Visual Servoing by Partitioning Degrees of Freedom

Paul Y. Oh, *Member, IEEE,* and Peter K. Allen, *Member, IEEE*

***Abstract*—There are many design factors and choices when mounting a vision system for robot control. Such factors may include the kinematic and dynamic characteristics in the robot's degrees of freedom (DOF), which determine at what velocities and fields-of-view a camera can achieve. Another factor is that additional motion components (such as pan-tilt units) are often mounted on a robot and introduce synchronization problems. When a task does not require visually servoing every robot DOF, the designer must choose which ones to servo. Questions then arise as to what roles, if any, do the remaining DOF play in the task. Without an analytical framework, the designer resorts to intuition and try-and-see implementations. This paper presents a frequency-based framework that identifies the parameters that factor into tracking. This framework gives design insight which was then used to synthesize a control law that exploits the kinematic and dynamic attributes of each DOF. The resulting multi-input multi-output control law, which we call partitioning, defines an underlying joint-coupling to servo camera motions. The net effect is that by employing both visual and kinematic feedback loops, a robot can quickly position and orient a camera in a large assembly workcell. Real-time experiments tracking people and robot hands are presented using a 5-DOF hybrid (3-DOF Cartesian gantry plus 2-DOF pan-tilt unit) robot.**

***Index Terms*—Coupled systems, gantry, hybrid robot, partitioning, robot control, tracking, visual servoing.**

I. Introduction

A ROBOT is physically characterized by the kinematic and dynamic attributes of its degrees of freedom (DOF). The number, range limits, and time responses of the DOF define the robot's performance. A vision system is often mounted on a robot with image data servoing the DOF into desired camera poses. In designing a control law to govern these robot-mounted camera motions, one must choose which robot DOF to invoke and determine if they are well suited for the designated visual control task. For example, tracking tasks require choosing robot DOF with fast response times so that the camera can be quickly centered over the moving target. Workspace monitoring tasks, however, may demand wide fields-of-view, and thus one chooses DOF that permit a larger range of motion (albeit possibly slower) for camera maneuvering.

In addition to the task, other factors confound the choice of which DOF to invoke. Oftentimes, a robot is retrofitted with additional motion components such as a configurable gripper or a motorized pan-tilt unit (PTU) that holds the camera. This results in a hybrid robot with extra layers of motion control. These components introduce additional servo update rates to the robot's underlying trajectory generator. The net result is disparate timing loops and this requires proper gain tuning to coordinate all robot DOF and avoid a misconfigured camera pose. A robot with many DOF (possibly redundant) can also present multiple choices for visual-servoing implementation. For example, in centering the camera's image plane over a moving target, only two DOF, like pan and tilt, may be required. The questions that arise are as follows: which two to choose and what roles, if any, do "left-over" DOF play.

Pose regulation is ubiquitously discussed in the visual servoing literature and implemented on 6-DOF robots to track geometrically simple targets like blocks, cylinders, and spheres [8], [9], [13], [17], [3]. It is a specific case where no design choices, aside from image feature selection, resulting image Jacobian, and gain tuning, are really necessary since every DOF is visually servoed. Its design synthesis and implementation are well known [11]. Pose regulation forces the end-effector mounted camera to mimic target motions. One only needs to define the image and manipulator Jacobians to achieve this task in joint space. The caveat, however, is that all the robot's DOF must have sufficient bandwidth to keep the target's fiducials in view. The DOF with the slowest time response will limit performance. Corke discusses how lag errors result from ignoring robot dynamics in the control law and prescribes compensator designs [5]. Kalman filtering [1], [18] is an alternative solution, but requires *a priori* knowledge of target trajectory which is not always available.

In the big picture, when visual servoing of all the robot's DOF is not required, like camera centering over a moving target, a more general problem arises. The combination of task, robot DOF attributes, motion component update rates, and multiple DOF choices compound the decision of which DOF to invoke through visual servoing. Tracking more geometrically complex targets, without *a priori* knowledge of trajectories, in unstructured environments add to this problem. One must then resort to trial-and-error implementations and perhaps *ad hoc* gain tuning of the control laws.

