Presented at 1992 IEEE International
Conference on Robotics and
Automation, May 10-15, 1992
Nice, France

SHARED AND TRADED TELEROBOTIC VISUAL CONTROL

N. P. Papanikolopoulos and P. K. Khosla

Department of Electrical and Computer Engineering
The Robotics Institute
Camegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

This paper addresses the problem of integrating the human operator with autonomous robotic visual tracking and servoing modules. A CCD camera is mounted on the end-effector of a robot and the task is to servo around a static or moving rigid target. In manual control mode, the human operator, with the help of a joystick and a monitor, commands robot motions in order to compensate for tracking errors. In shared control mode, the human operator and the autonomous visual tracking modules command motion along orthogonal sets of degrees of freedom. In autonomous control mode, the autonomous visual tracking modules are in full control of the servoing functions. Finally, in traded control mode, the control can be transferred from the autonomous visual modules to the human operator and vice versa. This paper presents an experimental setup where all these different schemes have been tested. Experimental results of all modes of operation are presented and the related issues are discussed. In certain degrees of freedom the autonomous modules perform better than the human operator. On the other hand, the human operator can compensate fast for failures in tracking while the autonomous modules fail. Their failure is due to difficulties in encoding an efficient contingency plan.

1. Introduction

In unstructured and/or hazardous environments such as space. underwater, and radioactive sites, autonomous robots are needed to reliably cope with complex and diverse tasks. Autonomous capabilities are developed through the use of sensors such as force, vision. tactile, etc., that provide information about the state of the task, the robot. and the environment. and through sophisticated algorithms that acquire. interpret, and use sensory information to drive the robot system. Recent advances in sensors. algorithms, and computer technology have enabled the integration of sensors in the feedback loop for real-time task execution. In spite of these advances, present day robotic devices do not possess desirable characteristics such as flexibility, reliability, adaptability, to name a few. and pure autonomy remains an unachieved goal.

The goal of using autonomous robots for applications in unstructured and/or hazardous environments is to remove the human from the task site but not necessarily to remove the human/operator from the task itself. In fact due to safety considerations, it is very important that the human be able to remotely take control of the task and the robot system. For example. unpredictable events and complex evolving environments require the intervention of the human operator. The integration of the human operator in a robotic system results in a system that combines man and machine capabilities and such a system is often called a *telerobotic system* During the last decade, research efforts [1, 2, 3, 4] have focused on building telerobotic systems which provide a wide range of control modes. from pure autonomous control to full teleoperation. Most of these efforts integrate the human operator with force/position robot control schemes. Other sensors, such as CCD cameras, are simply used to provide visual information to the human operator through a video display. In contrast, we view a telerobotic system as a general multi-sensor environment where the human operator and the autonomous sensory modules can cooperate and exchange control of the decision-making process. The exchange of control depends on the specific events which occur during the execution of a task.

In this papa, we address the problem of the integration of a human operator with autonomous visual servoing and tracking modules. In particular, we examine ways that the human operator can cooperate with these modules and improve their performance. Our previous work in visual servoing and tracking emphasized the design of vision and control algorithms for tracking of targets in unknown environments [5, 6.71. The camera was mounted on the robot manipulator (eye-in-hand configuration) and the objective was to keep the projection of the target at some desired position on the image plane. Our work differs significantly from previous work in visual servoing [8, 9] and recursive depth estimation [10, 11]. The differences are in the use of a single moving camera (binocular viewing requires the mounting and the calibration of two cameras which is difficult). the ability to compensate for uncertainties in the relative distance of the moving target from the camera frame and for unknown camera parameters, and the generality of our framework (we have demonstrated full 3-D robotic visual tracking [121).

We present examples of visual servoing around a static or moving target. The autonomous modules arc based on adaptive control techniques especially appropriate for operation in an unknown environment. We assume that we do not know the exact depth map of the targets and that we have a limited knowledge of the camera model and the image noise. In certain cases, the autonomous modules perform better than the human operator. especially when an inexperienced user operates the joystick and interprets the visual data. Another important observation is the way the delays in the human reaction influence the stability of our algorithms. Large delays make the tracking unsuccessful while small delays can be included in the mathematical model, and therefore, an appropriate control scheme can be designed.

The papa is organized as follows: Sdon 2 presents the mathematical modeling of our problem. The different strategies for the solution of the robotic visual tracking and savoing problem arc discussed in Section 3. In Section 4, we present ways for integration of the human operator with the autonomous modules. The architecture of our experimental testbed, CMU DD Arm II, is given in Section 5. The experimental results are presented in Section 6. Finally, in Seaion 7, the paper is summarized.

2. System Modeling

This Scction describes the mathematical modeling of our problem. Due to the fact that the majority of the uacking and servoing examples can be decomposed into the simultaneous tracking of several features, only the tracking and scrvoing model for one feature is presented in this Section.

We assume a pinhole camera model with a frame $(R,)$ atlached to it. In addition, we also assume a perspective projection. Consider a static target with a feature, located at a point' P with coordinates (X_s, Y_s, Z_s) in (R_s). The projection of this point on the image plane is the point p with image coordinates (x, y) given by

$$x = \frac{fX_s}{Z_s s_x} \quad and \quad y = \frac{fY_s}{Z_s s_y} \tag{1}$$

when f is the focal length of the camera and s_x, s_y are the dimensions (mm/pixel) of the camera's pixels. In addition. it is assumed that $Z_s >> f$. If (c_x, c_y) is the origin of the image coordinate system $\{F_a\}$ then

$$x_a = x + c_x \quad and \quad y_a = y + c_y \tag{2}$$

where x_a and y_a arc the actual image coordinates in $\{F_a\}$. Any displacement of a rigid object can be described by a rotation about an axis through the Origin 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. Let us assume that the camera moves in a static environment with a translational velocity $\mathbf{T} = (T_x, T_y, T_z)^T$ and with an angular velocity $\mathbf{R} = (R_x, R_y, R_z)^T$ with respect to the camera frame $\{R_s\}$. The velocity of point P with respect to the $\{R_s\}$ frame is

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

By taking the time derivatives of the expressions for x and y and using (1) and (3) we obtain:

$$u = x\frac{T_z}{Z_s} - \frac{fT_x}{Z_s s_x} + \frac{xys_y}{f}R_x - \left(\frac{f}{s_x} + \frac{x^2 s_x}{f}\right)R_y + \frac{ys_y}{s_x}R_z \tag{4}$$

$$v = y\frac{T_z}{Z_s} - \frac{fT_y}{Z_s s_y} + \left(\frac{f}{s_y} + \frac{y^2 s_y}{f}\right)R_x - \frac{xys_x}{f}R_y - \frac{xs_x}{s_y}R_z \tag{5}$$

when $u = \dot{x}$ and $v = \dot{y}$. u and v are also known as the optical flow measurements. If we assume $s_x = s_y = f = 1$, equations (4)-(5) become:

$$u = [x\frac{T_z}{Z_s} - \frac{T_x}{Z_s} 1 + [xyR_x - (1 + x^2)R_y + yR_z] \tag{6}$$

$$v = [y\frac{T_z}{Z_s} - \frac{T_y}{Z_s}] + [(1 + y^2)R_x - xyR_y - xR_z] \tag{7}$$

