Robotic Manipulation Using High Bandwidth Force and Vision Feedback

Bradley J. Nelson*
J. Daniel Morrow*
Pradeep K. Khosla**

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

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

Abstract

High bandwidth sensor feedback is necessary for performing precise manipulation tasks within imprecisely calibrated and dynamically varying environments. Force controlled manipulation is a common technique for compliantly contacting and manipulating uncertain environments. Visual servoing is an effective technique for guiding imprecisely calibrated manipulators with imprecisely calibrated camera-lens systems. These two types of manipulator feedback, force and vision, represent complementary sensing modalities: visual feedback provides information over a relatively large area of the workspace without requiring contact with the environment, and force feedback provides highly localized and precise information upon contact. This paper presents three different strategies which combine force and vision within the feedback loop of a manipulator; traded control; hybrid control; and shared control. These strategies can be used alone, or in concert with each other, depending on the manipulation task. A discussion of the types of tasks that benefit from the strategies is included, as well as experimental results which show that the use of visual servoing to stably guide a manipulator simplifies the force control problem by allowing low gain force control with relatively large stability margins.

Submitted to *Mathematical and Computer Modelling - An International Journal*, 1995

1. Introduction

Force and vision feedback complement one another. Vision allows accurate part alignment within imprecisely calibrated and dynamically varying environments without requiring that objects contact one another. Force provides localized but accurate contact information, thus allowing the task to be controlled with a precision beyond the capability of vision alone.

Combining force and vision feedback within a feedback loop presents many challenges which are addressed in this paper. The camera-lens system, the manipulator, and the force sensor must all be properly modeled. Because of the different nature of force and vision feedback, traditional sensor integration techniques are not applicable, and separate control strategies specific to particular classes of tasks must be derived. The most common method of implementing force control is through a wrist force sensor and a stiff manipulator position control loop. Because the force sensor measures all forces (inertial, gravitational, and tactile), the inertial coupling of the end-effector mass beyond the sensor introduces a fundamental problem when sharing control between force and vision sensors. When the vision system commands motions, the resulting accelerations can result in inertial forces which cause unstable excitations of the force control system. In order to compensate for unstable excitations, it is necessary to develop robust strategies for avoiding the excitations.

This paper presents three different strategies with which force and vision can be combined within a manipulator feedback loop. These three strategies are traded, hybrid, and shared control. Traded and hybrid control avoid force instabilities due to inertial coupling by specifying that force and visual servoing do not occur simultaneously along the same direction. **A** novel shared control strategy was developed which does not require this assumption. Our proposed controller effectively compensates for inertial coupling while responding to both vision and force feedback along the same direction. Depending on the **type** of manipulation task and the accuracy of the visual feedback, these three strategies may all be used during the task, or a single strategy may be capable of performing an entire task. Experimental results from all three control strategies are presented in the paper, along with a discussion of the applicability **of** each of these strategies to different types of tasks. Experimental results **show** that the use of visual servoing to stably guide a manipulator simplifies the force control problem by allowing low gain force control with relatively large stability margins.

2. Previous Work

2.1. Force Servoing

Robotic force control has been studied since the 1950's. A survey on the history of force control can be found in (Whitney 1985). Active impedance control has been suggested **as** the most general form of force control (Hogan 1985), however, difficulties in programming impedance controlled manipulators have resulted in very limited use of **this** strategy. Hybrid control (Raibert and Craig 1981) separates position control and force control into two separate control loops that operate in orthogonal directions; **as** shown in Figure 1, and was extended by (Yoshikawa 1987) to include manipulator dynamics.

Qne **of** the most important issues in force control is maintaining manipulator stability during force control (Whitney 1985). Force controllers must be properly formulated and tuned in order to maintain stability, and this can be difficult, particularly upon initial contact with a stiff surface (Volpe and Khosla 1991). ***An*** important contribution of the work to be presented in this paper is that we show how vision can be used to greatly simplify the stability problem. Since stability becomes much less of an issue, the use of force feedback during manipulator fine motion is more easily realized.

2.2. Visual Servoing

Visual servoing has a less extensive history than that **of** force control, mainly due to the lack of computational resources available for processing the large amounts of data contained in an image. Although previous researchers had considered fast visual feedback for guiding manipulator motion, for example (Shirai and Inoue 1973), the visual servoing field was first well defined by (Weiss 1984). Since the work by Weiss, two types of visual servoing configurations have emerged, eye-in-hand configurations and static camera configurations. Eye-in-hand visual servoing tracks objects **of** interest with a camera mounted on a manipulator's end-effector (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). Static camera visual servoing guides manipulator motion based on feedback from a camera observing the end-effector (Koivo and Houshangi 1991; Nelson, Papanikolopoulos and Khosla 1993; Castaño and Hutchinson 1994; Hager 1994). Most of this past work has been with monocular systems, though recently stereo systems have been used for visual servoing (Maru et al. 1993; Hager 1994; Hosoda and Asada 1994). It is important to note that almost all proposed control schemes use locally linearized approaches, although recent work in applying perspective systems theory to the visual servoing problem has shown promise (Ghosh 1992), and the use of Extended Kalman Filtering (Wilson 1993) has demonstrated impressive experimental results.

A typical visual servoing feedback loop is shown in Figure 2. Differences between the various approaches to visual servoing include the space in which reference inputs are provided, the dimensionality of the control space, the structure of the controller, the physical configuration of the system, the derivation of the control law, and the feature tracking algorithms used. An excellent survey of recent work in visual servoing can be found in (Corke 1993).

2.3. Force and Visual Servoing

One of the first papers to realize the benefits of combining high bandwidth visual and force feedback is by (Shirai and Inoue 1973). They implemented a 0.1Hz visual servoing scheme and mentioned the benefits of force servoing, but a lack of computational resources hampered their effort, and many of the issues of combining the two sensing modalities went unnoticed. In (Ishikawa, Kosuge and Furuta 1991), visual servoing of 2 Hz was used to align a wrench with a bolt before a compliant wrenching operation is performed. Again, vision and force were not explicitly combined, and the issues concerning their integration remained unaddressed.

We consider here for the first time the integration of high bandwidth visual feedback with high bandwidth force feedback within the same manipulator feedback loop. First, we discuss the visual servoing problem and derive a visual servoing control law. We then present a force control scheme upon which our visual/force servoing strategies are based. Three different strategies are discussed, and experimental results presented.

3. Visual Servoing

For visually controlling an object held by a manipulator end-effector, static camera visual servoing is considered. The visual servoing component of our system is an extension of the controlled active vision paradigm (Papanikolopoulos, Khosla and Kanade 1991), originally created for monocular eye-in-hand visual servoing. In this section we extend controlled active vision to static camera visual servoing. A model of the camera-lens-manipulator system is presented, followed by a derivation of the visual servoing control law.

3.1. System Model

In formulating the visual servoing component of our system, the Jacobian mapping from task space to sensor space is first derived. We desire a Jacobian for the camera-lens system of the form

$$\dot{\boldsymbol{x}}_S = \boldsymbol{J}_c(\phi)\dot{\boldsymbol{x}}_T \qquad (1)$$

where $\dot{x}_S$ is a velocity vector in sensor space, $\boldsymbol{J}_c(\phi)$ is the Jacobian matrix and is a function of the extrinsic and intrinsic parameters of the visual sensor as well as the number of features tracked and their locations on the image plane, and $\dot{x}_T$ is **a** velocity vector in task space.

3.1.1. Camera-Lens Model

A simple pin hole camera model (Gremban, Thorpe and Kanade 1988) has proven adequate for visual tracking using our experimental setup. If we place the camera coordinate frame {C} at the focal point of the lens **as** shown in Figure 3, a feature on **an** object at ${}^C\boldsymbol{P}$ with coordinates (X , Y_C, Z_C) in the camera frame projects onto the camera's image plane at

$$x_i = \frac{fX_C}{s_x Z_C} + x_p \qquad (2)$$

$$y_i = \frac{fY_C}{s_y Z_C} + y_p \qquad (3)$$

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. This model assumes that $Z_C >> f$.

The mapping from camera frame feature velocity to image plane optical flow, or sensor space velocity, can be obtained simply by differentiating (2) and (3). This yields the following equations

$$\dot{x}_S = \frac{f\dot{X}_C}{s_x Z_C} - \frac{fX_C\dot{Z}_C}{s_x Z_C^2} = \frac{f\dot{X}_C}{s_x Z_C} - x_S\frac{\dot{Z}_C}{Z_C} \qquad (4)$$

$$\dot{y}_S = \frac{f\dot{Y}_C}{s_y Z_C} - \frac{fY_C\dot{Z}_C}{s_y Z_C^2} = \frac{f\dot{Y}_C}{s_y Z_C} - y_S\frac{\dot{Z}_C}{Z_C} \qquad (5)$$

where $x_s = x_i - x_p$ and $y_s = y_i - y_p$. This defines the mapping of object velocity with respect to the camera frame onto the image plane. The next step is to transform task space velocities into the camera frame, and then project these camera frame velocities onto the sensor space to obtain the mapping from task space velocity to sensor space velocity.