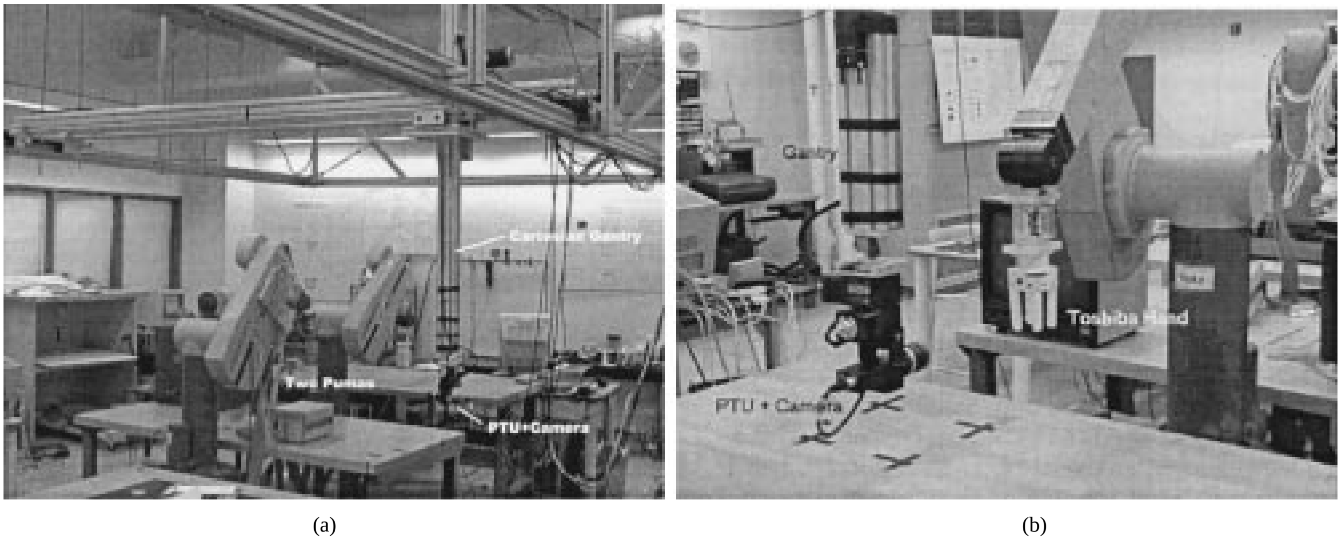
This paper presents a design methodology under an analytical framework. Frequency-based domain techniques are used to decide which DOF should be visually servoed and root-locus is used for gain tuning. This analysis is then used to develop a novel multi-input multi-output control scheme we call *partitioning* [14]. The scheme defines an underlying joint-coupling among the various DOF using both image and kinematic joint data. Joint-coupling and kinematic data are introduced to actuate DOF that are not visually servoed but still serve in the vision

Manuscript received January 7, 2000; revised August 28, 2000 and December 28, 2000. This paper was recommended for publication by Associate Editor Y. Xu and Editor V. Lumelsky upon evaluation of the reviewers' comments. This work was supported in part by an ONR/DARPA MURI Award ONR N00014-95-1-0601 and the National Science Foundation under Grant CDA-96-25374. This paper was presented in part at the International Conference on Robotics and Automation, San Francisco, CA, 2000.

P. Y. Oh is with the Mechanical Engineering and Mechanics Department at Drexel University.

P. K. Allen is with Columbia University, New York, NY 10027 USA.

Publisher Item Identifier S 1042-296X(01)03158-5.