To keep the notation simple and without any loss of generality, in the mathematical analysis that follows. we use only the relations described by (6)-(7). Assume that the optical flow of the point p at time kT is $(u(kT), v(kT))$ where T is the time between two consecutive frames. It can be shown [5] that at time kT, the optical flow is:

$$u(kT) = \mu_x u_o(kT) + u_c(kT) \tag{8}$$

$$v(kT) = \mu_y v_o(kT) + v_c(kT) \tag{9}$$

when $u_c(kT), v_c(kT)$ arc the components of the optical flow induced at the time instant kT by the servoing motion of the camera, and where $u_o(kT), v_o(kT)$ arc the components of the optical flow induced at the time instant kT by the possible motion of the target. The coefficients μ_x, μ_y are defined as:

$$\mu_x = \mu_y = \begin{cases} 1 & \text{Moving Target} \\ 0 & \text{Static Target} \end{cases} \tag{10}$$

Equations (8) and (9) will henceforth be used with k instead of kT. Equations (8) and (9) do not include any computational delays that arc associated with the computation and the realization of the servoing motion of the camera. If we include these delays in the model, equations (8) and (9) arc transformed to:

$$u(k) = \mu_x u_o(k) + u_c(k - d + 1) = \mu_x u_o(k) + q^{-d+1} u_c(k) \tag{11}$$

$$v(k) = \mu_y v_o(k) + v_c(k - d + 1) = \mu_y v_o(k) + q^{-d+1} v_c(k) \tag{12}$$

where d is the delay factor ($d \in \{1, 2, \ldots\}$) and q^{-1} is the backward shift operator [13]. For the time being, it is assumed that $d = 1$ From (6) and (7), $u_c(k)$ and $v_c(k)$ are given by:

$$u_c(k) = x(k)\frac{T_z(k)}{Z_s(k)} - \frac{T_x(k)}{Z_s(k)} + x(k)y(k)R_x(k) - [1 + x^2(k)]R_y(k) + y(k)R_z(k) \tag{13}$$

$$v_c(k) = y(k)\frac{T_z(k)}{Z_s(k)} - \frac{T_y(k)}{Z_s(k)} + [1 + y^2(k)]R_x(k) - x(k)y(k)R_y(k) - x(k)R_z(k) \tag{14}$$

In addition, it is known that:

$$u(k) = \frac{x(k+1) - x(k)}{T} \tag{15}$$

$$v(k) = \frac{y(k+1) - y(k)}{T} \tag{16}$$

If we substitute $u(k)$ and $v(k)$ in (11) and (12) with their equivalent expressions from (15) and (16). then equations (11) and (12) can be written as:

$$x(k+1) = x(k) + Tu_c(k) + \mu_x T u_o(k) + v_x(k) \tag{17}$$

$$y(k+1) = y(k) + Tv_c(k) + \mu_y T v_o(k) + v_y(k) \tag{18}$$

when the white noise terms $v_x(k)$ and $v_y(k)$ are included io model the inaccuracies of the model (neglected accelerations. inaccurate robot control, etc.). $v_x(k), v_y(k)$ are zero-mean. mutually uncorrelated. stationary random variables with variances σ_x^2 and σ_y^2, respectively. For every feature point we obtain two equations that relate the new feature coordinates to the previous coordinates in terms of the sampling time (T) and optical flow. Equations (17) and (18) can be represented compactly in matrix-vector form (also known as state-space form) as:

$$\mathbf{x}_F(k+1) = \mathbf{A}_F(k)\mathbf{x}_F(k) + \mathbf{B}_F(k)\mathbf{u}_c(k) + \mathbf{E}_F(k)\mathbf{u}_F(k) + \mathbf{H}_F(k)\mathbf{v}_F(k) \tag{19}$$

where[**] $\mathbf{A}_F(k) = \mathbf{H}_F(k) = \mathbf{I}_2$, $\mathbf{E}_F(k) = T\,diag\{\mu_x, \mu_y\}$, $\mathbf{x}_F(k) \in R^2$, $\mathbf{u}_c(k) \in R^6$, and $\mathbf{v}_F(k) \in R^2$. The matrix $\mathbf{B}_F(k) \in R^{2 \times 6}$ is:

$$\mathbf{B}_F(k) = T \begin{bmatrix} \frac{-1}{Z_s(k)} & 0 & \frac{x(k)}{Z_s(k)} & x(k)y(k) & -(1 + x^2(k)) & y(k) \\ 0 & \frac{-1}{Z_s(k)} & \frac{y(k)}{Z_s(k)} & (1 + y^2(k)) & -x(k)y(k) & x(k) \end{bmatrix} \tag{20}$$

The vector $\mathbf{x}_F(k) = (x(k), y(k))^T$ is the state vector, $\mathbf{u}_c(k) = (T_x(k), T_y(k), T_z(k), R_x(k), R_y(k), R_z(k))^T$ is the control input vector. $\mathbf{u}_F(k) = (u_o(k), v_o(k))^T$ is the disturbance vector. and $\mathbf{v}_F(k) = (v_x(k), v_y(k))^T$ is the white noise vector. The measurement vector $\mathbf{y}_F(k) = (y_x(k), y_y(k))^T$ for this feature is given by:

$$\mathbf{y}_F(k) = \mathbf{C}_F\mathbf{x}_F(k) + \mathbf{w}_F(k) \tag{21}$$

when $\mathbf{w}_F(k) = (w_x(k), w_y(k))^T$ is a white noise vector ($\mathbf{w}_F(k) \sim N(\mathbf{0}, \mathbf{W})$) and $\mathbf{C}_F = \mathbf{I}_2$. The elements of the covariance matrix $\mathbf{W}$ are set to some constant and nominal values. Plausible estimates of these elements can be computed from the image [14]. The measurement vector is computed using the SSD algorithm which is described in [5]. In the next Section. we will examine the autonomous control and estimation techniques that compute the control commands to the robotic system.

[*] Bold symbols denote vectors or matrices.

