These situations occur when the same axes are used for avoiding singularities and joint limits and for tracking, and no redundant tracking motion is allowed. For these cases, the system responds by tracking object motion until the manipulability measure becomes low enough that the term

$$\frac{S}{2w'(\boldsymbol{q})^2}\nabla_{\boldsymbol{u}}w'(\boldsymbol{q}) \tag{30}$$

completely dominates the control law (29). The control law will not allow the manipulator to enter singular configurations or exceed joint limits, but the system will not track the object either. This type of tracking failure is almost always preferable to failures caused by singularities or joint limits which may damage the manipulator.

3.3. Calculating the Cartesian Manipulability Gradient

The term $\nabla_{\boldsymbol{u}}w'(\boldsymbol{q})$ appearing in the control law (29) represents the local gradient of manipulability in end-effector velocity space. The actual value of this term represents an end-effector velocity vector that, if held constant over the next time interval, will move the end-effector in a direction which maximally increases the manipulability of the manipulator configuration based on the local manipulability surface. This local 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_c velocity component of the manipulability gradient can be calculated by the following

$$\frac{\partial}{\partial \dot{x}_c}w' \approx \frac{w'(\boldsymbol{q} + \boldsymbol{J}^{-1}(\boldsymbol{q})\delta_{\boldsymbol{x}}) - w'(\boldsymbol{q})}{|\delta_{\boldsymbol{x}}|}T \tag{31}$$

where $\delta_{\boldsymbol{x}}=[0.001\text{m } 0\ 0\ 0\ 0\ 0]^{\mathrm{T}}$. The other five components of the gradient can be calculated in a similar fashion. The result is a vector which gives the cartesian direction in which the manipulator should move in order to increase manipulability maximally.

4. Hardware Implementation

The visual tracking algorithms previously described have been implemented on a robotic assembly system consisting of three Puma 560's called the Troikabot. One of the Pumas has a Sony XC-77RR camera with a 16mm lens mounted at its end-effector. The camera is connected to a Datacube Maxtower Vision System.The Pumas are controlled from a VME bus with two Ironics IV-3230 (68030 CPU) processors, an IV-3220 (68020 CPU) processor 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 Chimera 3.0 reconfigurable real-time operating system (Stewart, Schmitz and Khosla 1992). A diagram of the hardware setup is given in Figure 6. The vision system VME communicates with the controller VME using BIT3 VME-to-VME adapters. The Datacube Maxtower Vision System calculates the optical flow of the features using the SSD algorithm discussed in Section 2.3. A special high performance floating- point processor on the Datacube is used to calculate the optical flow of the feature, and a 68030 board, also on the vision system, computes the control inputs. An image can be grabbed and displacements for up to five 16x16 features in the scene can be determined at 30Hz.

The cartesian manipulability gradient as well as the cartesian control inputs are calculated on one of the control VME's IV-3230 boards. The control input for tracking and singularity/joint limit avoidance is sent to the Mercury floating point processor at 30Hz, where a cartesian controller calculates the proper joint control commands.

A second Puma 560 can be used as a target. The eye-in-hand Puma has no prior knowledge of the target Puma's path other than that the object being tracked on the target Puma moves in a plane approximately parallel to the eye-in-hand Puma's tool frame. When testing singularity/joint limit avoidance strategies, however, the range of motion of the second Puma is usually not sufficient to fully exercise the tracking capabilities of the eye-in-hand system (Nelson and Khosla 1992), so targets are sometimes placed on small mobile robots that move on the floor.

5. Experimental Results

Eye-in-hand object tracking can be performed using several different strategies. Two common types of visual tracking which are useful for observing objects with planar translational motion include pure translational tracking and translational with rotational tracking. The advantage of pure translational tracking is that it maintains a constant transformation from camera to object. On the other hand, tracking which uses two translational and two rotational axes allows for faster object motion, though the depth of the object with respect to the camera changes, and the orientation of the optical axis of the camera with the plane of object motion is not constant. More importantly, four axis tracking provides redundancies which can be used to avoid singularities and joint limits and extend tracking regions.

In this section, we present experimental results from different singularity/joint limit avoidance strategies. The capabilities of translational tracking of an object moving in X and Y without using

any type of singularity or joint/limit avoidance is first shown. Next, the tracking capabilities of a four axis tracker of two axis object motion that uses two of its tracking axes for singularity/joint limit avoidance is demonstrated. Figure 7 illustrates the experimental setup in its initial configuration for the experimental results presented.

5.1. Two Axis Tracking of 2D Motion without Singularity/Joint Limit Avoidance

Figures 8 and 9 show the translation of the object and the camera versus time. Three tracking strategies are shown in the figures. The first strategy used 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 just 0.94m. The limitations on the tracking region for this case are due to both singularities and joint limits. Figure 10 is a plot of the manipulability measure $w'(\boldsymbol{q})$ (26) of the Puma versus x and shows that at x=0.53m a singularity is reached. This occurs when joints 2 and 3 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 actually increases, however, joint 1 reaches a limit on its allowable range after the object travels only 0.40m. The penalty function (27) dominates the modified manipulability measure $w'(\boldsymbol{q})$, as shown in Figure 11, and causes the modified measure to quickly descend to zero.

5.2. Four Axis Tracking of 2D Motion with and without Singularity/Joint Limit Avoidance

In this experiment, the constraints on tracking are reduced in order to decrease the translational motion requirement of the camera. Tracking is allowed along and about X and Y. For one trial, 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 8 and 9 show the translation of the object and the estimated translation of the object based on camera motion according to the equation

$$x_{est} = x + \frac{z}{\cos\theta} \tag{32}$$

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. For the trial without singularity and joint limit avoidance, the object can be tracked approximately twice as far as when only two-axis

tracking is allowed, since approximately half the target motion is tracked by translational motion and half by rotational motion. When singularity and joint limit avoidance are enabled, the tracking region is extended until the object being tracked reaches the boundaries of the lab.