(a)

(b)

Fig. 1. (a) The workcell contains two Puma robots. The gantry and PTU can position and orient the camera anywhere in the workcell. (b) The desired vision system would automatically track objects like grippers, tools, and workpieces the Pumas manipulate.

task. The paper presents its implementation on an eye-in-hand 5-DOF hybrid robot. This robot is a 2-DOF PTU mounted on a 3-DOF Cartesian gantry that can track geometrically complex targets like tools, workpieces and grippers, which have nondeterministic motion trajectories, over a large 3.6 × 6.4 × 1 m^3 workspace. The tracking results suggest that large bandwidth DOF (i.e., fast time response) can be visually servoed to physically act as compensators for slower DOF. Section II gives an overview of our vision interests and our eye-in-hand robot. Highlights from pose regulated tracking experiments illustrate performance limitations and motivates our need for an alternative approach. Section III analyzes the results of visually servoing individual DOF from the frequency domain. Under this framework, the DOF to be visually servoed are chosen. Section IV then presents our joint-coupled approach called partitioning. The remaining DOF assist in target tracking and are kinematically servoed. Three example applications are shown in Section V tracking people and a robot hand. Section VI gives conclusions and potential extensions.

II. Vision Interests

Our vision interests are in monitoring a large $3.6 \times 6.4 \times 1$ m^3 workcell containing two Puma 560 robots. We wanted a robot-vision system to track objects the Pumas manipulate such as grippers, tools, and workpieces. We custom-built a ceiling mounted 3-DOF Cartesian gantry and attached a 2-DOF PTU and camera to its end-effector. The resulting 5-DOF hybrid robot can position and orient the camera anywhere in the workcell (see Fig. 1).

The gantry has three steppers driven by a commercially available PC motion board and external chopper unit that reports and updates gantry positions and velocities on-the-fly. Motors 1 and 2 have 7.50×10^{-6} m/step resolutions with maximum velocities of 0.7 and 0.5 m/s, respectively. Motor 3 has a 3.75×10^{-6} m/step resolution and a 0.5-m/s maximum velocity (see Fig. 2).

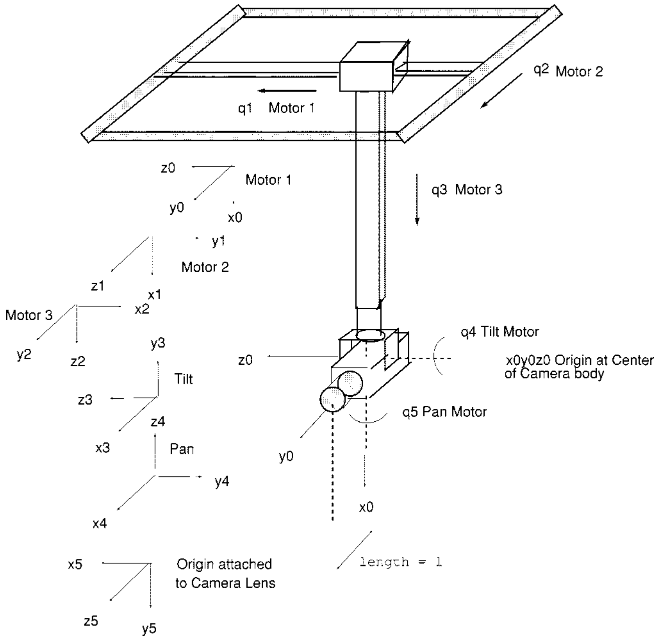


Fig. 2. Denavit–Hartenberg reference frames.

The off-the-shelf PTU is a modularized unit with a 9600-baud serial port and consists of two steppers, driver chips and an embedded microprocessor (Motorola 68HC11). Pan (±150° range, 128°/s maximum velocity) and tilt (−47° to +31° range, 113°/s maximum velocity) angles and velocities can be monitored and changed on-the-fly. The net effect is a 5-DOF hybrid eye-in-hand robot with two separate motion controllers and with different servo update rates.

Images are acquired with an off-the-shelf charge-coupled device (CCD) camera and digitized by a Sparc 20 installed framegrabber and X-Vision [10] was used for image processing. The Sparc is the host computer which handles image data (acquisition and processing) and issues ASCII-encoded gantry and PTU motion commands via serial communications.