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

3. 3-D Visual Servoing Around a Static/Moving Target

This Section examines the control strategies that realize the scrvoing and tracking motion. the estimation scheme used to estimate the unknown parameters of the model and the ways we can integrate the human operator in the feedback loop. First. we present the control structure for servoing around a static target, and then we present the structure for tracking a target that moves with 3-D translational motion It is assumed that the dcpth $Z_s(k)$ and the depth related parameters (in (19) and (20)) arc unknown or known inaccurately. In other words, we initialize our system with some estimates of the depth and the dcpth related parameters. The values of these parameters can be different from the actual values by a factor of 2 to 3. This fad allows for the use of our algorithms in poorly calibrated environments like underwater, space. and nuclear sites. The dcpth and the depth related parameters are estimated on-line by using the motion of the camera-robot system and the possible motion of the target.

3.1. Visual Servoing Around a Static Target

The first example is visual scrvoing around a static target. First. we present the autonomous visual servoing modules that perform this task, and then, in Section 6, we compare the results with the results that a human operator can achieve. The control objective is to move the features' projections on the image plane to some desired positions. The repositioning of the projections is realized by an appropriate motion of the system robot-camera. In [9], it is proved that at least three feature points are needed, especially when a certain pose is not required. If a certain pose is required [15], more than three feature points are needed. We assume that the target is stationary and therefore $\mu_x = \mu_y = 0$. The measurement vector $y(k)$ is composed of the positions of three feature points at every instant k. A simple control law can be derived by the minimization of a cost function J which includes the control signal:

$$J(k+1) = E\{[\mathbf{y}(k+1) - \mathbf{y}^*(k+1)]^T \mathbf{Q} [\mathbf{y}(k+1) - \mathbf{y}^*(k+1)] + \\ + \mathbf{u}_c^T(k)\, \mathbf{L}\, \mathbf{u}_c(k) | F_k\} \qquad (22)$$

where the symbol $E\{X\}$ denotes the expected value of the random variable X and F_k denotes data (past measurements and control inputs) up to time k. The vector $y^*(k)$ represents the desired positions of the projections of the three features on the image plane. In our experiments, the vector $\mathbf{y}^*(k)$ is known *a priori* and is constant over time. The control law is:

$$\mathbf{u}_c(k) = -[\hat{\mathbf{B}}^T(k)\, \mathbf{Q}\, \mathrm{B}(k) + \mathbf{L}]^{-1} \hat{\mathbf{B}}^T(k)\, \mathbf{Q} [\mathbf{y}(k) - \mathbf{y}^*(k+1)] \qquad (23)$$

where $\hat{\mathbf{B}}(k)$ is the estimated value of the matrix $\mathrm{B}(k)$. The design parameters in this control law are the elements of the matrices Q , L. The matrix $\hat{\mathrm{B}}(k)$ is depmdent on the estimated values of the features' depth $\hat{Z}_s^{(j)}(k)$ $\{((j) \in ((1), (2), (3)))\}$ and the coordinates of the features' image projections. In particular, the matrix $\hat{\mathrm{B}}(k)$ is defined as follows:

$$\mathrm{B}(k) = \begin{bmatrix} \hat{\mathbf{B}}_F^{(1)}(k) \\ \hat{\mathbf{B}}_F^{(2)}(k) \\ \hat{\mathbf{B}}_F^{(3)}(k) \end{bmatrix}$$

where $\hat{\mathbf{B}}_F^{(j)}(k)$ is given by:

$$\hat{\mathbf{B}}_F^{(j)}(k) = T \begin{bmatrix} \dfrac{-1}{\hat{Z}_s^{(j)}(k)} & 0 & \dfrac{x^{(j)}(k)}{\hat{Z}_s^{(j)}(k)} & x^{(j)}(k)\, y^{(j)}(k) & -[1+(x^{(j)}(k))^2] & y^{(j)}(k) \\ 0 & \dfrac{-1}{\hat{Z}_s^{(j)}(k)} & \dfrac{y^{(j)}(k)}{\hat{Z}_s^{(j)}(k)} & [1+(y^{(j)}(k))^2] & -x^{(j)}(k)\, y^{(j)}(k) & -x^{(j)}(k) \end{bmatrix}$$

The estimation of the feature's depth $Z_s^{(j)}(k)$ with respect to the camera frame can be done in multiple ways We define the inverse of the depth $Z_s^{(j)}(k)$ as $\zeta_s^{(j)}(k)$. Then. equations (19)-(21) of each feature point can be rewritten as:

$$\mathbf{y}_F^{(j)}(k) = \mathbf{A}_F^{(j)}(k-1)\, \mathbf{y}_F^{(j)}(k-1) + \zeta_s^{(j)}(k-1)\, \mathbf{B}_{F1}^{(j)}(k-1)\, \mathrm{T}(k-1) + \\ + \mathbf{B}_{F2}^{(j)}(k-1)\, \mathrm{R}(t-1) + \mathbf{n}_F^{(j)}(k) \qquad (24)$$

when $\mathrm{B}_{F1}^{(j)}(k)$, $\mathrm{B}_{F2}^{(j)}(k)$ arc given by:

$$\mathbf{B}_{F1}^{(j)}(k) = T \begin{bmatrix} -1 & 0 & x^{(j)}(k) \\ 0 & -1 & y^{(j)}(k) \end{bmatrix},$$

$$\mathbf{B}_{F2}^{(j)}(k) = T \begin{bmatrix} x^{(j)}(k)\, y^{(j)}(k) & -[1+(x^{(j)}(k))^2] & y^{(j)}(k) \\ [1+(y^{(j)}(k))^2] & -x^{(j)}(k)\, y^{(j)}(k) & -x^{(j)}(k) \end{bmatrix},$$

and the vector $\mathbf{n}_F^{(j)}(k)$ is a gaussian noise vector with mean 0 and covariance $\mathbf{N}^{(j)}(k)$. By defining $\mathbf{u}_t^{(j)}(k)$ and $\mathrm{u}_r^{(j)}(k)$ as $\mathbf{u}_t^{(j)}(k) = \mathrm{B}_{F1}^{(j)}(k)\, \mathrm{T}(k)$ and $\mathrm{u}_r^{(j)}(k) = \mathbf{B}_{F2}^{(j)}(k)\, \mathbf{R}(k)$, respectively, equation (24) is transformed into:

$$\mathbf{y}_F^{(j)}(k) = \mathbf{A}_F^{(j)}(\&-1)\, \mathbf{y}_F^{(j)}(k-1) + \zeta_s^{(j)}(k-1)\, \mathbf{u}_t^{(j)}(k-1) + \\ + \mathbf{u}_r^{(j)}(k-1) + \mathbf{n}_F^{(j)}(k) \qquad (25)$$