For four-axis tracking with singularity and joint limit avoidance, the manipulability of the Puma remains high as the object is tracked over the entire length of object motion, as Figure 12 illustrates. 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 $w'(\boldsymbol{q})$ (26) remains relatively high. Figure 13 shows that 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.

For all three tracking strategies shown by Figures 8 and 9, the same control law (29) was used with different gains for each strategy. The cost on feature error was the same for all three strategies with a value $\boldsymbol{Q}$=diag($2.0\text{x}10^{-6}$, $2.0\text{x}10^{-6}$). For two-axis tracking without singularity or joint limit avoidance, $\boldsymbol{R}$=diag(10.0,10.0,10^6,10^6) and S=0.0. For four-axis tracking without singularity and joint limit avoidance, $\boldsymbol{R}$=diag(10.0,10.0,10.0,10.0) and S=0.0, and with singularity and joint limit avoidance, $\boldsymbol{R}$=diag(10.0,10.0,10.0,10.0) and S=0.001. For each $\boldsymbol{R}$, $\boldsymbol{Q}$ can be increased to $\boldsymbol{Q}$=diag($2.7\text{x}10^{-6}$, $2.7\text{x}10^{-6}$) until the system reached a state of instability. For $\boldsymbol{Q}$=diag($2.0\text{x}10^{-6}$, $2.0\text{x}10^{-6}$), elements of $\boldsymbol{R}$ below 6.0 result in unstable systems. The value of S depends mainly on the maximum speed with which objects being tracked may move. If fast moving objects are to be tracked, higher values of S should be used. Smaller values of S allow the manipulator to approach nearer singularities and joint limits.

When the object reaches the extreme boundaries of the lab, Figures 8 and 9 show that the estimate of the position of the object becomes less accurate. This is due to the increasing depth of the object from the camera at these object positions, as Figure 14 illustrates. At greater depths, errors in camera calibration become more pronounced. There are several ways in which tracking precision could have been improved. For the visual servoing experiments performed, camera and lens manufacturer specifications were used in $\boldsymbol{B}$ in the control law (29). If the camera-lens system was precisely calibrated, tracking precision could have been improved. A more complex sensor model would also result in more precise estimates of object motion. The error could be further reduced by sub-pixel fitting techniques, and larger feature templates would allow correspondingly more precise object location estimates. With the exception of precisely calibrating the camera and lens, these techniques for improving the tracking precision of the system would incur an increase in

computational complexity, thus reducing the sampling rate of the system and decreasing the speed at which objects could be tracked. Another source of error in estimating object position is from the imprecisely calibrated tracking manipulator. The estimate of object position is determined from (32), which assumes that the translation and rotation of the camera can be accurately determined from manipulator kinematics. The kinematic parameters of the tracking Puma used were obtained from the literature, rather than from precise calibration of the actual Puma used in the experiments. As the depth of the object increases, kinematic calibration errors could significantly impact the estimated depth obtained from (32).

Since the objective of the system is to track the object, the system still operates successfully at extreme distances as long as features being tracked are not lost. Figure 15 illustrates this by showing the tracking error in pixels versus time for two-axis tracking and four-axis tracking with singularity and joint limit avoidance. The trials shown correspond to those in Figure 9. For two-axis tracking, errors are suppressed more slowly than when four tracking axes are used. This is as expected, since the manipulator uses more degrees of freedom for four-axis tracking. After approximately five seconds, a joint limit is exceeded for the two-axis case and the system fails. For the four-axis trial, errors are suppressed more quickly. Although the target travels at an approximately constant speed of 10cm/sec, the steady-state error continues to decrease because less control effort is required to suppress tracking errors on the image plane due to the increasing distance of the target from the camera. Eventually, the target reaches the boundaries of the laboratory, and the error is reduced to zero.

Figures 8 and 9 show that the object is tracked by the eye-in-hand system over the object's entire linear range of motion using only two translational axes for singularity and joint limit avoidance. This represents an approximate ten fold increase in the linear tracking region over the two-axis tracking system. In fact, the increase is even greater, but limitations were placed on the range of motion of the object due to the 5m width of the Advanced Manipulators Laboratory at Carnegie Mellon. Upon experimentation, it was discovered that the failure modes of the system tracking planar motion occurred 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. For planar object motion, two translational axes for singularity and joint limit avoidance are sufficient, because of the dimensionality of the tracking space. If this technique is extended to tracking objects with 3D motion, situations may arise in which local minima are reached on the surface of the objective function. These situations could create unavoidable singularities or joint limits if tracking and avoidance axes are not properly chosen.

6. Conclusion

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 user to determine, based on the system 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. Appendix

This appendix presents the derivation of the visual tracking control law that incorporates a manipulability measure representing nearness to singularities and joint limits. The goal is to find an expression for the control input $\boldsymbol{u}(k)$ at the current time instant which minimizes the objective function at the next time instant. The objective function is given by

$$F(k+1) = [\boldsymbol{x}(k+1) - \boldsymbol{x}_D(k+1)]^T \boldsymbol{Q} [\boldsymbol{x}(k+1) - \boldsymbol{x}_D(k+1)] + \boldsymbol{u}^T(k)\boldsymbol{R}\boldsymbol{u}(k) + \frac{S}{w'(\boldsymbol{q}(k))} \tag{33}$$

From the state equation (14), the expression

$$\boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \boldsymbol{B}(k)\boldsymbol{u}(k) \tag{34}$$

is obtained and used to rewrite the objective function as

$$F(k+1) = [\boldsymbol{x}(k) + \boldsymbol{B}(k)\boldsymbol{u}(k) - \boldsymbol{x}_D(k+1)]^T \boldsymbol{Q} [\boldsymbol{x}(k) + \boldsymbol{B}(k)\boldsymbol{u}(k) - \boldsymbol{x}_D(k+1)] + \boldsymbol{u}^T(k)\boldsymbol{R}\boldsymbol{u}(k) + \frac{S}{w'(\boldsymbol{q}(k))} \tag{35}$$