3.1.2. Objects Defined in a Task Frame

For visually servoing a manipulator holding an object, the objective is to move the image coordinates of ${}^C P$ to some location on the image plane by controlling the motion of ${}^C P$. Typically, ${}^C P$ is some feature on an object being held by a manipulator. Thus, the motion of ${}^C P$ is induced relative to the tool frame of the manipulator being observed. Figure 3 shows the coordinate systems used to define the mapping from task space to sensor space, where ${}^C P$ is defined with respect to the task frame as ${}^T P$ with coordinates of (X_T, Y_T, Z_T). For now, we assume that the rotation of the task frame {T} with respect to {C) ${}^C_T R$ is known. The velocity of ${}^C P$ can be written as

$$cP = {}^C_T R\left({}^T V + {}^T\dot{P} + {}^T\Omega \times {}^T P\right) \tag{6}$$

where ${}^T V = \left[\dot{x}_T\ \dot{y}_T\ \dot{z}_T\right]^T$ and ${}^T\Omega = \left[\omega_{x_T}\ \omega_{y_T}\ \omega_{z_T}\right]^T$ are the translational and rotational velocities of the task frame with respect to itself. These are manipulator end-effector velocities that can be commanded. Since we assume that the object being servoed or observed is rigidly attached to the task frame, ${}^T\dot{P} = 0$, and (6) becomes

$$cP = {}^C_T R\left({}^T V + {}^T\Omega \times {}^T P\right) \tag{7}$$

Furthermore, if we assume that {C} and {T}are aligned, as shown in Figure 3, then ${}^C_T R = Z$ and the elements of ${}^C\dot{P}$ can be written as

$$\begin{aligned} \frac{dX_C}{dt} &= \dot{x}_T + Z_T\omega_{y_T} - Y_T\omega_{z_T} \\ \frac{dY_C}{dt} &= \dot{y}_T - Z_T\omega_{x_T} + X_T\omega_{z_T} \\ \frac{dZ_C}{dt} &= \dot{z}_T + Y_T\omega_{x_T} - X_T\omega_{y_T} \end{aligned} \tag{8}$$

The assumption that {C} and {T) are aligned is only used in formulating the Jacobian from task space to sensor space. If the transformation from task space to sensor space is initially known, and the commanded task frame velocity is known, then the coordinates (X_T, Y_T, Z_T) can be appropriately updated while visual servoing. It will also be necessary to account for task frame rotations when determining the velocity to command the task frame based on ${}^T V = \left[\dot{x}_T\ \dot{y}_T\ \dot{z}_T\right]^T$ and ${}^T\Omega = \left[\omega_{x_T}\ \omega_{y_T}\ \omega_{z_T}\right]^T$. It would have been possible to include the terms of ${}^C_T R$ in (8), however, the assumption made simplifies the derivation and does not affect the end result.

By combining (8) with (4) and (5), the entire Jacobian transformation for a single feature from task space to sensor space can now be written in the form

$$\begin{bmatrix} \dot{x}_S \\ \dot{y}_S \end{bmatrix} = \begin{bmatrix} \frac{f}{s_x Z_C} & 0 & -\frac{x_S}{Z_C} & -\frac{Y_T x_S}{Z_C} & \left[\frac{fZ_T}{s_x Z_C} + \frac{X_T x_S}{Z_C}\right] & -\frac{fY_T}{s_x Z_C} \\ 0 & \frac{f}{s_y Z_C} & -\frac{y_S}{Z_C} & -\left[\frac{fZ_T}{s_y Z_C} + \frac{Y_T y_S}{Z_C}\right] & \frac{X_T y_S}{Z_C} & \frac{fX_T}{s_y Z_C} \end{bmatrix} \begin{bmatrix} \dot{x}_T \\ \dot{y}_T \\ \dot{z}_T \\ \omega_{x_T} \\ \omega_{y_T} \\ \omega_{z_T} \end{bmatrix} \tag{9}$$

For the above form of the Jacobian, the parameters of the Jacobian are given by $\phi = (f, s_x, s_y, x_S, y_S, Z_C, X_T, Y_T, Z_T)$.

Generally, at least three features on an object are tracked, in order to control both the position and orientation of the object in three dimensions. For n feature points, the Jacobian is of the form

$$J_c = \begin{bmatrix} J_1 \\ \cdots \\ J_n \end{bmatrix} \tag{10}$$

where J_i is the Jacobian matrix for each feature given by the 2x6 matrix in (9) or (10).

3.2. Visual Servoing Controller

The state equation for the visual servoing system is created by discretizing (9) and rewriting the discretized equation as

$$x(k+1) = Ax(k) + TJ_c(k)u(k) \tag{11}$$

where M is the number of features being tracked, A=I_{2M}, $x(k) \in R^{2M}$, Tis the sampling period of the vision system, and $u(k) = \begin{bmatrix} \dot{x}_T & \dot{y}_T & \dot{z}_T & \omega_{x_T} & \omega_{y_T} & \omega_{z_T} \end{bmatrix}^T$, the manipulator end-effector velocity. The camera Jacobian $J_c(k)$ for the experimental system varies with time due to the changing feature coordinates on the image plane. The intrinsic parameters of the experimental camera-lens system are constant.

A control strategy can be derived using the controlled active vision paradigm (Papanikolopoulos, Khosla and Kanade 1991). The control objective of the visual tracking system is to control end-effector motion in order to place the image plane coordinates of features on the target at some desired position. The desired image plane coordinates could be constant or changing with time. 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

$$F(k+1) = [\boldsymbol{x}(k+1) - \boldsymbol{x}_D(k+1)]^T \boldsymbol{Q}\, [\boldsymbol{x}(k+1) - \boldsymbol{x}_D(k+1)] + \boldsymbol{u}^T(k) \boldsymbol{R} \boldsymbol{u}(k) \qquad (12)$$

This expression is minimized with respect to the current control input $\boldsymbol{u}(k)$. The end result yields the following expression for the control input

$$\boldsymbol{u}(k) = -\left(\boldsymbol{J}_c^{\,T}(k) \boldsymbol{Q} \boldsymbol{J}_c(k) + \boldsymbol{R}\right)^{-1} \boldsymbol{J}_c^{\,T}(k) \boldsymbol{Q}\, [\boldsymbol{x}(k) - \boldsymbol{x}_D(k+1)] \qquad (13)$$

The weighting matrices Q and 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 R must be positive definite for a bounded response. Although no standard procedure exists for choosing the elements of Q and R, general guidelines can be found in (Papanikolopoulos, Nelson and Khosla 1992).

The system model and control derivation 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.

3.3. 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 $\boldsymbol{\Delta x}=(\Delta x,\Delta y)$ which minimizes the SSD measure

$$\mathrm{e}\,(\boldsymbol{p}_a, \boldsymbol{\Delta x}) = \sum_{W} [I_a(x+i, y+j) - I_{a'}(x+i+\Delta x, y+j+\mathrm{Ay})]^2 \qquad (14)$$

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 comers, are typically the best features to select.

In order to decrease the search space, a pyramidal search scheme shown in Figure 4 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.

4. Visual and Force Servoing

The force control portion of our proposed visual/force servoing strategies is based on past work in hybrid force control. The implemented force control scheme is a combination of hybrid force/position control (Raibert and Craig 1981) and damping force control (Whitney 1977), resulting in a hybrid force/velocity control scheme. Because the dynamics, particularly friction, of the laboratory robot (a Puma 560) are difficult to accurately model, a simple Cartesian control scheme is used in which a manipulator Jacobian inverse converts Cartesian velocities to joint velocities, which are then integrated to joint reference positions. High servo rate (500Hz) PID controllers are implemented for each joint in order to follow joint trajectories which achieve the desired Cartesian motion. A diagram of the control system is shown in Figure 5.

For manipulation, visual servoing brings objects near one another and into proper alignment, and force servoing ensures compliant contact and accomplishes the final parts mating. For teleoperation, a mouse cursor can be used to guide the manipulator by a teleoperator observing live video images of the robot. Visual servoing is used to control the manipulator's end-effector position based on reference inputs from the cursor position, while force sensing is used to contact the environment compliantly. Feedback from other devices, for example a trackball, may also be used to guide the manipulator. Three different types of controllers C(t) have been implemented and tested—a shared, a traded, and a hybrid controller. A generic control framework for visual/force servoing is shown in Figure. 6.

4.1. Traded Control

There are two main benefits to using visual servoing and force servoing together. First, visual servoing is effective in significantly reducing manipulator instability upon impact. Second, visual servoing can be used to perform precision manipulation tasks in imprecisely calibrated environments by accurately aligning objects for manipulation despite poorly calibrated camera-lens-manipulator systems. A traded control strategy realizes both of these benefits. For traded control, the control law given by C(t) in Figure 6 is written as

$$\boldsymbol{u}(k) = \begin{cases} \left(\boldsymbol{J}_c^T \boldsymbol{Q} \boldsymbol{J}_c + \boldsymbol{R}\right)^{-1} \boldsymbol{J}_c^T \boldsymbol{Q}\,[\boldsymbol{x}_D - \boldsymbol{x}(k)], & \|\boldsymbol{x}_D - \boldsymbol{x}(k)\| > \varepsilon \qquad (15)\,\mathrm{a} \\ \boldsymbol{S}_F \boldsymbol{G}_F (\boldsymbol{F}_r - \boldsymbol{F}_m(k)), & \|\boldsymbol{x}_D - \boldsymbol{x}(k)\| \le \varepsilon \qquad (15)\,\mathrm{b} \end{cases}$$

During traded control, manipulator motion is first controlled by visual feedback (15)a. The controller then switches to force control after visual servoing approaches sufficiently near a surface (15)b. The desired state of the manipulator on the image plane $\boldsymbol{x}_D$ represents a state near the surface to be contacted. $\boldsymbol{S}_F$ is a matrix that selects axes along which force control will be applied, $\boldsymbol{G}_F$ is a matrix of force control gains, and $\boldsymbol{F}_r$ and $\boldsymbol{F}_m$ represent reference and measured forces. The main benefit of this strategy is that stable impact with a surface can be achieved under force control without inducing large contact forces. With simple force damping control alone, a manipulator can easily become unstable during impact, unless force gains are tuned to extremely low values, resulting in unacceptably slow motion. Because of this, most manipulation strategies use a guarded move to initiate contact with a surface. During a guarded move, surfaces are approached under position control while the force sensor is monitored. If the sensed force exceeds a threshold, motion is immediately stopped and a force control strategy can then be invoked. The main limitation of this strategy is that high contact forces can result unless the effective mass of the manipulator is low so that the end-effector can be quickly stopped before contact forces increase significantly. In Section 7.1, experimental results are presented which demonstrate these three impact strategies, their stability, and the impact forces created by each strategy.