A last transformation of equation (25) is done by using the vector $\Delta \mathbf{y}_F^{(j)}(k)$ which is defined as:

$$\Delta \mathbf{y}_F^{(j)}(k) = \mathbf{y}_F^{(j)}(k) - \mathbf{y}_F^{(j)}(k-1) - \mathbf{u}_r^{(j)}(k-1)$$

The new form of equation (25) is:

$$\Delta \mathbf{y}_F^{(j)}(k) = \zeta_s^{(j)}(k-1)\, \mathbf{u}_t^{(j)}(k-1) + \mathbf{n}_F^{(j)}(k) \qquad (26)$$

The vectors $\Delta \mathbf{y}_F^{(j)}(k)$ and $\mathbf{u}_t^{(j)}(k-1)$ are known every instant of time, while the scalar $\zeta_s^{(j)}(k)$ (which is the inverse of the unknown depth $Z_s^{(j)}(k)$) is continuously estimated. It is assumed that an initial estimate $\hat{\zeta}_s^{(j)}(0)$ of $\zeta_s^{(j)}(0)$ is given and $p^{(j)}(0) = E\{[\zeta_s^{(j)}(0) - \hat{\zeta}_s^{(j)}(0)]^2\}$ is a positive scalar p_0. $p^{(j)}(0)$ can be interpreted as a measure of the confidence that we have in the initial estimate $\hat{\zeta}_s^{(j)}(0)$. Accurate knowledge of the scalar $\zeta_s^{(j)}$ corresponds to a small covariance scalar p_0. In our examples. $\mathrm{N}^{(j)}(k)$ is a constant predefined matrix. In addition, for simplicity in notation, $\mathrm{h}(k)$ is used instead of $\mathbf{u}_t^{(j)}(k)$.

The estimation equations arc (the superscript $(-)$ denotes the predicted value of a variable while the superscript $(+)$ denotes its updated value) [16]:

$$^{-}\hat{\zeta}_s^{(j)}(k) = {}^{+}\hat{\zeta}_s^{(j)}(k-1) \qquad (27)$$

$$^{-}p^{(j)}(k) = {}^{+}p^{(j)}(k-1) + s^{(j)}(k-1) \qquad (28)$$

$$^{+}p^{(j)}(k) = [\{^{-}p^{(j)}(k)\}^{-1} + \mathbf{h}^T(k-1)\{\mathrm{N}^{(j)}(k)\}^{-1}\mathbf{h}(k-1)]^{-1} \qquad (29)$$

$$\mathbf{k}^T(k) = {}^{+}p^{(j)}(k)\, \mathbf{h}^T(k-1)\{\mathrm{N}^{(j)}(k)\}^{-1} \qquad (30)$$

$$^{+}\hat{\zeta}_s^{(j)}(k) = {}^{-}\hat{\zeta}_s^{(j)}(k) + \mathbf{k}^T(k)[\Delta \mathbf{y}_F^{(j)}(k) - {}^{-}\hat{\zeta}_s^{(j)}(k)\, \mathbf{h}(k-1)] \qquad (31)$$

when $s^{(j)}(k)$ is a covariance scalar which corresponds to the white noise that characterizes the transition between the states. The depth related parameter $\zeta_s^{(j)}(k)$ is a time-varying variable since the camera translates along its optical axis and rotates along the X and Y axis. The estimation scheme of equations (27)-(31) can compensate for the time-varying nature of $\zeta_s^{(j)}(k)$ because it is designed under the assumption that the estimated variable undergoes a random change. One problem is to keep the covariance scalar $p^{(j)}(k)$ finite. Solutions for this type of problem can be found in [13]. In addition, we implemented estimation techniques such as *exponential data weighting* and *covariance resetting* [13] which deal with lime-varying parameters.

3.2. Visual Servoing Around a Target with 3-D Translational Motion

This Section presents our strategies for visual tracking of a moving target. Let us assume that the target moves with 3-D translational motion and the objective is to keep the target's projection on the image plane stationary while the target moves. Therefore, we must only compute $T_x(k)$, $T_y(k)$, and $T_t(k)$. Due to the fact that the target is moving. we assume $\mu_x = \mu_y = 1$. In order to simplify the modeling of the noise. white noise terms are assumed to accompany the terms $u_o(k)$ and $v_o(k)$. Therefore, we transform equations (17) and (18) ($n_{mx}(k)$, $n_{my}(k)$ are the white noise terms) into the following equations for a single feature point $(R_r(k)=R_y(k)=R_z(k)=0)$:

$$x(k+1)=x(k)+b_xS_x(k)+Tu_o(k)+(T^2/2)n_{mx}(k) \tag{32}$$

$$y(k+1)=y(k)+b_yS_y(k)+Tv_o(k)+(T^2/2)n_{my}(k) \tag{33}$$

$$u_o(k+1)=u_o(k)+Tn_{mx}(k) \tag{34}$$

$$v_o(k+1)=v_o(k)+Tn_{my}(k) \tag{35}$$

where $S_x(k)$ and $S_y(k)$ are defined as:

$$S_x(k)=-T_x(k)+x(k)T_z(k)$$

$$S_y(k)=-T_y(k)+y(k)T_z(k)$$

and coefficients b_x and b_y depend on the values of the dcpth Z_s and of the sampling interval T. For M feature points, we have M sets of the equations (32)-(35).

The next step in our modeling is to remove the terms $u_o(A)$ and $v_o(k)$ from our formulation. In order to achieve that, we derive two time-varying SISO (Single-Input Single Output) ARMAX (AutoRegressive Moving-Average model with eXternal input) models (the ARMAX [13] model in this case can be viewed as a compact way of writing down the innovations model). The ARMAX models are derived by subtractions and additions of the expressions for $x(k)$, $x(k-1)$ and $x(k-2)$, and $y(k)$, $y(k-1)$ and $y(k-2)$, respectively. For each feature, the corresponding two time-varying SISO ARMAX model are:

$$A_i(q^{-1})y_i(k)=q^{-d}B_i(q^{-1})u_{ci}(k)+C_i(q^{-1})w_i(k) \quad k \geq 0 \quad i=1,2 \tag{36}$$

where

$$A_i(q^{-1})=1+a_{i1}q^{-1}+a_{i2}q^{-2} \quad i=1,2$$

$$B_i(q^{-1})=b_{i0}+b_{i1}q^{-1} \quad i=1.2$$

$$C_i(q^{-1})=1+c_{i1}q^{-1}+c_{i2}q^{-2} \quad i=1,2$$

The values of the coefficients $a_{i1}, a_{i2}, b_{i0}, b_{i1}, c_{i1}, c_{i2}$ depend on the values of T, b_x, and b_y. The noise sequences $w_i(k)$ are assumed to satisfy the assumptions

$$E\{w_i(k) \mid F^i_{k-1}\}=0 \quad E\{w_i^2(k) \mid F^i_{k-1}\}=\sigma_i^2 \quad i=1,2$$

where the symbol $E\{X\}$ denotes the expected value of the random variable X and F^i_{k-1} denotes data (past control inputs and measurements) up to time $k-1$. The index i corresponds to the two different SISO ARMAX models per feature point, the scalar input $u_{ci}(k)$ now represents either $S_x(k)$ or $S_y(k)$. and the scalar $y_i(k)$ corresponds to the measured deviation of the feature point from its desired position in one of the X or Y directions. It can be shown that $b_{10}=-b_{11}=b_x$, and $b_{20}=-b_{21}=b_y$.