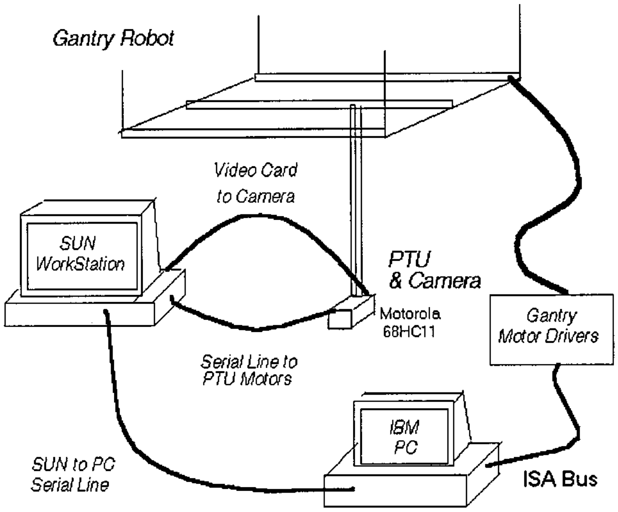


Fig. 3. Robot and hardware communications.

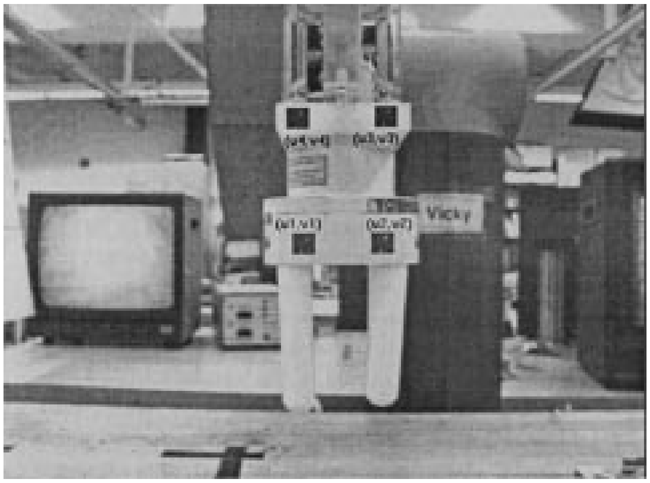


Fig. 4. The camera-to-hand pose is to be regulated. For co-planar fiducial marks (black squares) were taped onto the hand and tracked with four SSDs returning $(ui,\ ui)$, $i = 1 \cdots 4$ pixel locations.

The Sparc-to-PC and Sparc-to-PTU communicates at 38 400 and 9600 baud, respectively. The gantry's PC motion board updates stepper pulses every 30 ms and low-level interrupt programming was used to minimize serial latency to 6 ms. The PTU however has a large latency (0.12 s) due to Unix context switching. The PTU uses proprietary firmware and has no provisions to handle low-level handshaking. To maintain fast image handling, Unix's multitasking capabilities were exploited by generating two processes connected by an interprocess pipe. The first, a vision-handling process grabs and processes image data continuously. Its loop rate depends on the amount of image processing computation required. It oversamples by repeating computations on image data stored in video memory until a new frame of image data is acquired. This oversampling results in loop rates faster than 30 Hz. The second, a main process, requests pixel data, computes, and then issues PTU and gantry commands. The net effect is an integrated robot-vision system that could acquire real-time image data and move the camera in five DOF (see Fig. 3).

In using this robot for visually-servoed tracking, we first implemented a pose regulation scheme, Fig. 4, to perform the classic block/plane following experiment [8], [9], [13], [17], [3]. Given known side-lengths, the plane's pose is extracted from the image positions of four co-planar fiducials painted on the target. In our implementation, we used four 40×40 pixel sum-of-squared difference (SSD) region-based trackers to report the pixel positions of the four fiducials. The tracking task results in pose regulation whereby robot servoing maintains a predefined camera-to-target pose.

In our own experiments with this approach [14], we encountered a number of problems. First, the gantry motors have a slow rise time due to the large inertial loads of its links. The target translating at speeds greater than 2 cm/s would leave the camera's field-of-view before the gantry could accelerate up to speed. Tracking performance was poor for even slow target velocities; pose regulation forces camera motions to mimic target motions, thus target rotations, like yaw about its center of mass, force the gantry to sweep large rotations quickly. In these instances, the target's fiducials often became occluded before the gantry could circle into position and tracking would fail.