4.2. Hybrid Control

An obvious technique for combining visual servoing with other types of feedback, for example position feedback (Castaño and Hutchinson 1994), is in a hybrid control scheme. This type of scheme, however, is limited because it ignores much of the information provided by visual feedback. The proper use of vision in the feedback loop provides information relevant to motion along all six degrees of freedom with which an object can move. Hybrid control allows visual servoing only in those directions that are orthogonal to directions along which other types of feedback is used. Despite this drawback there are specific classes of task which fall naturally into a hybrid control scheme. One useful application is the grasping of electrical lines through teleoperation (Nio and Maruyama 1993). For this task, hybrid control could maintain the alignment of the gripper with the wire to be grasped, and a teleoperator or other sensor, such as a force sensor or an optical switch, could signal when the line had been reached, much as a guarded move is used to indicate when contact is made under position control.

We have used this scheme to demonstrate remote teleoperated grasping of objects in which a supervisor is presented with live video input from a single camera mounted on the manipulator's gripper (Figure 7). Visual servoing maintains gripper alignment, while the supervisor controls the velocity with which the object is approached, thus a hybrid control scheme is used. Initial experimental results demonstrate that the task can be completed much more quickly and with a higher probability of success when using hybrid visual servoing than when the teleoperator must control all degrees of manipulator motion. An important advantage of this strategy is that the camera-lens-manipulator system need not be precisely calibrated.

The hybrid control law we use is of the form

$$\begin{aligned} \boldsymbol{u}(k) &= \boldsymbol{S}_v\left(\boldsymbol{J}_c{}^T\boldsymbol{Q}\boldsymbol{J}_c+\boldsymbol{R}\right)^{-1}\boldsymbol{J}_c{}^T\boldsymbol{Q}\,[\boldsymbol{x}_D(k)-\boldsymbol{x}\,(k)\,]+\boldsymbol{S}_r\dot{\boldsymbol{x}}_r+\boldsymbol{S}_F\boldsymbol{G}_F(\boldsymbol{F}_r-\boldsymbol{F}_m) \\ & \boldsymbol{S}_v+\boldsymbol{S}_r+\boldsymbol{S}_F = \boldsymbol{I} \end{aligned} \tag{16}$$

S_v, S_r, and S_F are diagonal selection matrices, the elements of which are 0 or 1, and which must sum to the identity matrix.

4.3. Shared Control

Shared control allows the use of visual and force control along the same directions simultaneously. This strategy is useful if surfaces can not be reliably detected with vision and it is necessary that the manipulator reacts compliantly to stiff environments along directions of visual servoing. One application of visual servoing for which shared control is particularly useful is when visual feedback is used in teleoperation. Experiments have been performed in which a user guides a manipulator under visual control with a mouse and cursor on a live video image. When guiding the manipulator to grasp **an** object, the user may sometimes command the manipulator to contact a surface. This can damage the object and often causes the manipulator to become unstable. With shared control, this is easily avoided because force **as** well **as** visual feedback is used throughout the task.

A fundamental problem when sharing control between force and vision sensors occurs due to end-effector inertial effects. Because force sensors measure all forces (inertial, gravitational, and tactile), the inertial coupling of the end-effector mass beyond the sensor introduces inertial forces into force sensor readings. When the vision system commands motions, the resulting accelerations cause unstable excitations of the force control system. In order to compensate for the unstable excitations, it is necessary to develop robust strategies for avoiding the excitations. Thresholding of force readings is not feasible, since inertial effects can often be as large as desired contact forces. Figure **8** shows the magnitude of experimentally determined inertial forces and the associated measured Cartesian accelerations which cause these forces.

We have developed a robust shared strategy based on the realization that large accelerations induce inertial forces. If visual servoing results in measurable end-effector accelerations of sufficient magnitude, then force readings in directions opposite to these accelerations are being induced. Because measured Cartesian accelerations are derived from joint encoder readings, thus requiring two differentiations of measured joint values and a transformation from joint space to Cartesian space, measured Cartesian accelerations can be noisy. Therefore, we also consider the measured direction of end-effector motion. If measured Cartesian accelerations have been induced by visual servoing, and if a measurable Cartesian velocity exists, then sensed forces must be due to inertial coupling, and force control commands should be ignored. This strategy can be written as

$$\dot{x}_{refv} = \left(J_c{}^T Q J_c + R\right)^{-1} J_c{}^T Q \left[x_D(k) - x(k)\right]$$

$$\dot{x}_{reff} = G_F (F_r - F_m)$$

```
for each axis, i {
    if $( ( ( |\ddot{x}_{mi}| > \varepsilon_a) \wedge (\dot{x}_{mi} sign(F_{mi}) < \varepsilon_v) ) \vee (\dot{x}_{refvi} F_{mi} > 0.0) \vee (|F_{mi}| < F_T) )$
        $S_v[i, i] = 1.0 \quad S_F[i, i] = 0.0$
    else
        $S_v[i, i] = 0.0 \quad S_F[i, i] = 1.0$
}
```

$$u(k) = S_v \dot{x}_{refv} + S_r \dot{x}_r + S_F \dot{x}_{reff} \tag{17}$$

where E„ E„ and F_T threshold sensor noise.

5. 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. 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). An Adept robot is also used for providing accurate target motion. The entire system is shown in Figure 9.

A diagram of the hardware setup is shown in Figure 10. 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 3.3. A special high performance floating point processor on the Datacube is used to calculate the optical flow of features, and a 68030 board, also on the vision system, computes the control input. An image can be grabbed and displacements for up to five 16x16 features in the scene can be determined at 30Hz. The force sensor provides force and torque values for each Cartesian axis at 100Hz.

6. Experimental Results

6.1. Traded Control

Throughout this section, experimental results given will be referenced to the coordinate frames shown in Figure 11. Results from a traded control strategy are shown in Figure **12** in which visual servoing is first used to servo the end-effector to a surface 5.9cm from the initial end-effector position. The controller then switches to force control and achieves stable contact with the surface, maintaining a force of **-2N** between the end-effector and the surface. This strategy achieves contact after 1.43s, and achieves a stable **2N** contact force after approximately 4.5s. With simple damping force control alone, the manipulator travels 5.9cm in **3.1s** before reaching the surface. As soon as the surface is contacted, the manipulator becomes unstable, as Figure 13 shows. The only way to achieve stable contact using damping control alone given the force control implementation used, is to reduce the force gains to extremely low values, resulting in unacceptably slow motion. Because of this, most manipulation strategies use a guarded move to initiate contact with a surface. During a guarded move surfaces are approached under position control while the force sensor is monitored. If the sensed force exceeds a threshold, motion is immediately stopped and a force control strategy can then be invoked. Figure **14** shows a force plot of a guarded move in which the force sensor **is** monitored at **100Hz**. High contact forces are created because of the finite amount of time required to stop the end-effector after contact is sensed illustrating the main limitation of a guarded move strategy.

Figure **15** shows a comparison of the motion and force time histories for the three impact strategies. Traded visual/force servoing uses visual servoing to quickly approach the surface before switching to a stable low gain (0.001 (m/s)/N) damping force controller. For force control alone, higher force gains (0.005 (m/s)/N) had to be chosen in order to induce end-effector motion of a reasonable speed in free space, but this gain, while resulting in less than half the speed of visual servoing, still proved to be highly unstable. The guarded move strategy also allowed only moderate speeds (0.02m/s), and still resulted in high impact forces. At higher speeds, extremely high impact forces would result which could have easily damaged the manipulator. Using visual servoing to bring the manipulator near the surface provides a simple technique for slowing the end-effector before contact is imminent. These results clearly show that visual servoing greatly simplifies the impact problem by providing low-level feedback on the proximity of the surface to the end-effector.

6.2. Hybrid Control

To illustrate hybrid visual/force servoing, the manipulator is servoed to maintain a force of 2N normal to a moving plate while maintaining the point of contact in the center of the plate. The setup is shown in Figure **9.** The edges of the plate are visually tracked, and the manipulator is visually servoed along X to maintain the proper contact point. **As** the plate moves in X, force servoing along Y maintains the 2N contact force. Plots of the varying motion of the end-effector in X and the applied force in Yare shown in Figure 16. Motion along the visual servoing direction is precise, but the force servoing data shows the effects of force sensor noise introduced by the motion along X. Even so, contact is maintained and is repeatable. Figure 17 shows visual servoing performance on the image plane. The plate moves with a maximum speed of approximately 5cm/s and with extremely high accelerations when switching directions, but the manipulator very accurately tracks this motion despite the large accelerations required.