The control and estimation techniques which are used in order to compute an efficient adaptive control law can be found in [7].

4. Issues in Shared and Traded Control

In shad control mode. some degrees of freedom (DOF) are commanded by the human operator and some by the autonomous modules. In traded control mode, there is a transition from the autonomous modules to the human operator and vice vena. There are. a lot of different ways of looking at the design of modules that suppon these modes. In [2], mixing matrices and weighs are used to decide when and how the autonomous modules will mix with the human operator. On the other hand, in [17, 3], the human operator and the autonomous modules are not allowed to command the Same degree of freedom (DOF). The reason is to avoid the "fork in the road" problem which may occur when the human operator and the autonomous modules suggest opposite directions of motion along a specific degree of freedom (DOF).

In order to avoid the "fork in the road" problem, we choose to follow the latter approach. As was mentioned in Section 2, the control input signal $u_c(k)$ is expressed with respect to the camera frame (R_s) Let us assume that the human operator commands a speed signal $^j u_c(k)$ with respect to the joystick frame. The speed signal $^j u_c(k)$ can be transformed through the transformation $^s T_j$ to the camera frame signal $^m u_c(k)$. We define the matrices Ψ, and Ω (Ψ, $\Omega \in R^{6x6}$):

$$\Psi(\alpha,\beta)=\begin{cases} 1 & \text{if } a=\beta \text{ \& dof}(\alpha)=1 \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

$$\Omega(\alpha,\beta)=\begin{cases} 1-\Psi(\alpha,\beta) & \text{if } \alpha=\beta \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

when $dof(\alpha)=1$ ($a \in \{1,\ldots,6\}$) implies that the human operator controls the a degree of freedom (DOF). The new rate reference signal $^s u_c(k)$ is expressed in camera frame coordinates as:

$$^s u_c(k)=Q u_c(k)+\Psi\, ^m u_c(k) \tag{39}$$

One of the possible problems of shared control mode is the possible coupling between the degrees of freedom (DOF) that the human operator commands and the degrees of freedom commanded by the autonomous modules. For example. in our system, there is a strong coupling between $T_x(k)$ and $R_y(k)$, and between $T_y(k)$ and $R_x(k)$. Therefore, a modification of one of them makes necessary a modification of the other's value. The implementation of this scheme depends on the knowledge of the human operator about the type of coupling The selection of the degrees of freedom (DOF) which are manually commanded is not an easy task. The way that the CCD camera is mounted on the end-effector and the 2-D nature of the visual information force the human operator to select $R_z(k)$, or T_x and $T_y(k)$, as possible teleoperated degrees of freedom (DOF). This observation is verified by the experimental results which are presented in Section 6. The experimental results show that manual control of $T_z(k)$ or $R_x(k)$ and $R_y(k)$ gives poor tracking performance.

Another potential problem is the delay in the manual rate signal. If d_m is a delay factor associated with the transmission of the manual rate signal then equation (39) becomes:

$$^s u_c(k)=\Omega u_c(k)+\Psi\, ^m u_c(k-d_m) \tag{40}$$

Large transmission delays which are likely to be present for space telerobots can significantly influence the tracking performance and the stability of our algorithms. The estimation schemes are influenced too, due to the fact that we apply different inputs to the system than the ones that the estimation schemes consider as current. Therefore. the transmission delays must be taken into consideration during the design phase of the whole system.

The traded control mode is useful during three phases; a) Initialization, b) Emergency events, and c) Final stage. During these phases, there is an exchange of control between the human operator and the autonomous modules. In particular. during Initialization. the human operator grossly positions the manipulator with respect to the target and selects a number of candidate features for tracking. Then, he/she passes control to the autonomous modules During Emergency events (e.g. loss of one feature due to occlusion by an unknown object, singular *configuration* of the manipulator), the human operator takes control and tries to solve the accumulated problems. If the problems arc solved, then he/she again passes control to the autonomous modules. Finally. during the Final stage, the human operator takes control of the system and places the manipulator in its *home* position The next S d o n describes the architecture of *our* experimental setup. CMU DD Arm IL

5. System Architecture

The vision and joystick modules of the CMU DD *Arm* II (Direct Drive Arm II) system arc parts of a bigger hardware environment which runs under the CHIMERA II [18] real-time Operating system. The IDAS/150 image processing system carries out all the computational load of the image processing calculations while the Mercury Floating Point *Unit* does all the control calculations *An* Ironics board is responsible for the realization of the shared control planner. The IDAS/150 contains a Heurikon 68030 board as the controller of the vision module and two floating point boards, each one with computational power of 20 Mflops. The system *can* be expanded *to* contain as many as *eight* of these boards. The *six* degrees of freedom joystick is a Dimension *6* TrackBall [19, 17]. The applied forces and toques to the trackball arc transformed to *six* reference signals which are computed at once. The reference signals correspond to either velocities or forces. The human Operator *can* select the type and the reference frame of the reference signals on-line. In *our* experiments, the user commands motions with respect to the joystick frame. and then these motions are transformed in *camera* frame coordinates. The camera frame is parallel to the end-effector frame.

The software is organized around 5 processes:

- *Visioo* process. This process docs all the image processing calculations and has a period of 150ms.
- Interpolation process. This program reads the data from the vision system. interpolates the data and sends the reference signals to the robot cartesian controller. It has a period of 5ms.
- Robot controller process. This process drives the robot and has a period of 3.33 ms. The robot control scheme that is used is a cartesian PD controller with gravity compensation.
- Joystick process. This process reads the data from the joystick and does all the necessary data transformations. It has a period of 30 ms.
- Joystick reference process. This process interpolates the data from the joystick process in order to guarantee a smooth robot trajectory. Its period is 9 ms.

The next Section describes the experimental results.

6. Experimental Results

A number of experiments were performed on our experimental testbed, the CMU DD *Arm* II robotic system. A camera, whose parameters are given in Table 1, is mounted on the end-effector. The operator by using the joystick commands motions in the end-effector frame which is parallel to the camera frame. The objects are static *or* moving (the initial depth of the objects' center of mass with respect to the camera frame Z_s varies from *500* mm to 1000 mm). The maximum permissible translational veloaty of the end-effector is 10 cm/sec and *each* one of the components (roll. pitch, yaw) of the end-effector's rotational velocity does not exceed 0.05 rad/sec.