Second, abrupt gantry accelerations led to end-point camera vibrations which corrupted image data. Kalman-based filters [18], [1] were implemented to improve image robustness. However, Kalman filtering requires *a priori* knowledge of the target's trajectory, which in some of our workcell operations are unknown beforehand. To reduce the frequency of vibrations we physically added mass to the links, but these vibrations were not completely eliminated.

Third, targets like our grippers change configuration while in flight. As the fingers begin closing into a grasp they can occlude fiducials. Also, some of our workpieces are geometrically complex and would require additional image features for estimating its pose. This results in a larger image Jacobian matrix which is more computationally expensive to invert and more sensitive to image noise.

The net effect of these problems is that performance for tracking moving targets with this robot is related to which DOF are invoked in the tracking task. For example, we can track a target at high velocities using the PTU alone (fixed gantry position); however, the range of the PTU angles are limited, and arbitrary pose configurations of the camera-to-target cannot be satisfied. If we allow all five DOF to be used, we then limit our tracking velocities.

Restating our objective, we wish to track geometrically complex targets like workpieces, tools, and grippers traveling in a large unstructured workspace with *a priori* unknown trajectories. Our tracking needs only require camera image plane centering over the moving target within certain resolution limits. For this, a pose regulation solution over-engineers our needs; only two DOF are needed for camera centering. Although one could take an *ad hoc* try-and-see approach by visually servoing any two DOF, the aforementioned problems underscore the need for a more methodic approach and design synthesis. The next section approaches DOF selection by analyzing the visual servoing responses of individual DOF in the frequency domain.

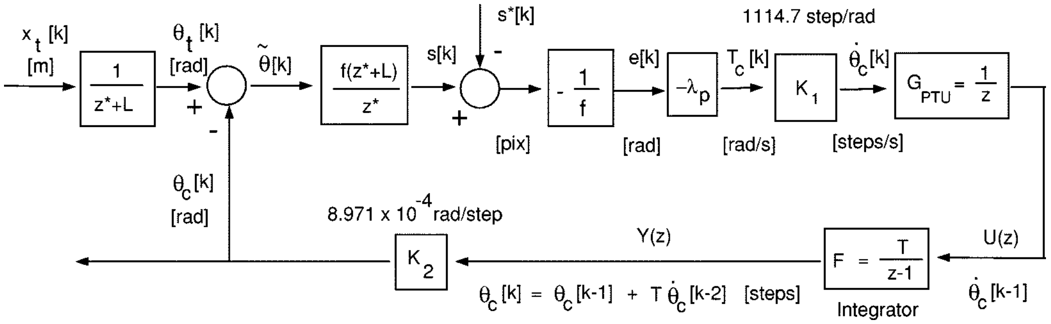


Fig. 5. Block diagram for pan-only design.

This gives quantitative measures of each DOF's tracking performance. Bandwidth, phase characteristics, and root locus gain tuning are then used to choose a suitable DOF pair.

III. Analytical Framework

Our tracking task is the visually-servoed centering of the camera's image plane over a moving target. The analytical framework used to measure task performance is frequency based. The target is oscillated and the frequency response (bandwidth and phase) of the visually-servoed DOF is measured. This task only requires measuring the center of gravity of the target's image centroid. The resulting horizontal u and vertical v pixel position pair is compared to the pixel location of the image plane center and camera motions are invoked to minimize the difference. Typically, the invoked motions are velocity commands, and tracking in this manner is cited in the visual-servoing literature as *piloting* [4] or *steering* [5].

These two references also report frequency response results of visually-servoed DOF. The former used a target placed on a turntable (for pan-tilt response) and the later used a pair of alternating LEDs. Beyond reporting frequency response results and gain tuning, we analyze frequency responses for design insight. Bandwidth and phase quantitatively identify the advantages and disadvantages of visually servoing a DOF. Our end goal is to then design a controller that combines these advantages; the result would exploit each DOF's kinematic and dynamic attributes to increase tracking performance.