If it is assumed that $\boldsymbol{B}(k)$, which represents the Jacobian mapping from 3D cartesian camera space to 2D image space, is slowly varying, the partial derivative of $F(k+1)$ with respect to $\boldsymbol{u}(k)$ can be written as

$$\frac{\partial}{\partial \boldsymbol{u}(k)} F(k+1) = 2\boldsymbol{B}^T(k)\boldsymbol{Q} [\boldsymbol{x}(k) - \boldsymbol{x}_D(k+1)] + 2\boldsymbol{B}^T(k)\boldsymbol{Q}\boldsymbol{B}(k)\boldsymbol{u}(k) + 2\boldsymbol{R}\boldsymbol{u}(k) + S\frac{\partial}{\partial \boldsymbol{u}(k)}\left(\frac{1}{w'(\boldsymbol{q})}\right)^T \tag{36}$$

The minimum of $F(k+1)$ is found by rewriting this expression and setting it equal to 0

$$0 = \boldsymbol{B}^T(k)\boldsymbol{Q} [\boldsymbol{x}(k) - \boldsymbol{x}_D(k+1)] + (\boldsymbol{B}^T(k)\boldsymbol{Q}\boldsymbol{B}(k) + \boldsymbol{R})\,\boldsymbol{u}(k) + S\frac{\partial}{\partial \boldsymbol{u}(k)}\left(\frac{1}{w'(\boldsymbol{q})}\right)^T \tag{37}$$

Solving for $\boldsymbol{u}(k)$ results in the control law

$$\boldsymbol{u}(k) = -(\boldsymbol{B}^T(k)\boldsymbol{Q}\boldsymbol{B}(k) + \boldsymbol{R})^{-1}\left[\boldsymbol{B}^T(k)\boldsymbol{Q}[\boldsymbol{x}(k) - \boldsymbol{x}_D(k+1)] + \frac{S}{2}\left(\frac{\partial}{\partial \boldsymbol{u}(k)}\left(\frac{1}{w'(\boldsymbol{q})}\right)\right)^T\right] \tag{38}$$

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

$$\frac{\partial}{\partial \boldsymbol{u}(k)}\left(\frac{1}{w'(\boldsymbol{q})}\right) = \left(\nabla_{\boldsymbol{u}}\frac{1}{w'(\boldsymbol{q})}\right)^T = \frac{-1}{w'(\boldsymbol{q})^2}(\nabla_{\boldsymbol{u}}w'(\boldsymbol{q}))^T \tag{39}$$

The final form of the control law is

$$\boldsymbol{u}(k) = -(\boldsymbol{B}^T(k)\boldsymbol{Q}\boldsymbol{B}(k) + \boldsymbol{R})^{-1}\left[\boldsymbol{B}^T(k)\boldsymbol{Q}[\boldsymbol{x}(k) - \boldsymbol{x}_D(k+1)] - \frac{S}{2w'(\boldsymbol{q})^2}\nabla_{\boldsymbol{u}}w'(\boldsymbol{q})\right] \tag{40}$$

8. 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. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies. The authors would like to thank Nikos Papanikolopoulos and anonymous reviewers for their suggestions concerning this work, and Dana Bookey for proofreading the manuscript.

9. References

Allen, P.K. 1989. Real-time motion tracking using spatio-temporal filters. *Proc. Image Understanding Workshop*. San Mateo, CA:Morgan Kaufmann, pp. 695-701.

Anandan, P. 1987. Measuring visual motion from image sequences. Tech. Rept. COINS-TR-87-21, Amherst, Mass.:University of Massachusetts COINS Department.

Corke, P.I. and Paul, R.P. 1989. Video-rate visual servoing for robots. *Lecture notes in control and information science*. eds. V. Hayward and O. Khatib. London: Springer-Verlag, pp. 429-451.

Egeland, O., Ebdrup, M. and Chiaverini S. 1990. Sensory control in singular configurations-applications to visual servoing. *Proc. IEEE Int. Workshop on Intelligent Motion Control*. New York:-IEEE, pp. 401-405.

Espiau, B., Chaumette, F. and Rives, P. 1992. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*. 8(3):313-326.

Feddema, J.T. and Lee, C.S.G. 1990. Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera. *IEEE Trans. on Systems, Man, and Cybernetics*,

20(5):1172-1183.

Ghosal, A. and Roth, B. 1988. A new approach for kinematic resolution of redundancy. *Int. J. of Robotics Res.* 7(2):22-35.

Gremban, K.D., Thorpe, C.E. and Kanade, T. 1988. Geometric camera calibration using systems of linear equations. *Proc. of Image Understanding Workshop*. San Mateo, CA: Morgan Kaufmann, pp. 820-825.

Hashimoto, K. and Kimura, H. 1993. LQ optimal and nonlinear approaches to visual servoing. *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, ed. K. Hashimoto. London:World Scientific, pp. 165-198.

Horn, B.K.P. 1986. *Robot Vision.* Cambridge, MA: MIT Press.

Kalata, P.R. 1992. Alpha-beta target tracking systems: a survey. *Proc. 1992 American Controls Conf.* Evanston, IL.:American Autom. Control Council, pp. 832-836.

Kim, J.O. and Khosla, P.K. 1991. Dexterity measures for design and control of manipulators. *Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems (IROS91)*, pp. 758-763.

Nakamura, Y and Hanafusa, H. 1986. Inverse kinematic solutions with singularity robustness for robot manipulator control. *J. of Dyn. Sys., Measurement and Control*, 108(3):163-171.