The objective of the first experiment was to move the manipulator so that the image projections of certain chosen features of the object move to some desired image positions. We tried to operate the system first in autonomous mode and then in manual mode. The results are plotted in Figures 3-5. The user. by using the mouse, proposes to h e system some of the object's features *that* he is interested in. Then, the syaem evaluates on-line the quality of the features, based on the confidence measures described in [5]. The same operation can be done automatically by a computer process that runs once and needs between 2 and 3 minutes, depending on the size of the interest operators which are used. The three best features are selected and used for the robotic visual servoing task The size of windows is 10x10. In addition, the user selects the desired positions of these features. The current and the desired positions are continuously highlighted in order to help the human operator accomplish his/her task. The system provides messages about the current servoing errors and the timing of the task. These graphical *aids* are realized on a graphical overlay of the video display which has a frame rate of 30 frames/sec.

The gain matrices for the autonomous visual control modules are $Q = \mathbf{I}_6$ and $\mathbf{L} = diag(0.025, 0.025, 0.25, 2x\ 10^5, 2x\ 10^5, 2x\ 10^5)$. The computation of the $[\hat{\mathbf{B}}^T(k)\,Q\,B(k) + L]^{-1}$ matrix is done on a Heurikon 68030 board. To reduce the computational load of the matrix inversion. we use the techniques described in [9]. The knowledge of the depth Z_s is assumed *to* be inaccurate. As shown in Figures 3-5, the results from the autonomous modules are far better than h e results that the human operator achieves. The results of the human operator show large delays and steady-state errors. Especially in figure 5, one *can* observe the large steady state error of the X component of the tracking error when the human operator is in charge (manual mode). One reason for the bad performance of the human operator is his/her lack of understanding of the depth. The images in the display give a good idea about motions in 2-D but not in 3-D. The large delays arc associated with the delays that the human operator introduces in the control loop (delays of reaction). These delays arc increased when the human operator is inexperienced which in certain cases results in unsuccessful completion of the task.

The second experiment involved tracking of a moving target *(3-D* translational motion) under autonomous operation and also mixed operation. In these experiments, we used four features and selected the two best based *on* the confidence measures described in [5]. The experimental results of the 3-D *case* are plotted in Figures 6-13. The knowledge of the depth Z_s is assumed to be inaccurate. The property of our algorithms to compensate for uncertain depth and camera parameters is extremely valuable due to the fact that in certain robotic applications (space, underwater) we do not have to ability to accurately calibrate the operational space. The *MezP* vector represents the position of the end-effector with respect to the world frame. In particular. in Figures 6-13 the three dotdashed trajectories correspond to the trajectories of the center of mass of the object in 3-D. In Figures 7, 9, 11, and 13 the vector $(X[0].Y[0],Z[0])^T$ denotes the initial position of the object or the manipulator's end-effector wh respect to the world frame. while the vector $(X,Y,Z)^T$ denotes the trajectories in 3-D of the object or the manipulator's end-effector. In Figures 6 and 7, the results from the application of h e autonomous adaptive techniques are presented (autonomous mode). Their main characteristic is good *tracking* performance with small oscillations. The presence of oscillations is due to the fact that the computation of the translational velocity vector T is sensitive to image noise. Image noise *can* affect the motion of the manipulator in the Z direction. In addition, a tracking error smaller than 2 mm in the Z direction cannot be detected as a pixel displacement in the image plane. This is due to the specific camera model and the specific positions of the features' projections on the image plane.

In Figures 8 and 9, we see the results of h e combined effon of the human operator with the autonomous *tracking* modules (shared control). The human *operator* commands the robot motion across the optical axis of the camera *MezP*[2] while *MezP*(0] and *MezP*[1] arc camera,

	f	s_x	s_y	c_x	c_y
Value	7.5 mm	$12.78 \frac{\mu m}{pixel}$	$9.86 \frac{\mu m}{pixel}$	255 pixels	246 pixels

Table 1: Parameters of the *camera* model.

7. Conclusions

This papa has addressed issues related with the integration of a human operator with autonomous robotic visual servoing and tracking modules. We view the telerobotic control as the integration of the human operator with an environment that consists of multiple sensors and robots. *Our* work provides a framework for such an integration and an experimental setup has been built *to* test its potential. In particular, we stated the problem of telerobotic visual savoing as a problem of combining the abilities and the experience of a human operator with the efficacy of simple autonomous visual tracking and servoing modules. We tested these ideas on our experimental testbed, the CMU DD *Arm* IL Experiments in visual servoing and visual tracking were performed while the human operator was included in the control loop through a joystick. The operator received visual information only through a video monitor and was not allowed *to* see the scene. The autonomous modules were based on the SSD *algorithm* for detection of motion and on adaptive control techniques for proper estimation of *the* unknown parameters of the environment (depth *camera* model, etc.).

The experiments lead us *to* some interesting observations. The human operator can compensate fast for errors in 2-D and *can* provide a contingency plan for tracking failures (bad feature selection, sudden target occlusion by another object etc.). *On* the other hand. the human operator reacts with delay due to delays associated with the transfer of visual information. and has a poor perception of the depth parameter. In particular, it is extremely difficult for the operator to understand and react to motion parallel to the *optical* axis of the camera Autonomous visual servoing modules allow for fast *tracking* in 3-D. even when the motion is parallel to the optical axis of the camera, adaptability *to* unknown parameters (unknown shape, *inaccurate* calibration), fast repositioning of the robot with *respect* to the target and high accuracy.

The integration of multiple cameras in the system (redundant visual information), the computational improvement of *our* algorithms. the introduction and use of "snakes" for autonomous contour scrvoing. the modeling of the delays of the human operator, the use of known CAD models for fast and accurate tracking, and the investigation of stability issues that arise with the presence of the human operator, are issues for future research.

8. Acknowledgements

This research was supported by the Defense Advanced Research Projects Agency, through ARPA Order Number DAAA-21-89C-0001. 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. We must also thank Innovision Inc. for providing us with the image processing equipment.

References

1. A.K. Bejczy, and Z.F. Szakaly, "An interactive manipulator control system", *Mini- and Microcomputers*, Vol. 5, No. 1, 1980.

2 S. Hayati and S.T. Venkataraman, "Design and implementation of robot control system with traded and shared control capability", *Proc. 0f the IEEE Int. Conf. on Robotics and Automation*, 1989, pp. 1310-1315.