Our analytical framework, to meet this end goal, begins by identifying the design parameters that affect tracking performance. Such parameters include communication latencies and gains. We thus mathematically model, in Section III-A, two visually servoed, input–output, tracking systems. The first invokes pan only and the second invokes a single gantry DOF (q_1 in Fig. 2). Each system alone can center the camera's image plane over a moving target; the target's line of motion is constrained to be horizontal (perpendicular to the initial camera optical axis) and parallel to the initial camera image plane. The performance limits of camera rotation versus camera translation become clear from the mathematical modeling. The pan and tilt motors are identical and all three gantry steppers are the same. Thus, the performance limits of the pan-only system can be extended to the tilt and those of one gantry stepper apply equally to the other translational DOF. Hence, only the visual-servoing models for pan and for one gantry DOF are presented.

To validate the models, Section III-B shows experimentally constructed Bode plots that compare closely to mathematically generated (Matlab) ones. Bandwidth, phase, and rise times of the two systems are summarized. Comparing the two systems reveals design insights that suggest a controller that defines a joint coupling among DOF can improve tracking performance.

A. Tracking Models

One can rotate or translate a camera to track a target constrained to horizontal motions. The two differ in light of two factors. The first is the relative camera and target motions. If the camera can only rotate, the camera-to-target distance z^* changes as the target translates horizontally. However, if the camera can translate, z^* remains constant. Measuring the center of gravity of the target's image centroid does not yield depth information. As a result, pan-only tracking relies on small angle approximations, using the known initial z^* and perspective, to map pixel locations into camera-to-target bearing angles. Camera translation however does not need such approximations. The second factor is the digital feedback law dependence on update rate T, which is limited by communication latency. The PTU and gantry have different latencies. These factors show up as parameters in the following mathematical models.

1) Pan-Only Tracking Model: The transfer function model for visually servoing the pan DOF only can be derived using the discrete-time block diagram (Fig. 5). It shows both sample instants k and units for clarity. Here, target position x_t is the input and results in a camera pan angle θ_c output. The block diagram includes elements with their linearized equations and are described as follows (see Fig. 6).

The target translates a distance x_t along the X-axis. x_t can be alternatively expressed as a bearing angle θ_t. If the target is not centered in the camera's image plane, then a nonzero angle difference $\tilde{\theta}$ exists between θ_t and camera angle θ_c. Assuming small angles, $\tilde{\theta}$ can be approximated by

$$\tilde{\theta} = \theta_t - \theta_c \approx \frac{x_t}{z^* + L} - \frac{x'_c}{z^* + L} \tag{1}$$

z^* and L are the initial lens-to-target distance and lens' radius of rotation, respectively. ϕ is the angle the target makes with respect to the lens center. A pin-hole camera model yields