Nelson, B. and Khosla, P.K. 1992. Increasing the tracking region of an eye-in-hand robotic system using controlled active vision. *Proc. 1992 IEEE Conf. on Control Applications*. New York:IEEE, pp. 1128-1133.

Nelson, B.J. and Khosla, P.K. 1994. The resolvability ellipsoid for visual servoing. *Proc. IEEE Comp. Soc. Conf. on Comp. Vision and Patter Recognition (CVPR94)*. New York:IEEE, pp.829-832.

Papanikolopoulos, N.P., Khosla, P.K. and Kanade, T. 1991. Adaptive robotic visual tracking. *Proc. of the American Control Conference*. Evanston, IL.:American Autom. Control Council, pp. 962-967.

Papanikolopoulos, N.P., Nelson, B. and Khosla, P.K. 1992. Full 3-d tracking using the controlled active vision paradigm. *Proc. 1992 IEEE Int. Symp. on Intelligent Control (ISIC-92)*. New York:-IEEE, pp. 267-274.

Papanikolopoulos N.P., and Khosla, P.K. 1993. Selection of features and evaluation of visual measurements for 3-d robotic visual tracking. *Proc. 8th Int. Symp. Intelligent Control*. New York:-IEEE, pp. 320-325.

Paul, R.P. and Stevenson, C.C. 1983. Kinematics of robot wrists. *Int. J. of Robotics Res.* 2(1):31-38.

Stewart, D.B., Schmitz, D.E. and Khosla, P.K. The Chimera II real-time operating system for

advanced sensor-based control systems. *IEEE Trans. Sys., Man Cyber.* 22(6):1282-1295.

Tomasi, C. and Kanade, T. 1991. Detection and tracking of point features. Tech. Rept. CMU-CS-91-132. Pittsburgh:Carnegie Mellon University School of Computer Science.

Tsai, M.J. 1986. Workspace Geometric Characterization of Industrial Robot. Ph.D. thesis, Ohio State University Department of Mechanical Engineering.

Uchiyama, M., Shimizu, K. and Hakomori, K. 1985. Performance evaluation of manipulators using the jacobian and its application to trajectory planning. *Robotics Research 2*. eds. H. Hanafusa and H. Inoue. Cambridge, MA: MIT Press, pp. 447-454.

Wampler, C. W. 1986. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Trans. Sys. Man Cyber.* 16(1):93-101.

Weiss, L.E., Sanderson, A.C. and Neuman, C.P. 1987. Dynamic sensor-based control of robots with visual feedback. *IEEE J. of Robotics and Automation* RA-3(5):404-417.

Wilson, W.J. 1993. Visual servo control of robots using kalman filter estimates of robot pose relative to work-pieces. *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, ed. K. Hashimoto. London:World Scientific, pp. 71-104.

Yoshikawa, T. 1985. Manipulability of robotic mechanisms. *Robotics Research 2*, eds. H. Hanafusa and H. Inoue. Cambridge, MA: MIT Press, pp. 439-446.

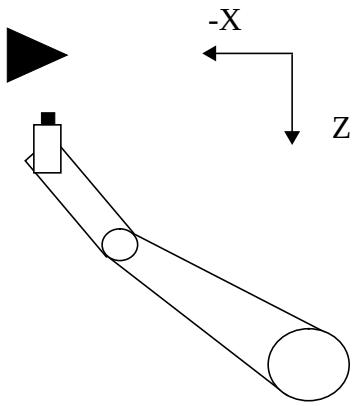

Figure 1. Overhead view of a SCARA-type manipulator visually tracking. If a constant transformation between the camera and object is desired and the object continues to move in the -X direction, the manipulator will soon reach a singularity and the system will fail.

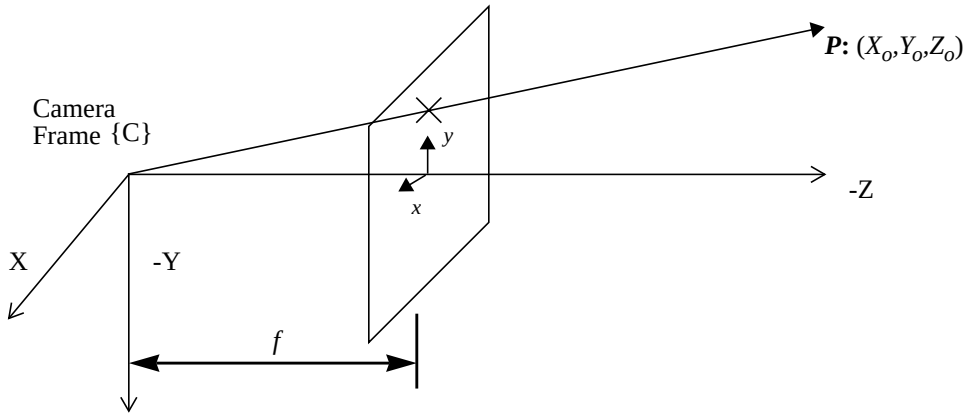

Figure 2. The pinhole camera model with the image plane moved in front of the camera to simplify signs in the equations.