3. *H.* Schneiderman and P. K. Khosla, "Implementation of traded and shared control strategies fa exploration using a tactile sensor", *Proc. of the Fourth ANS Topical Meeting on Robotics and Remote Systems*, February 25-27 1991, pp. 217-226

4. T.B. Sheridan, "Human supervisory control of robot systems", *Proc. of the IEEE Int. Conf. on Robotics and Automation*, April 7-10 1986.pp. 808-812.

5. N. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Virion and control techniques f a robotic visual tracking", *Proc. of the IEEE Int. Conf. on Roboiur and Automation*, 1991. pp. 857-864.

6. N. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Adaptive robotic visual tracking", *Proc. of the American Control Conference*, June 1991, pp. 962-967.

7. N. P. Papanikolopoulos and P. K. Khosla, "Feature based robotic visual tracking of 3-D translational motion". *Proc. of the 30th IEEE CDC. Brighton, UK*, December 1991, pp. 1877-1882.

8. P.K. Allen, "Real-time motion tracking using spatio-temporal filters", *Proc. DARPA Image Understanding Workshop*, 1989, pp. 695-701.

9. J.T. Feddema, C.S.G. Lee, and O.R. Mitchell, "Weighted selection ol image features fa resolved rate visual feedback control", *IEEE Trans Robotics and Automation*, Vol. 7, No. 1. 1991. pp. 3147.

10. V.H.L. Cheng and B. Sridhar, "Integration of active and passive sensors for obstacle avoidance", *IEEE Control Systems Magazine*, Vol. 10, No. 4, 1990, pp. 43-W

11. L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth from image sequences", *International Journal oj Computer Vision*, Vol. 3.1989. pp. 209-236.

12. N.P. Papanikolopoulos, B. Nelson and P.K. Khosla, "Monocular 3.D visual tracking of a moving target by an eye-in-hand robotic system", Tech. report .Carnegie Mellon University, The Robotics Institute, 1991.

13. G.C. Goodwin and K.S. Sin. *Adaptive filtering, prediction and control*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632. Information and Systems Science Series, Vol. 1, 1984.

14. W. Forstner and A. Pertl, "Photogrammetric standard methods and digital image matching techniques for high precision surface measurements", *ES. Gelsema and LN. Kanal, editors, Pattern Recognition in Practice II*, Elsevier Science Publishers, 1986. pp. 57-72.

15. F. Chaumette, P. Rives and B. Espiau, "Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing", *Proc. of the IEEE Int. Conf. on Robotics and Automation*, April 1991, pp. 2248-2253.

16. P.S. Maybeck, *Stochastic models, estimation, and control*. Academic Press, London. 1979.

17. H. Schneiderman, "Issues in a telerobotic system: control trading, control sharing, and safety during manual operation", Master's thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1990.

18. D.B. Stewart, D E Schmitz, and P.K. Khosla, "Implementing real-time robotic systems using CHIMERA II", *Proc. of 1990 IEEE Int. Conf. on Robotics and Automation*, Cincinnati, Ohio, May 1990, pp. 598-603.

19. *O.* Hirzinger, "The space and telerobotic concepts of DFVLR ROTEX", *Proc. of the IEEE Int. Conf. on Robotics and Automation*, March 31-April 3 1987. pp. 443-449.

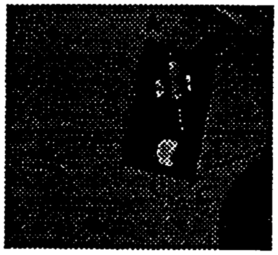

Figure 1: Initial image of the target in the servoing example around a static target.

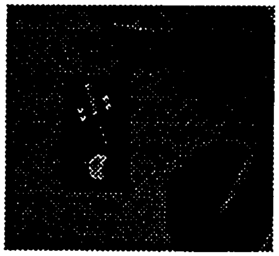

Figure 2: Final image of the target in the servoing example around a static target.

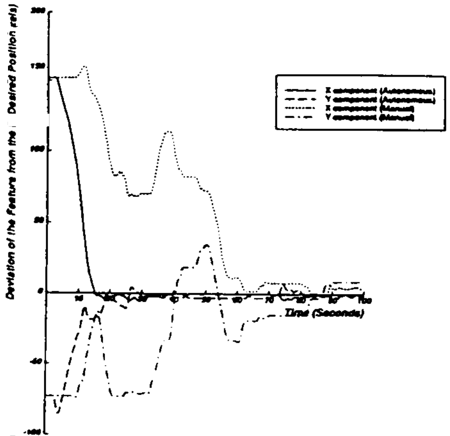

Figure 3: Servoing errors for the first feature (A). The target is static.

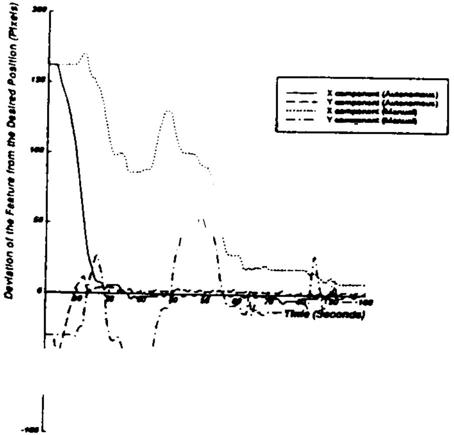

Figure 4: Servoing errors for the second feature (B). The target is static.

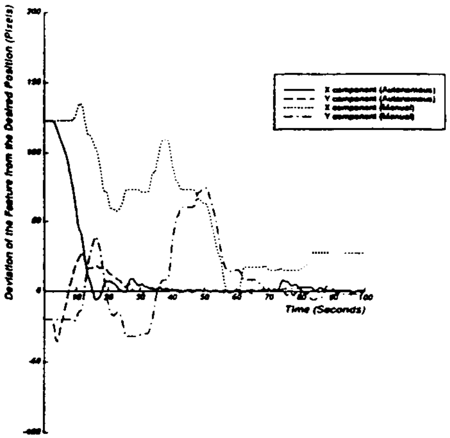

Figure 5: Servoing errors for the third feature (C). The target is static.

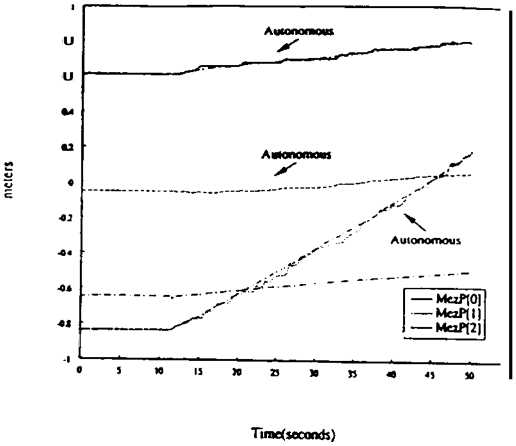

Figure 6: SISO adaptive controllers (3-D case). All the degrees of freedom are controlled by the autonomous visual trackers (autonomous control). This Figure shows the trajectories of the moving object and the manipulator's end-effector with respect to the world frame.