$$\phi \approx \frac{x_t - x'_c}{z^*} \equiv \frac{u}{f} \tag{2}$$

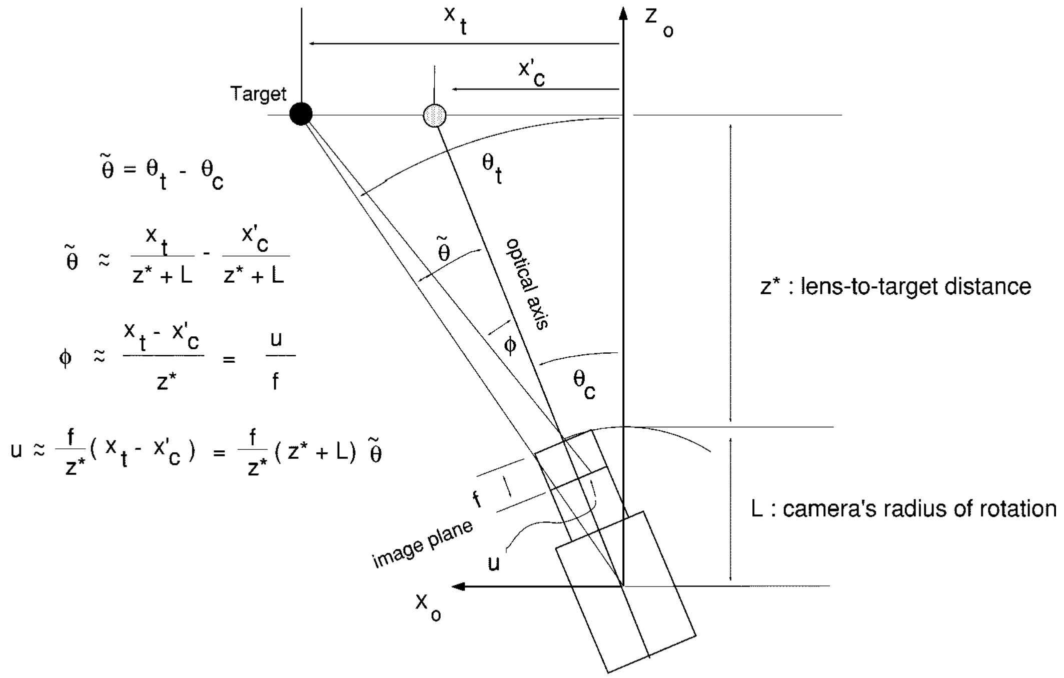

x_t
z_o
x'_c
Target
θ̃ = θ_t - θ_c
θ̃ ≈ x_t/(z* + L) - x'_c/(z* + L)
φ ≈ (x_t - x'_c)/z* = u/f
u ≈ f/z* (x_t - x'_c) = f/z* (z* + L) θ̃
θ_t
θ̃
optical axis
φ
θ_c
z* : lens-to-target distance
L : camera's radius of rotation
f
image plane
u
x_o

Fig. 6. Relevant notation used for linearization.

where u is the horizontal pixel coordinate of the target in the camera's image plane and f is the lens focal length in pixels. The two previous equations combine to form

$$u = \frac{f}{z^*}(x_t - x_c') = \frac{f}{z^*}(z^* + L)\tilde{\theta} \tag{3}$$

and re-expressed into the following relationships:

$$\frac{x_t}{\theta_t} = z^* + L \quad \text{and} \quad \frac{\tilde{\theta}}{u} = \frac{z^*}{f(z^* + L)}. \tag{4}$$

The $f(z^* + L)/z^*$ term is a camera-to-target distance dependent gain due to perspective. The horizontal and vertical pixel units u and v define the 2-D target position $\vec{s} = (u,\, v)$ in the camera's image plane.

Fig. 5 shows these two relationships as block elements before and after the $\tilde{\theta}$ adder and yields an image position $s[k]$ in pixels. $s^* = s^*(u^*,\, v^*)$ is a set point image location. Before tracking begins, the camera pose is initialized so that its image plane is centered over the target. Without loss of generality, let $s^* = (0,\, 0)$ define the image plane center.

As the target moves, the current image position $s[k]$ is compared to the set point s^*. The difference is a measure of the target position relative to the image center. Since the target only translates horizontally, one has

$$\dot{s} = \dot{u} = -\frac{1}{f}\,\omega_y \tag{5}$$

where ω_y is the target's rotational velocity [rad/s] about the lens. Assuming differential changes, we define the error [rad] to be

$$e[k] = -\frac{1}{f}\,(s[k]). \tag{6}$$

The camera is then panned at a velocity $\dot{\theta}_c[k]$ by the PTU stepper motor. This velocity is proportional to the error (steering problem) with λ_p and K_1 as the gain and unit conversion constants, respectively. Within a stepper's start/stop (pull-in) velocity range, it accelerates instantaneously. It is thus modeled, G_{PTU}, as a unit delay. Backward-rectangular integration is used with sample time T

$$\theta_c[k] = \theta_c[k-1] + T\dot{\theta}_c[k-2] \tag{7}$$

and yields the camera angle $\theta_c[k]$ in steps or in radians[1] using the unit conversion constant K_2. Centered target tracking is thus achieved if $e[k] = 0$ and the resulting closed-loop linearized transfer function is

$$G_\theta(z) = \frac{\theta_c}{x_t} = \frac{T\lambda_p}{z^* z(z-1) + T\lambda_p(z^* + L)}. \tag{8}$$

The serial latency mentioned in Section II was measured experimentally with an oscilloscope. This latency constitutes the bulk of the program loop time and pan speeds can be updated every $T = 0.12$ s. This latency does not affect the speed of image acquisition and processing. Image data handling is accomplished in real time under a separate forked process.