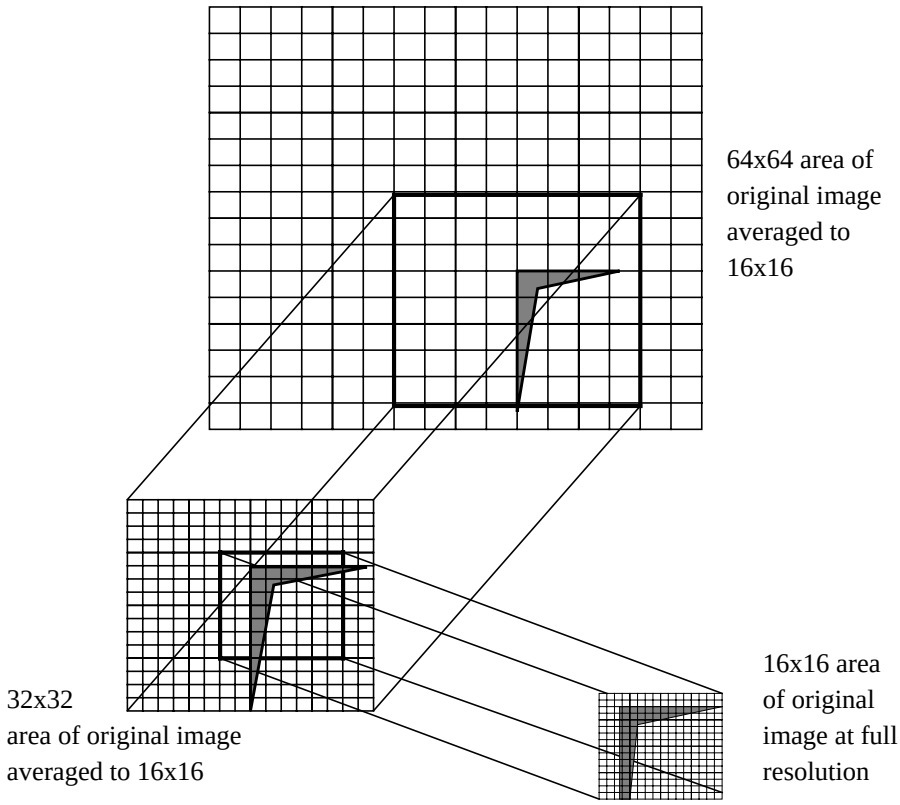

Figure 3. A pyramidal search scheme is used in the SSD optical flow algorithm in order to increase the overall sampling rate of the system.

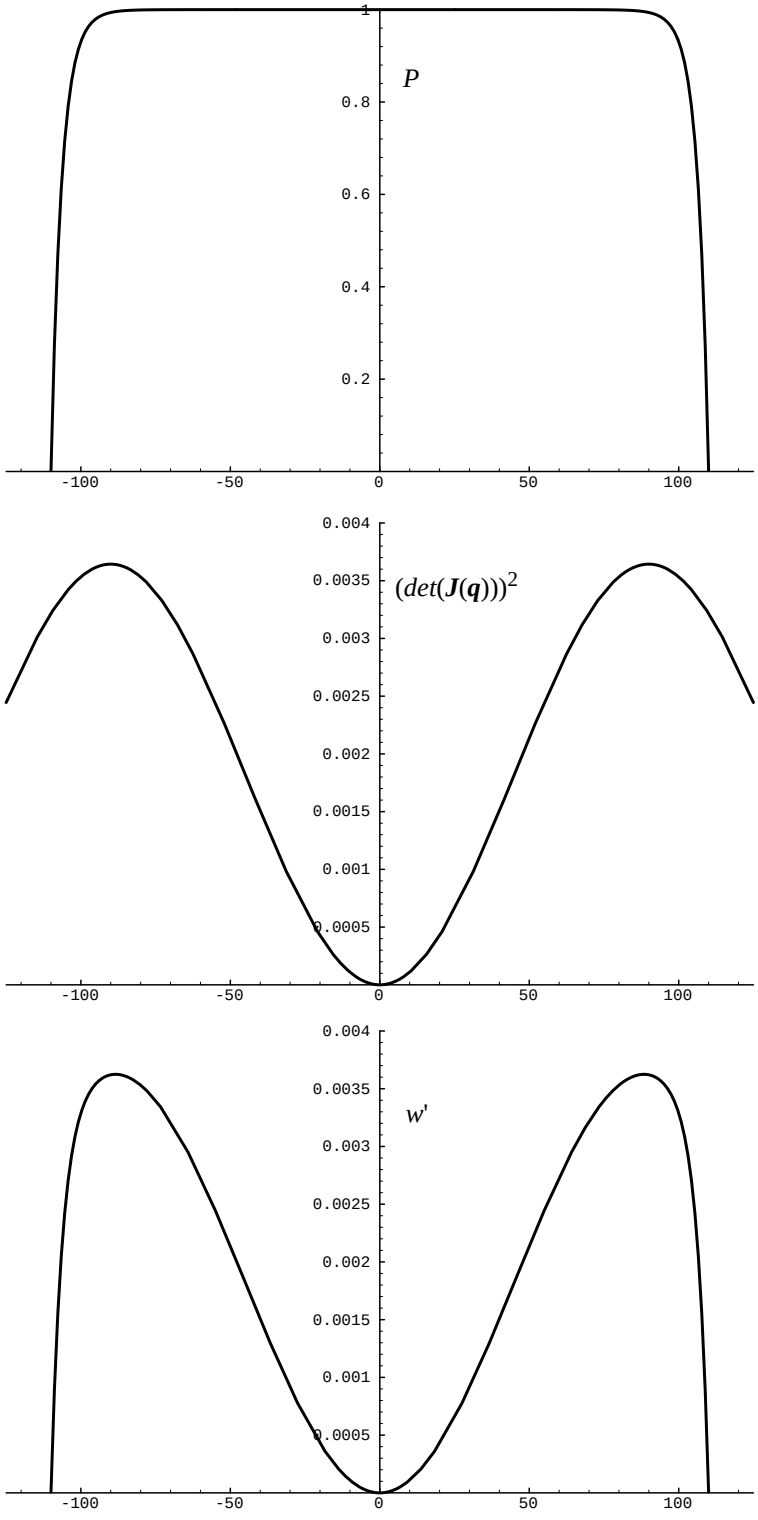

Figure 4. The modified manipulability measure employs a penalty function to indicate nearness to joint limits and a traditional manipulability measure to indicate nearness to singularities.