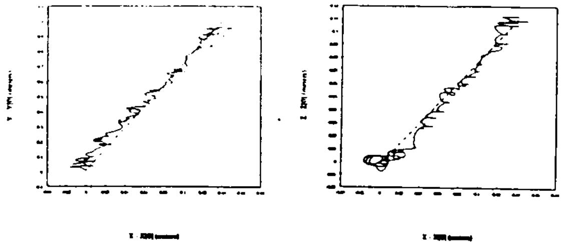

Figure 7: SISO adaptive controllers (previous example). All the degrees of freedom are controlled by the autonomous visual trackers (autonomous control). This Figure shows the relative trajectories of the object and the manipulator's end-effector in Y-X and Z-X.

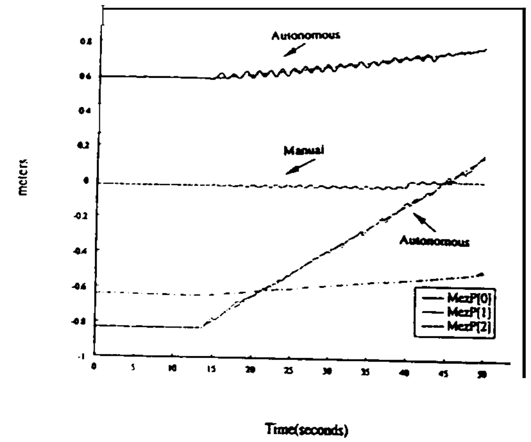

Figure 8: SISO adaptive controllers and the human operator (3-D case). *MezP*[0] and *MezP*[1] are commanded by autonomous visual servoing modules while *MezP*[2] is commanded by manual control (shared control). This Figure shows the trajectories of the moving object and the manipulator's end-effector with respect to the world frame.

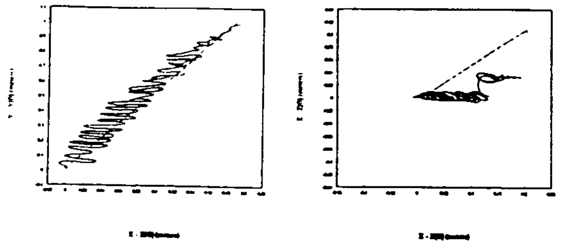

Figure 9: SISO adaptive controllers and the human operator (previous example). *MezP*[0] and *MezP*[1] are commanded by autonomous visual servoing modules while *MezP*[2] is commanded by manual control (shared control). This Figure shows the relative trajectories of the object and the manipulator's end-effector in Y-X and Z-X.

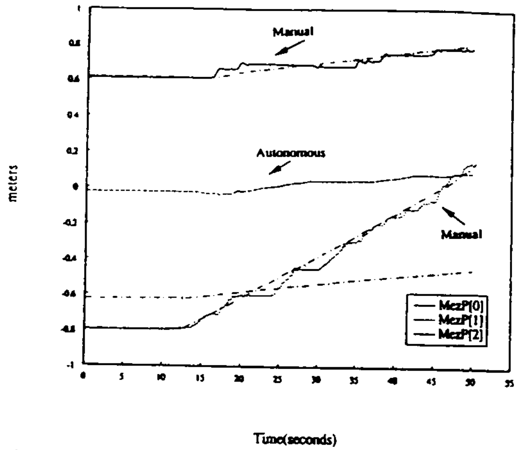

Figure 10: SISO adaptive controllers and the human operator (3-D case). *MezP*[0] and *MezP*[1] are commanded by manual control while *MezP*[2] is commanded by autonomous visual servoing modules (shared control). The human operator has little experience in using the joystick. This Figure shows the trajectories of the moving object and the manipulator's end-effector with respect to the world frame.

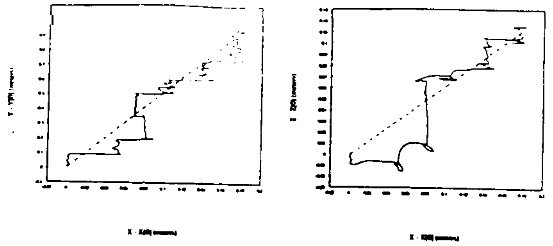

Figure 11: SISO adaptive controllers and the human operator (previous example). *MezP*[0] and *MezP*[1] are commanded by manual control while *MezP*[2] is commanded by autonomous visual servoing modules (shared control). The human operator has little experience in using the joystick. This Figure shows the relative trajectories of the object and the manipulator's end-effector in Y-X and Z-X.

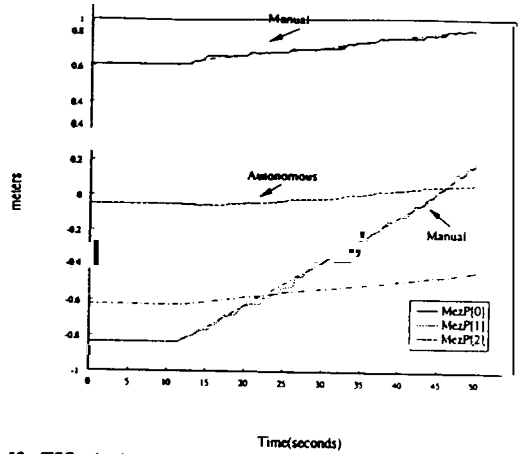

Figure 12: SISO adaptive controllers and the human operator (3-D case). *MezP*[0] and *MezP*[1] are commanded by manual control while *MezP*[2] is commanded by autonomous visual servoing modules (shared control). The human operator is experienced. This Figure shows the trajectories of the moving object and the manipulator's end-effector with respect to the world frame.

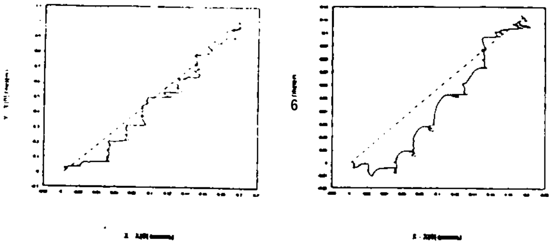

Figure 13: SISO adaptive controllers and the human operator (previous example). *MezP*[0] and *MezP*[1] are commanded by manual control while *MezP*[2] is commanded by autonomous visual servoing modules (shared control). The human operator is experienced. This Figure shows the relative trajectories of the object and the manipulator's end-effector in Y-X and Z-X.