6.3. Shared Control

To demonstrate shared control, visual servoing is again used to bring the end-effector to the surface, as in the traded control experiments. However, for these experiments the desired goal position for the visual servoing strategy is purposely chosen to be beneath the surface. We are then able to judge the abilities of our shared control strategy (17) to operate under conditions in which force and vision information disagree. Figure 18 shows the motion of the end-effector on the image plane for two trials in which the desired image plane position of the end-effector y_D actually falls beneath the true surface. In one trial the error in surface position is **45** pixels, and in the other trial the error is 15 pixels. In both trials, the end-effector impacts the surface after approximately 0.3s, when motion of the end-effector on the image plane abruptly stops. The commanded end-effector velocity for these two trials is shown in Figure 19. The solid lines correspond to $u(k)$ in (17), the dashed lines to the visual servoing velocity $\dot{x}_{refv}$, and the dotted/dashed lines to the force servoing velocity $\dot{x}_{reff}$. Visual servoing brings the end-effector quickly towards the surface, and upon contact force servoing takes over. Figure **20** shows a plot of end-effector position and measured force for the same trials. **As** expected, when visual feedback incorrectly estimates the location of the surface, high contact forces occur. The poorer the estimate of the surface, the higher the contact force, because the higher the commanded visual servoing velocity. Because the feature tracker can only track objects with a limited optical flow, it is necessary to clip end-effector velocities at 0.10m/s, so the **trial** in which the surface location is in error of 45pixels represents the worst case impact force, because the manipulator is traveling at approximately 0.10m/s at the time of impact.

7. Conclusion

Combining force and visual feedback within a manipulator feedback loop allows a robotic system to take advantage of the complementary nature of the two sensing modalities. Several issues must be addressed when integrating force and vision, including the inertial effects visual feedback may induce in the force control loop. We have shown how these effects can be eliminated, and how stable and robust control using visual and force feedback simultaneously can be achieved. Although we presented three different visual/force control strategies, a manipulator could benefit from all three strategies while executing a single task. While traveling in free space, shared control should be used to ensure that poor estimates in surface locations do not result in unrecognized impacts. As a surface is approached, the shared strategy is capable of stably contacting the surface if the visual surface estimate is actually within the surface, otherwise, traded control could be used and pure low gain force control could initiate the final contact. Throughout the task, hybrid control could be used depending on the requirements of the task. Whatever the manipulation task, the use of visual servoing to quickly and stably guide a manipulator towards a surface greatly simplifies the force control problem by allowing the efficient use of low gain force controllers with relatively high stability margins.

8. Acknowledgments

Brad Nelson was supported in part by a National Defense Science and Engineering Graduate Fellowship through 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.

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.

Castaño, A. and Hutchinson, S. 1994. Visual compliance: task-directed visual servo control. ***IEEE Trans. on Robotics and Automation.*** 10(3):334-342.

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. **429-45**1.

Corke, P.I. Visual control of robot manipulators-a review. *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback,* ed. K. Hashimoto. London:World Scientific, pp. **1-31**.

Espiau, B., Chaumette, F. and Rives, P. **1992.** A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation.* **8(3):3 13-326.**

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.

Ghosh, B.K., Jankovic, M. and Wu, Y.T. **1992.** Some problems in perspective system theory and its application to machine vision. *Proc. 1992 IEEERSJ Int. Conf. on Intelligent Robots and Sys. (IROS92).* New York:IEEE, pp. **139-146.**

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

Hager, G.D. Robot feedback control based on stereo vision: towards calibration-free hand-eye coordination. *Proc. 1994 Int. Con. on Robotics and Automation.* New York:IEEE, pp. **2850-2856.**

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.**

Hogan, N. **1985.** Impedance control: an approach to manipulation, Parts I-III. *ASME J. Dyn. Sys., Measurements, and Control.* 107(1): **1-24.**

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

Hosoda, K. and Asada, M. **1994.** Versatile visual servoing without knowledge of true jacobian. *Proc. 1994 IEEE/RSJ Int. Conf. on Intelligent Robots and Sys. (IROS94).* New York:IEEE, to appear.

Ishikawa, J., Kosuge, K. and Furuta, K. **1990.** Intelligent control of assembling robot using vision sensor. *Proc. 1985 Int. Conf. on Robotics and Automation.* New York:IEEE, pp. **1904-1909.**

Koivo, A.J. and Houshangi, N. **1991.** Real-time vision feedback for servoing of a robotic manipulator with self-tuning controller. *IEEE Trans. on Sys, Man, and Cybernetics,* **21**(1): **134-142.**

Maru, N., Kase, H., Yamada, S., Nishikawa, A. and Miyazaki, F. **1993.** Manipulator control by visual servoing with the stereo vision. *Proc. 1993 IEEERSJ Int. Conf. on Intelligent Robots and System (IROS-93).* pp. **1866-1870.**

B. Nelson, N.P. Papanikolopoulos, and P.K. Khosla. 1993. Visual servoing for robotic assembly. ***Visual Servoing–Real-Time Control*** **d** ***Robot Manipulators Based on Visual Sensory Feedback.*** ed. K. Hashimoto. River Edge, NJ:World Scientific Publishing Co. Pte. Ltd. pp. 139-164.

Nio, S. and Maruyama Y. 1993. Remote-operated robotic system for live-line maintenance work.

Papanikolopoulos, N.P., Khosla, P.K. and Kanade, T. 1991. Adaptive robotic visual tracking. ***Proc.*** **d** ***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 ZEEE Znt. Symp. on Intelligent Control (ISIC-92).*** New York:IEEE, pp. 267-274.

Raibert, M.H. and Craig, J.J. 1981. Hybrid positiodforce control of manipulators. ***ASME*** **J. d** ***Dyn. Sys., Measurement, and Control.*** 103(2): 126-133.

Shirai, S. and Inoue H. 1973. Guiding a robot by visual feedback in assembling tasks. ***Pattern Recognition.*** 5:99-108.

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.

Volpe, R.A. and Khosla, P.K. 1991. Experimental verification of a strategy for impact control. ***Proc. 1991 ZEEE Int. Conf. on Robotics and Automation.*** New York:IEEE. pp. 1854-1860.

Weiss. L.E. 1984. Dynamic visual servo control of robots: an adaptive image-based approach. Ph.D Thesis CMU-RI-TR-84-16, Pittsburgh, PA:The Robotics Institute Carnegie Mellon University.

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

Whitney, D.E. 1977. Force feedback control of manipulator fine motions. ***ASME*** **J.** ***of Dyn. Sys. Measurement, and Control.*** June:9 1-97.

Whitney, D.E. 1985. Historical perspective and state of the *art* on robot force control. ***Proc. 1985 Int. Conf. on Robotics and Automation.*** New York:IEEE, pp. 262-268.

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. 1987. Dynamic hybrid position/force control of robot manipulators-description of hand constraints and calculation of joint driving force. ***IEEE*** **J.** ***of Robotics and Automation.***

3(5):386-392.

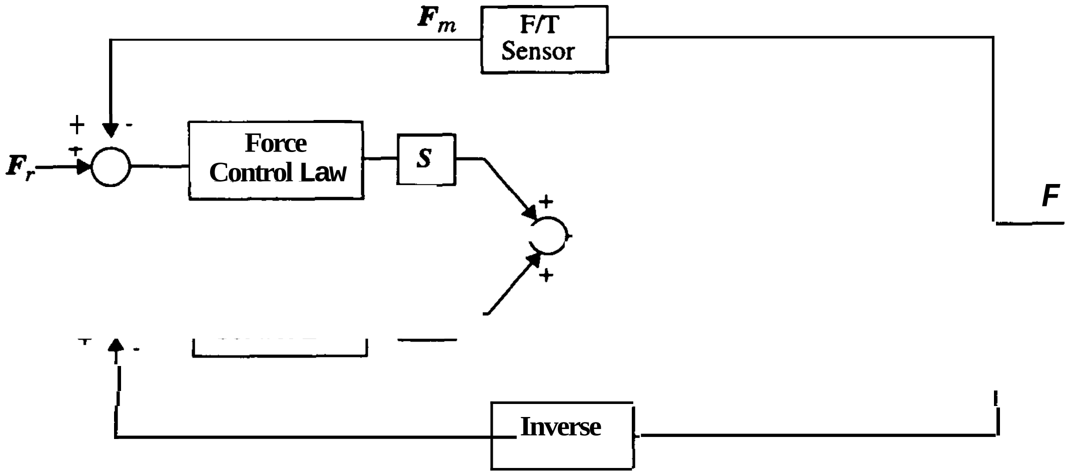

Figure 1. Hybrid force/position control loop, where F_r is the reference force vector, F_m is the measured force vector, X_r is the reference position, X_m is the measured position, *S* and **S'** are the orthogonal selection matrices, F is the applied force, and q_m is a vector of measured joint positions.

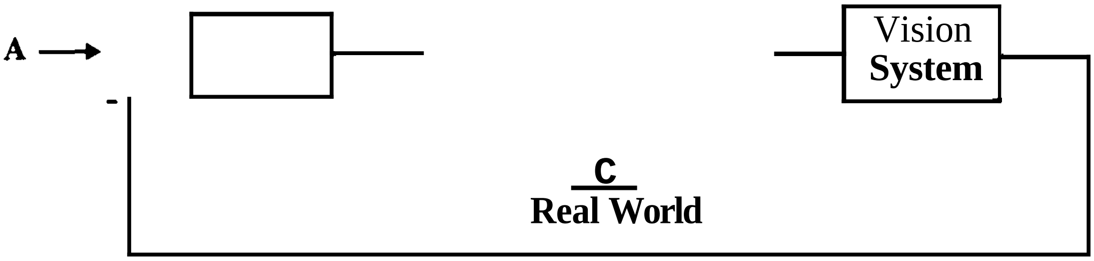

Figure 2. **A** visual servoing control loop. Differences in proposed control schemes include: A-reference inputs given in Cartesian or sensor coordinates, the dimensionality of the control space; B-manipulator commands in given by position or velocity setpoints or torques; C-eye-in-hand or static camera configurations; and D-feature tracking algorithms used.