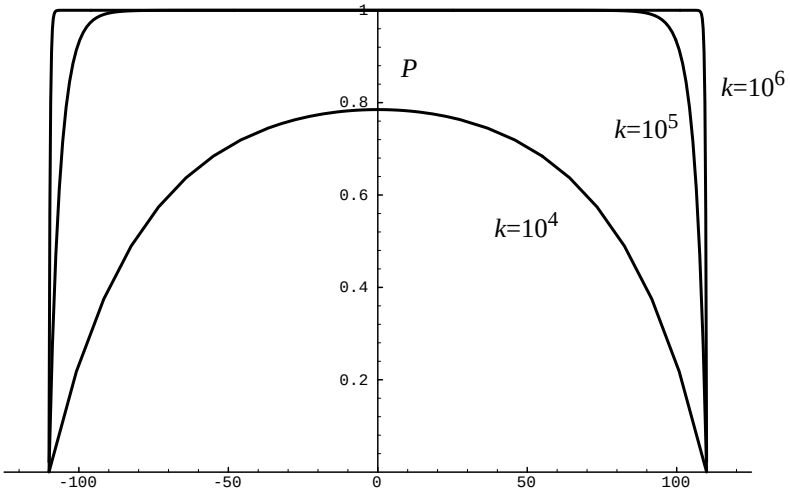

Figure 5. The joint limit penalty function for various values of k as the fifth joint of a Puma 560 travels between its joint limits.

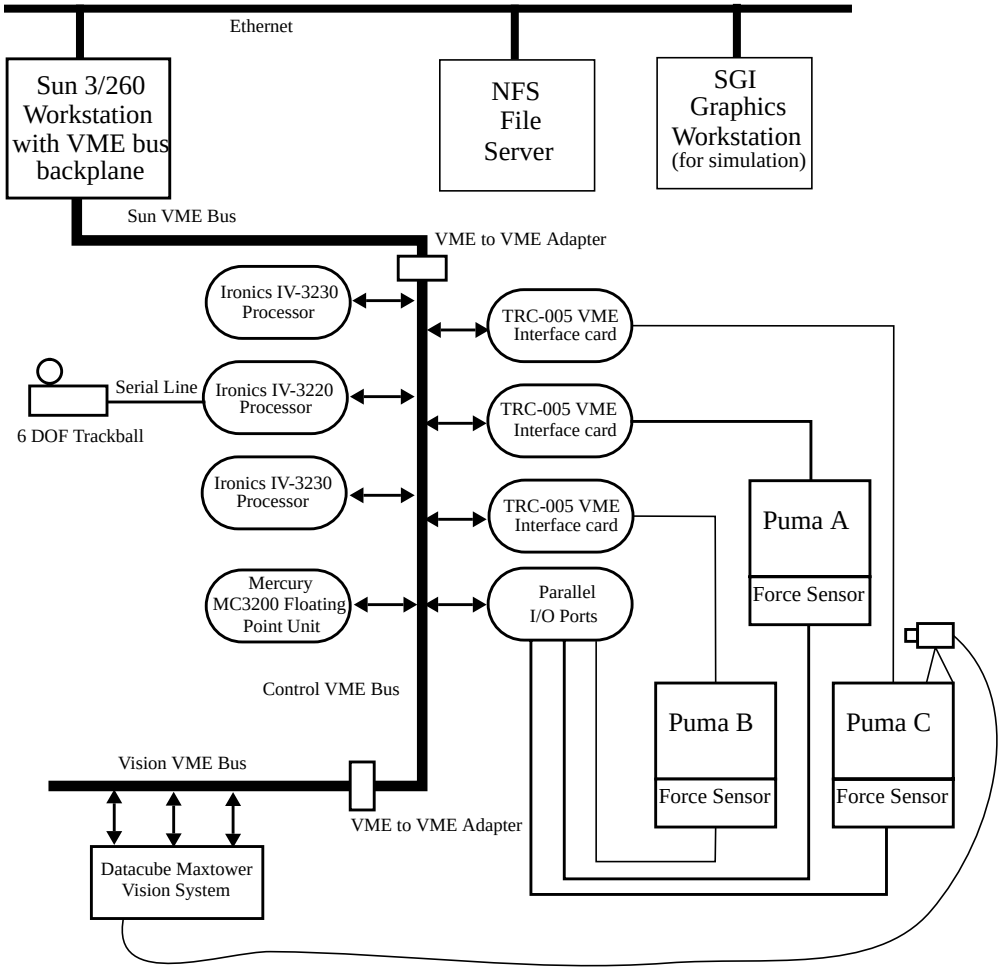

Figure 6. The Troikabot system architecture.

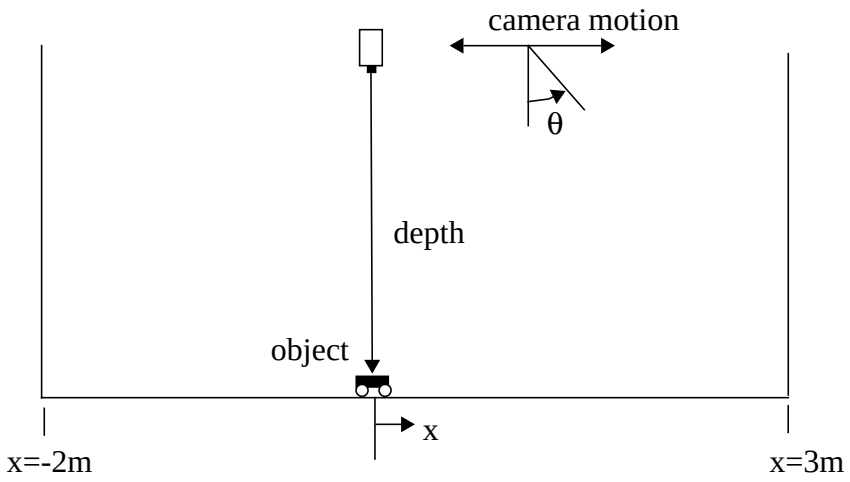

Figure 7. Experimental results were collected using the illustrated setup. The object was placed on a small mobile robot moving at a constant speed within a five meter wide workspace. The configuration shown represents the initial conditions of the trials from which the experimental results presented were obtained.