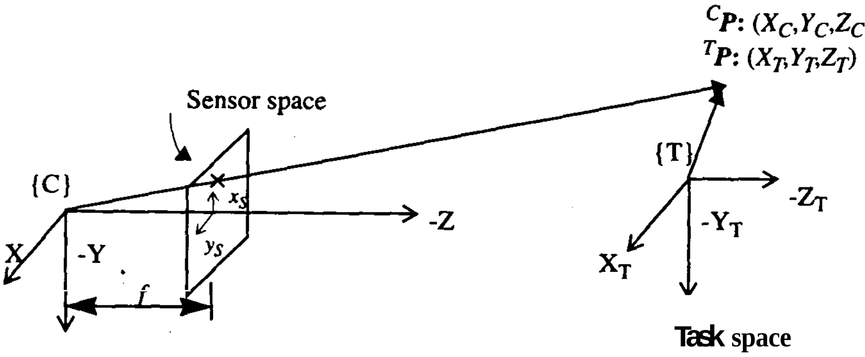

Figure 3. The pinhole camera model with the image plane moved in front of the camera to simplify signs in the equations.

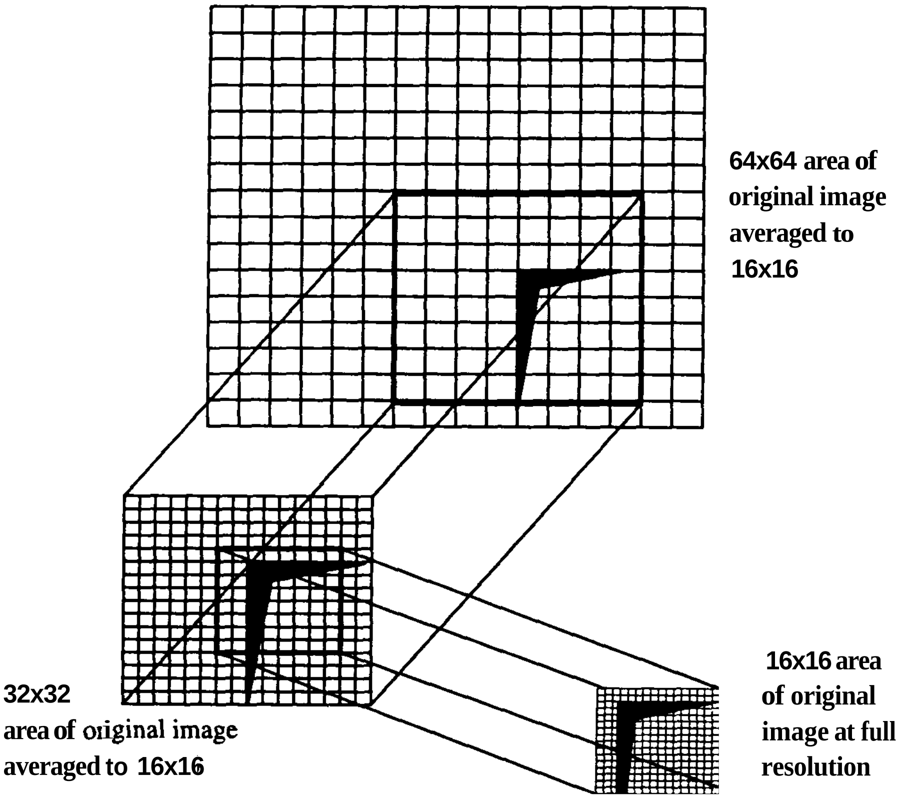

Figure 4. A pyramidal search scheme is used in the SSD optical flow algorithm in order to increase the overall sampling rate of the system.

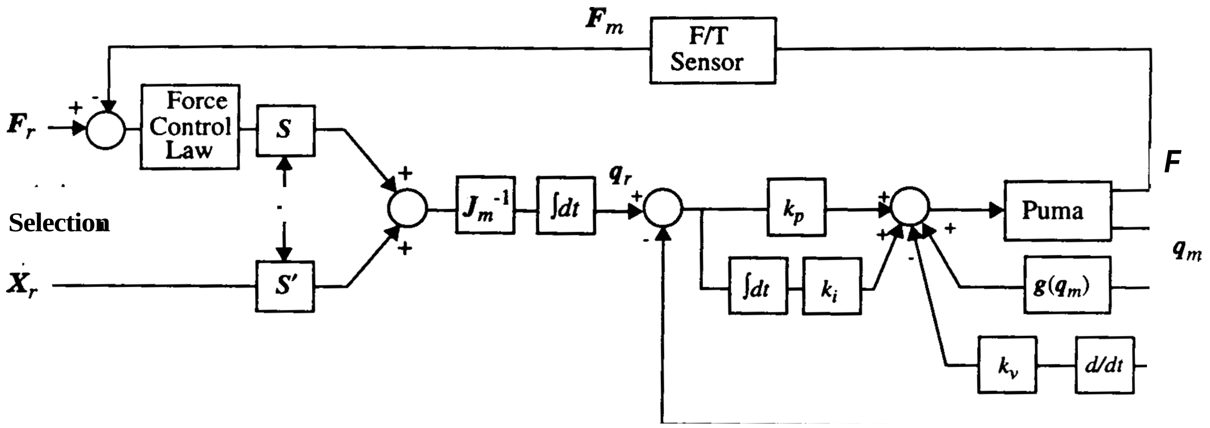

Figure 5. Hybrid force/velocity controller with inner loop PID controllers.

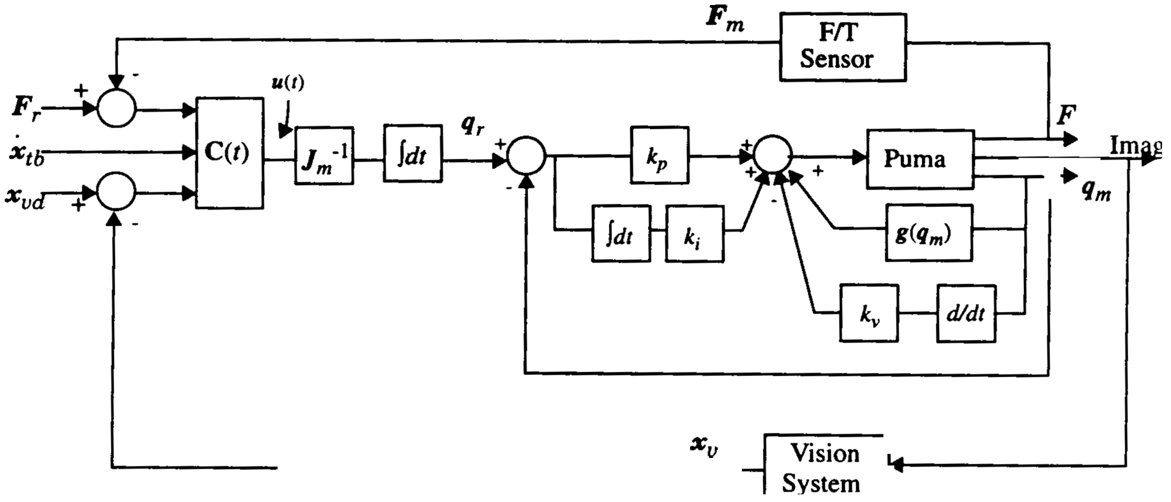

Figure 6. Force and vision in the feedback loop.

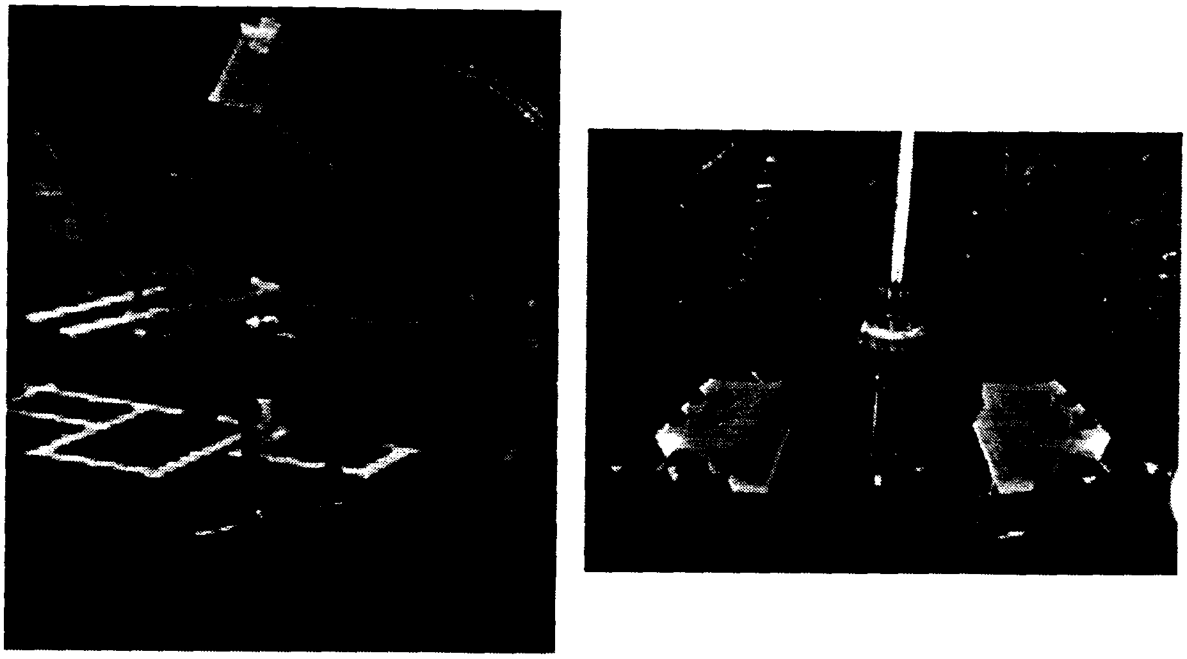

Figure 7. Visual servoing maintains gripper alignment, while a user remotely controls the gripper approach velocity using a hybrid visual servoing control scheme.