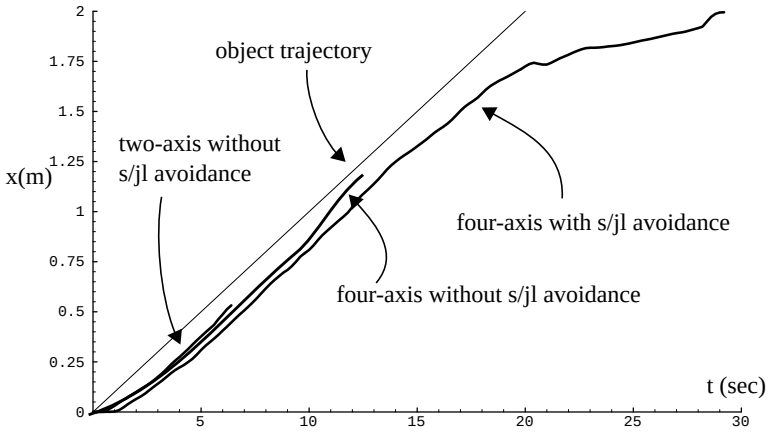

Figure 8. Camera and object translation along +x versus time for two-axis visual tracking without singularity/joint limit avoidance and for four-axis visual tracking with and without singularity/joint limit avoidance.

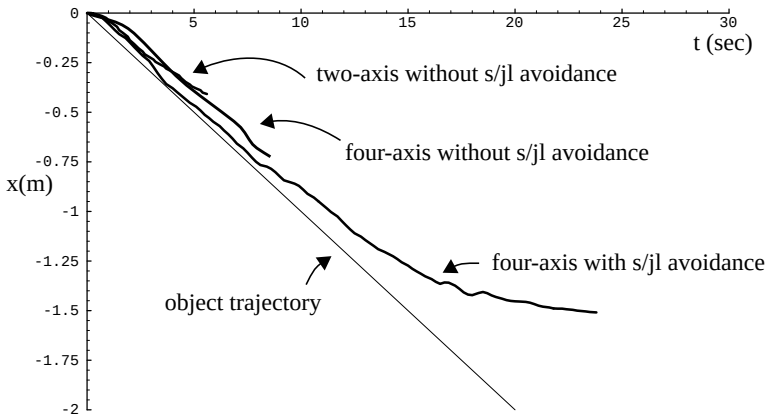

Figure 9. Camera and object translation along -x versus time for two-axis visual tracking without singularity or joint limit avoidance and for four-axis visual tracking with and without singularity/joint limit avoidance.

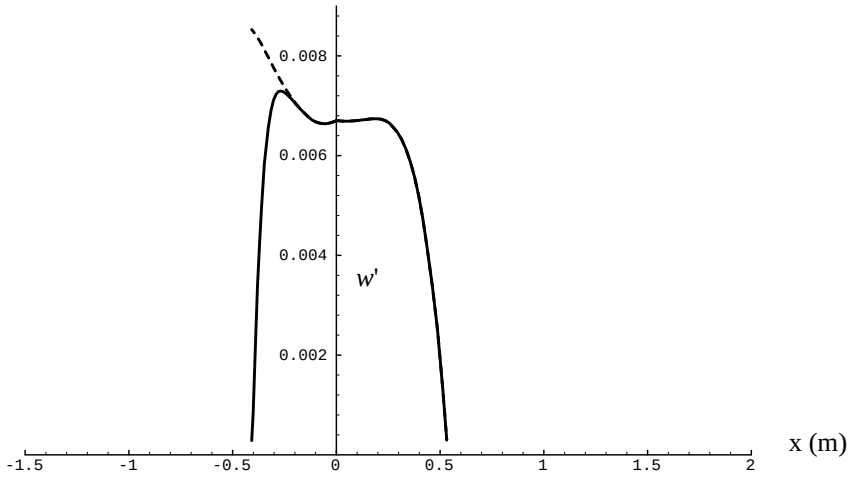

Figure 10. Manipulability versus x for two-axis visual tracking without singularity or joint limit avoidance. The dashed line indicates manipulability without the joint limit penalty function.

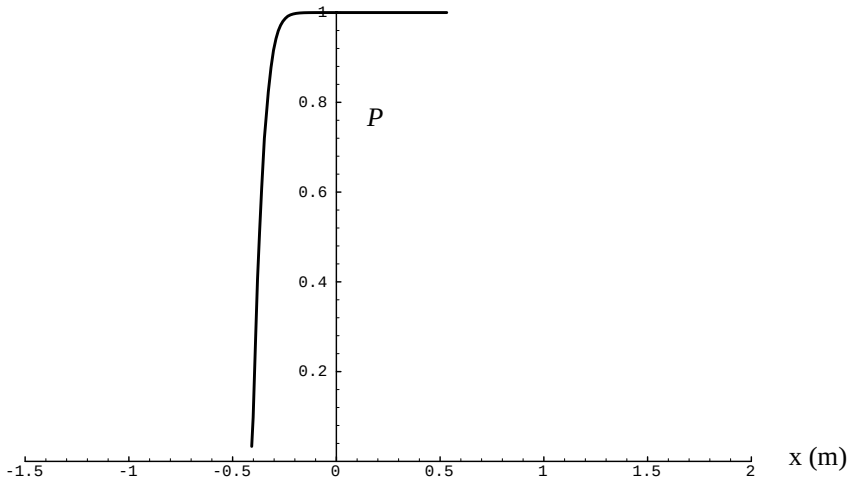

Figure 11. Joint limit penalty function *P* versus x for two-axis visual tracking without singularity or joint limit avoidance.