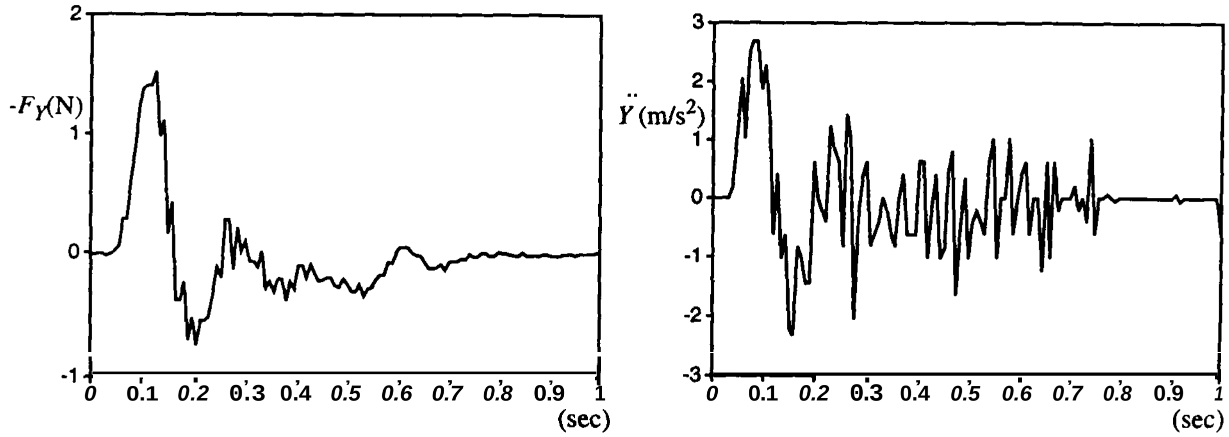

Figure 8. Inertial forces measured by the force sensor and the corresponding measured Cartesian accelerations which induced these forces.

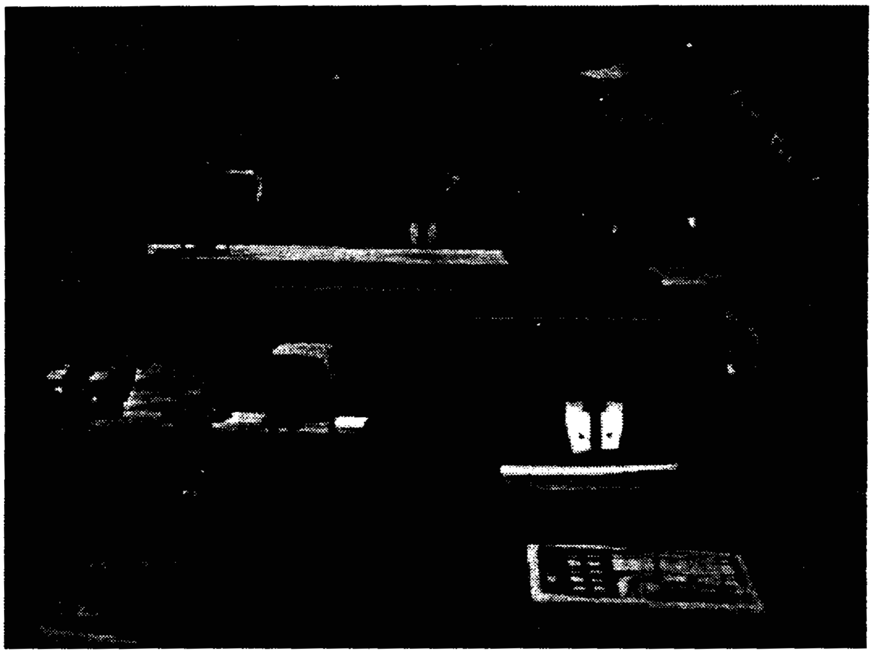

Figure 9. Laboratory setup used for performing vision/force servoing experiments.

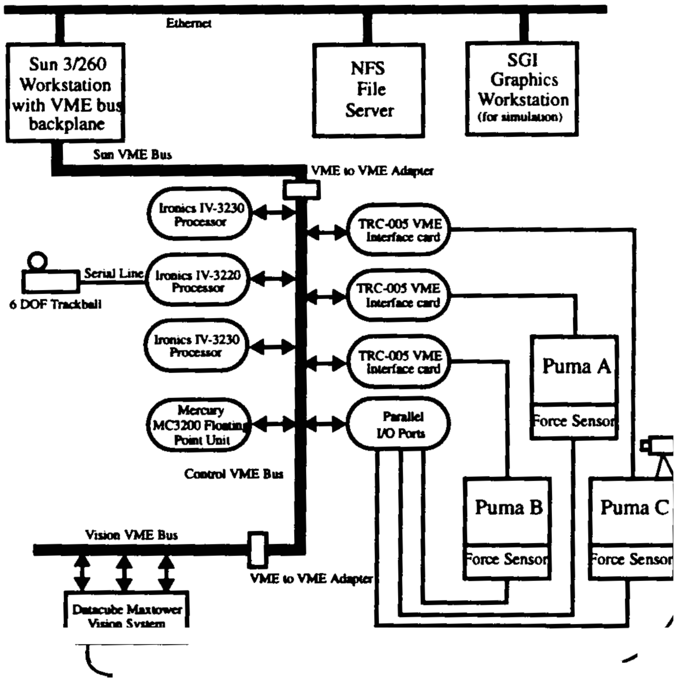
Ethernet
Sun 3/260 Workstation with VME bus backplane
NFS File Server
SGI Graphics Workstation (for simulation)
Sun VME Bus
VME to VME Adapter
Ironics IV-3230 Processor
TRC-005 VME Interface card
Serial Line
Ironics IV-3220 Processor
6 DOF Trackball
TRC-005 VME Interface card
Ironics IV-3230 Processor
TRC-005 VME Interface card
Puma A
Mercury MC3200 Floating Point Unit
Parallel I/O Ports
Force Sensor
Control VME Bus
Puma B
Puma C
Vision VME Bus
Force Sensor
Force Sensor
VME to VME Adapter
Datacube Maxtower Vision System

Figure 10. The Troikabot system architecture.

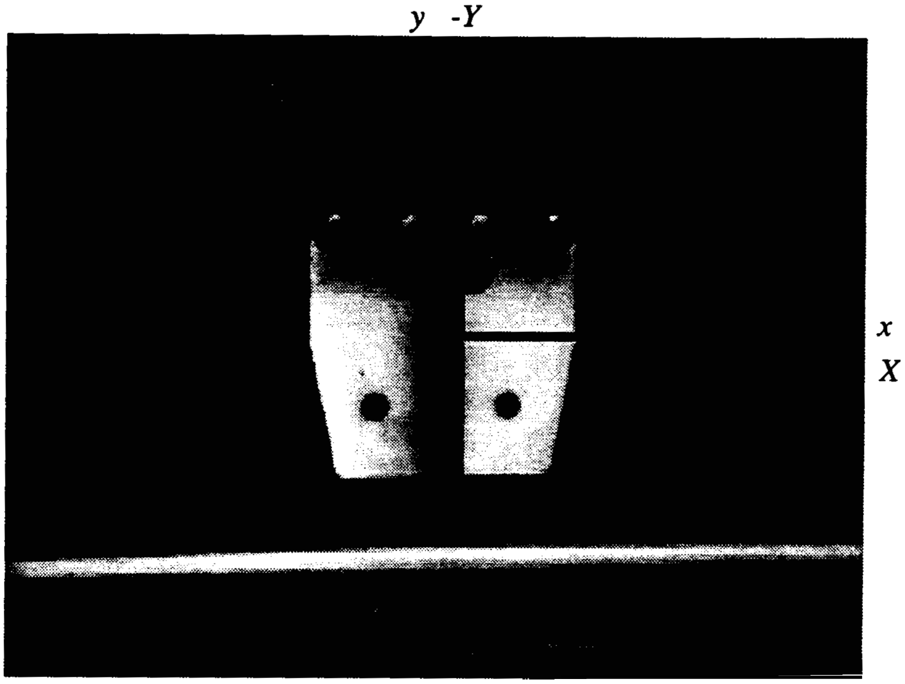

Figure 11. The camera view for visual servoing and coordinate axes on the image plane and in the task frame.

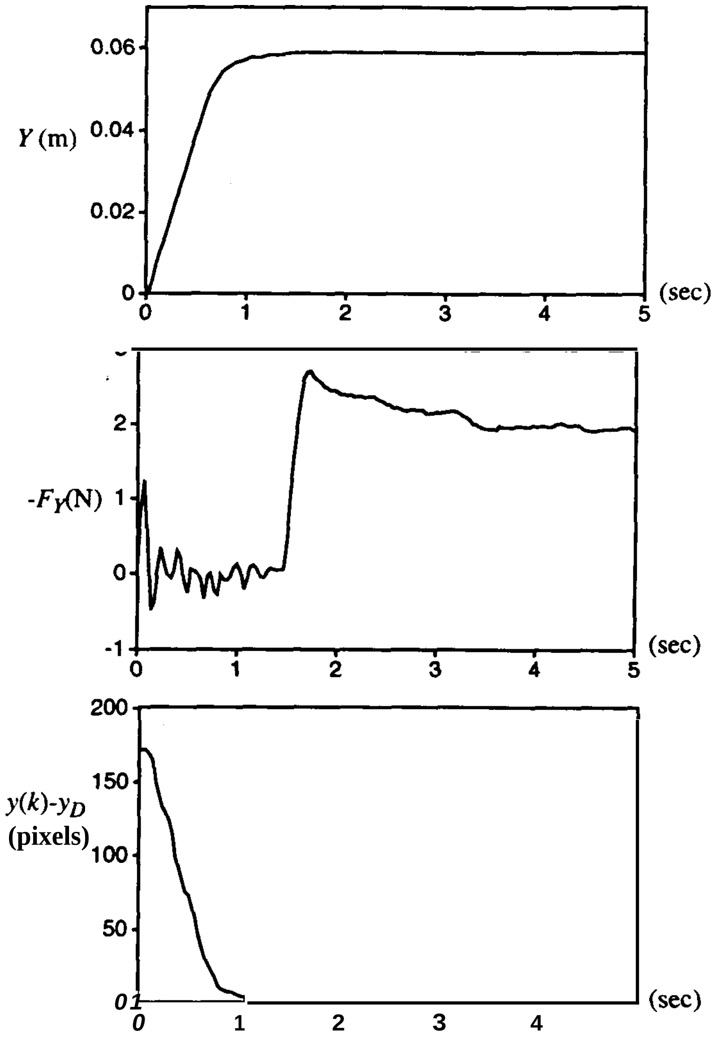

Figure 12. Vertical position of end-effector in Cartesian space, force measured along the vertical direction, and pixel error versus time for traded control.

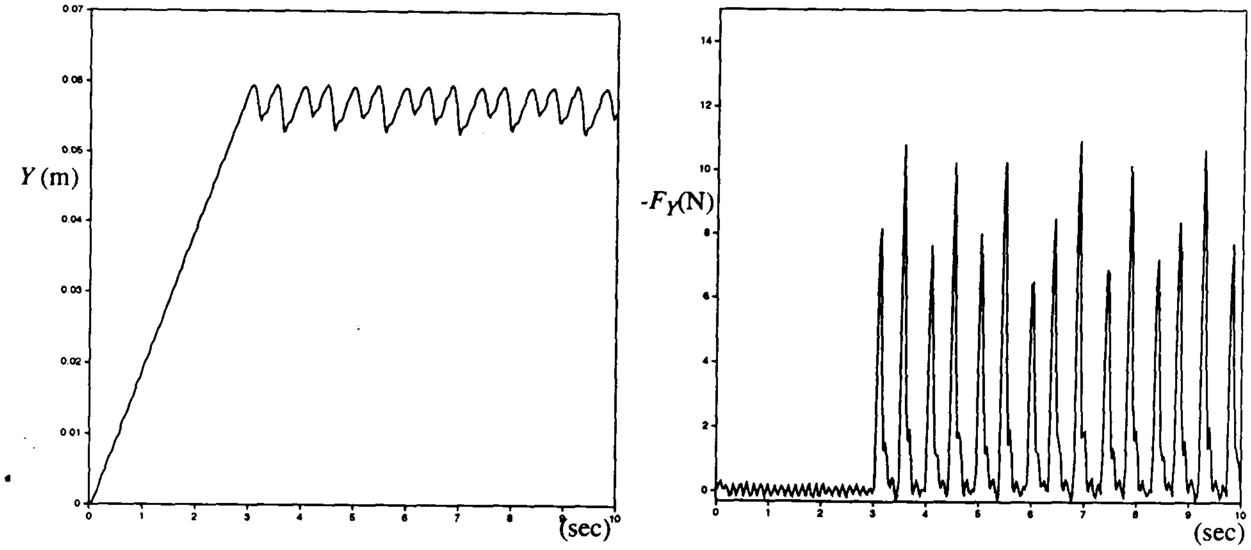

Figure 13. Vertical position of end-effector in Cartesian space and force measured along the vertical direction versus time for simple damping force control upon impact with a surface.

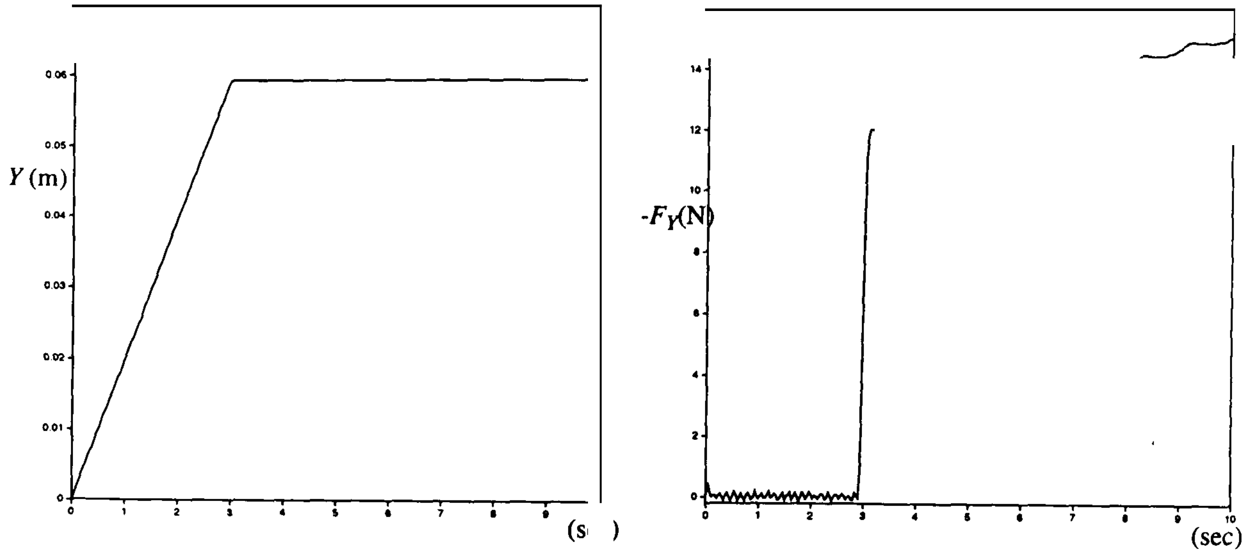

Figure 14. Vertical position of end-effector in Cartesian space and force measured along the vertical direction versus time for a guarded move impact strategy.

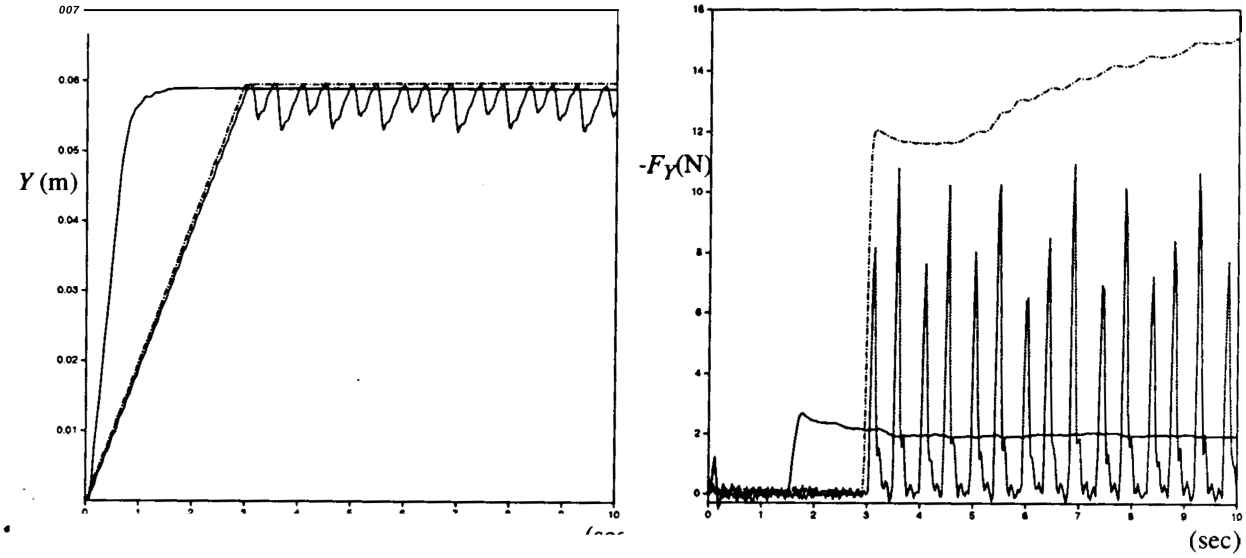

Figure 15. Combined plots of vertical position and measured force for traded visual and force control, damping force control, and a guarded move.