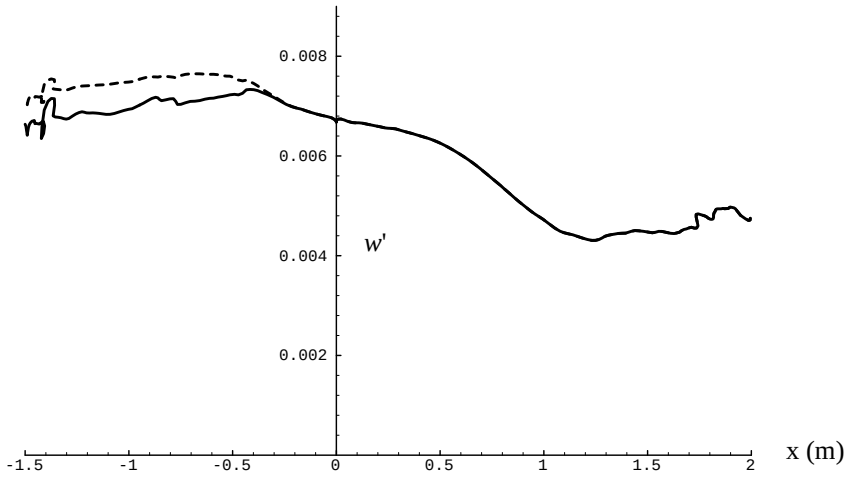

Figure 12. Manipulability versus estimated x for four-axis visual tracking with singularity and joint limit avoidance. The dashed line indicates manipulability without the joint limit penalty function.

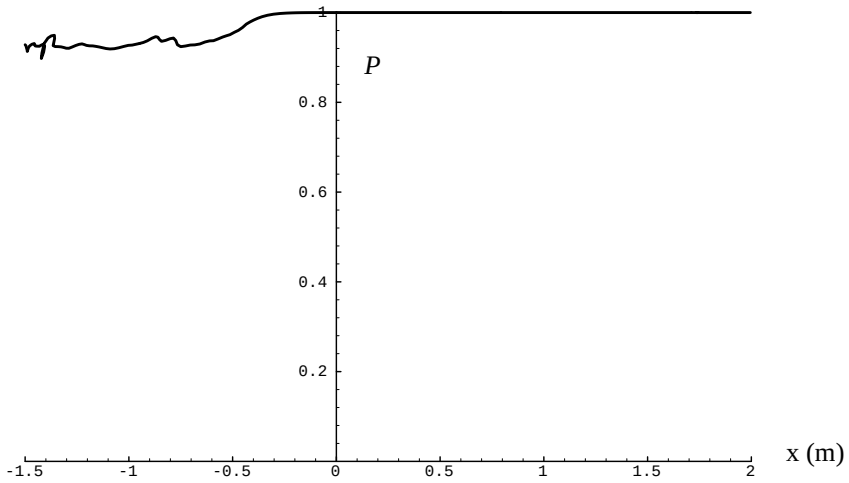

Figure 13. The joint limit penalty function versus estimated x for four-axis visual tracking with singularity and joint limit avoidance.

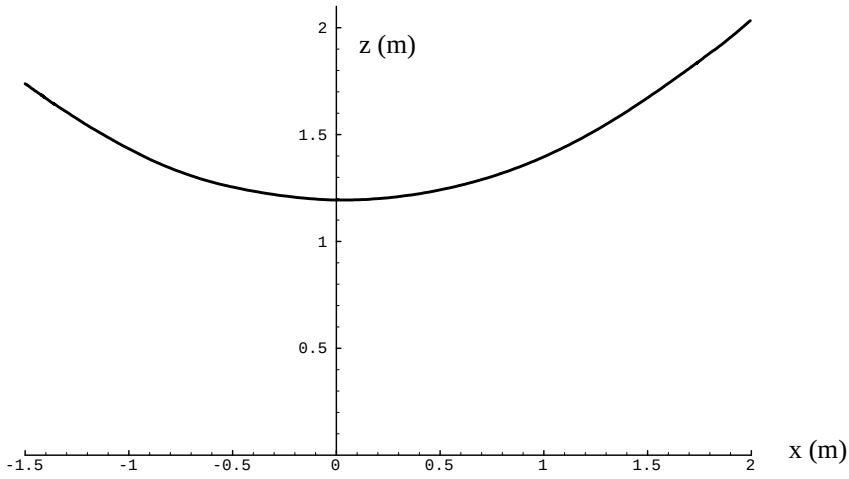

Figure 14. Depth of the object versus estimated x for four-axis visual tracking with singularity and joint limit avoidance.

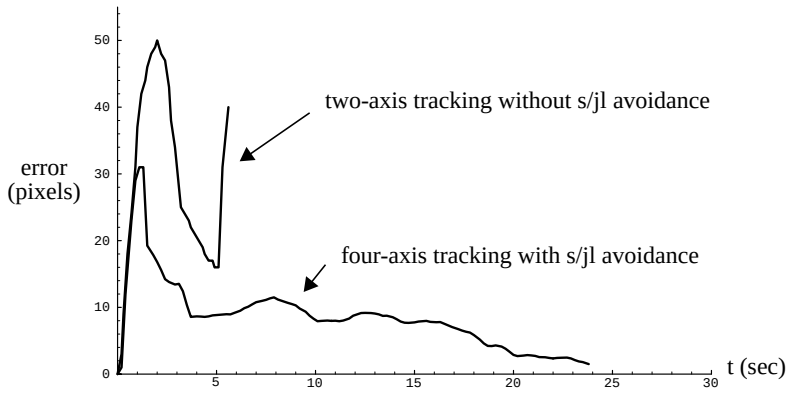

Figure 15. Feature error versus time for two and four axis tracking for the trial shown in Figure 9.