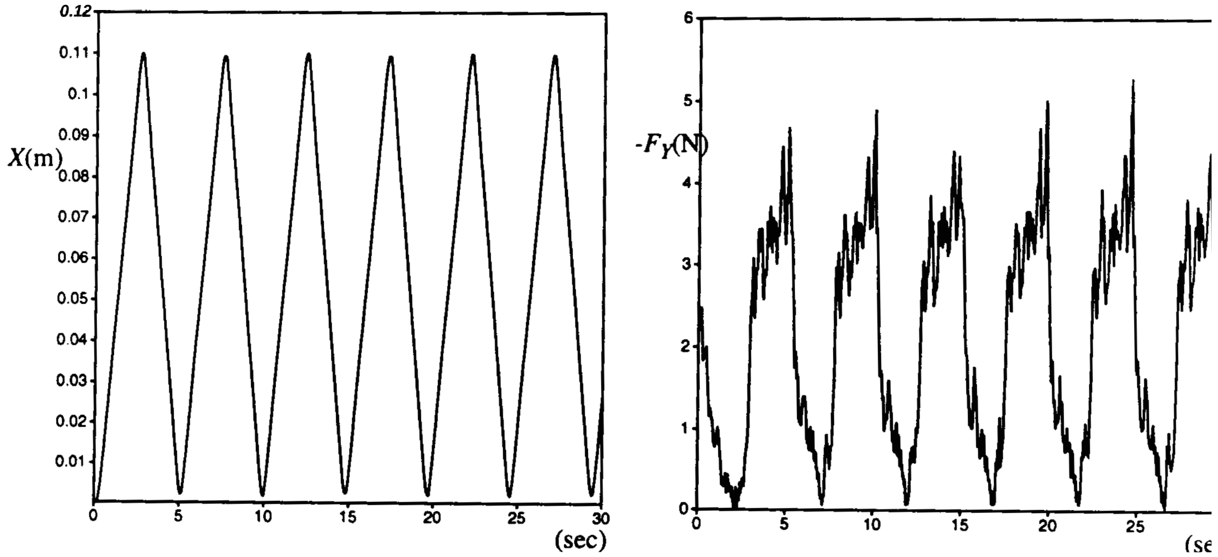

Figure 16. Translational motion along the surface and measured force applied to the surface versus time for hybrid visual/force servoing.

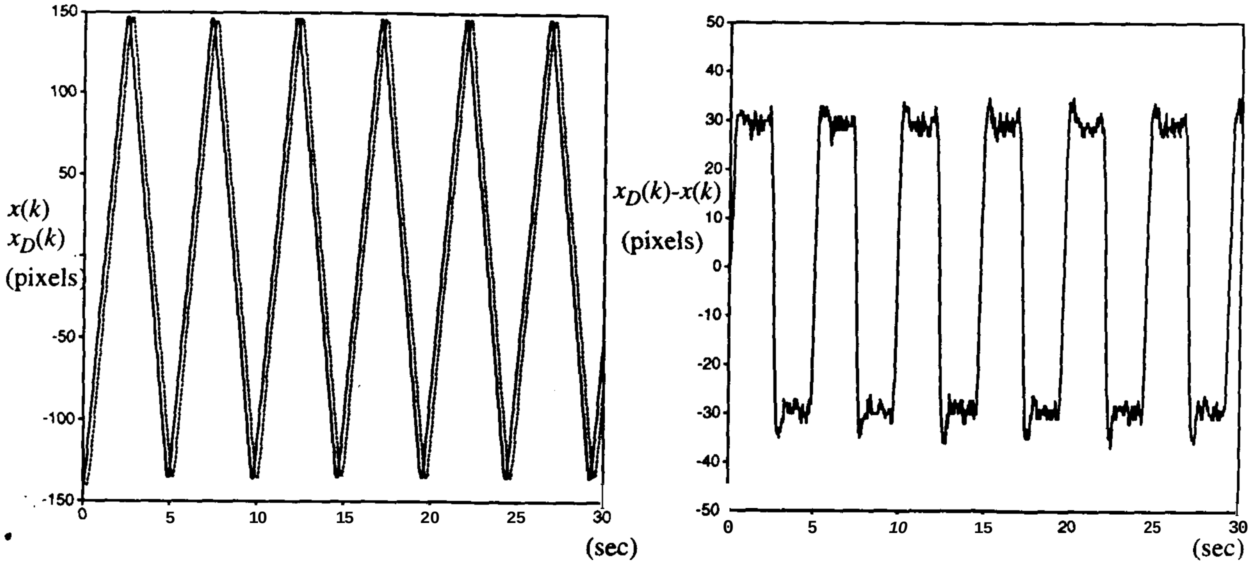

Figure 17. Horizontal motion of the target ($x_D(k)$, represented by the solid line) and of the end-effector ($x(k)$, represented by the dashed line) on the image plane and errors between target motion and end-effector motion.

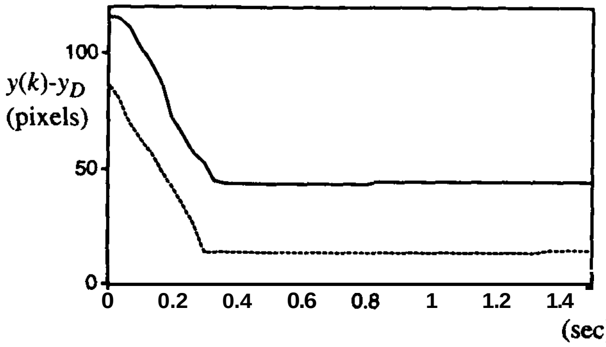

Figure 18. Vertical error between desired and measured end-effector position on the image plane for two examples in which the estimated location of the surface is in error. The solid line corresponds to an error in estimated surface location of 45 pixels, and the dashed line corresponds to an error of 15 pixels.

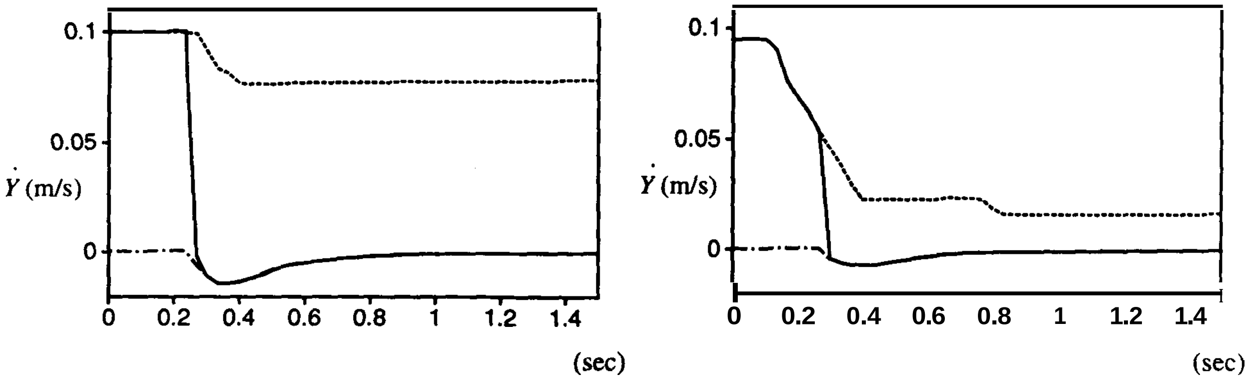

Figure 19. The two plots are of commanded vertical end-effector velocities for shared control for the two examples in Figure 18 in which the estimated position of the surface has significant error. The solid lines represent overall commanded velocity, the dashed lines represent commanded velocity from visual servoing, and the dashed/dotted lines represent commanded velocity from force servoing. The left plot corresponds to an error in surface position of **45** pixels, and the right plot corresponds to **an** error of 15 pixels.

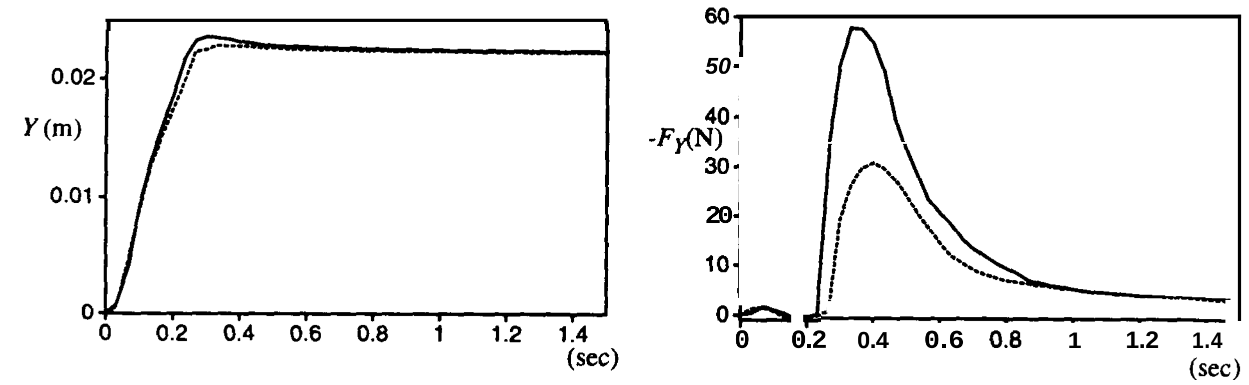

Figure 20. Vertical position of end-effector in Cartesian space and force measured along the vertical direction versus time for shared control. The solid line in both plots correspond to an error in surface position of **45** pixels, and the dashed lines correspond to an error of 15 pixels.