2) Gantry-Only Tracking Model: Unlike pan, the gantry can transport the camera anywhere in the workcell. As mentioned in Section II, the gantry's DOF are marked by large inertial loads. One can thus expect more lag when the camera is visually servoed through translation than by pan action. This will be revealed later in the frequency responses in Section III-B. Fig. 7 is the block diagram and (9) is its closed-loop transfer function. Here, the camera image and target motion planes remain parallel. The model is thus much simplier than the pan-only system,

[1] Steppers are pseudodiscrete devices. They move one fixed displacement increment in response to each pulse input. There is no digital-to-analog converter and $\theta_c(t) = K_2\theta_c[kT]$.

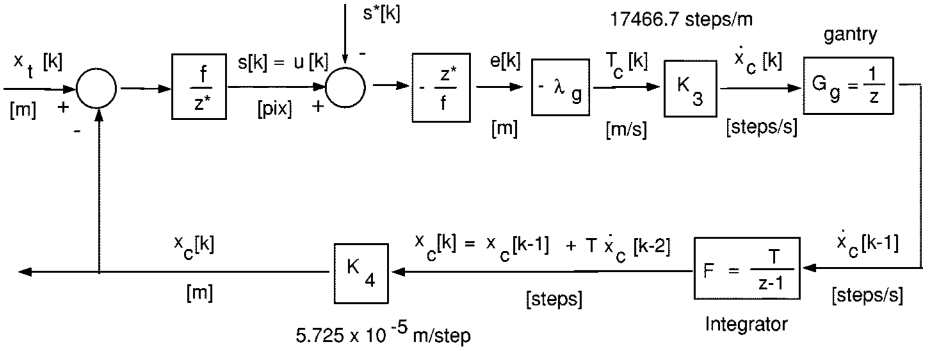


Fig. 7. Gantry-only block diagram.

since no bearing angle relationship is necessary. $u = fx_t/z^*$ is used to model perspective.

$$G_g(z) = \frac{x_c}{x_t} = \frac{T\lambda_g}{z(z-1)+T\lambda_g}. \tag{9}$$

The gantry is controlled by a PC through a commercial ISA-bus logic sequencer and chopper drive that deliver timed pulses to the gantry's stepper. This hardware allows pulse frequencies (hence, gantry velocity) to be updated *on-the-fly* as fast as every 30 ms. The host, a Sparc 20, issues velocity commands (in ASCII) to the PC via a 38 400 baud serial line. With low-level PC programming a serial latency of 6 ms was achieved. To be within a safe margin, gantry velocities were updated every $T = 40$ ms.

To summarize, visually-servoed tracking by camera pan and by camera translation have closed-loop transfer functions (8) and (9), respectively. Both systems use steering via a proportional gain to actuate camera velocities that minimize pixel errors. Pan-only tracking performance is characterized by a camera-to-target bearing angle approximation and a large update rate. The effects these characteristics have on tracking will be revealed in the ensuing Bode plots.

B. Frequency Response

To validate the closed-loop transfer function models, Bode plots were experimentally constructed. These plots were compared to those generated mathematically using Matlab and correspond closely. Bode diagrams plot sinusoidal input responses over a frequency range revealing bandwidth and phase metrics. They can be obtained experimentally in a number of ways. Reference [4] tracked a target placed on a turntable (for pan-tilt response) and [5] used a pair of alternating LEDs. Alternatively, one can use a spectrum analyzer or oscilloscope if the target input and servo encoder output can be directly measured.

In our implementation, a block target was mounted in a Puma gripper. The Puma oscillates the target along the horizontal line of motion at a user prescribed frequency (see Fig. 8). The target was visually tracked and both the target position x_t [m] and resulting joint positions (radians for the pan-only and meters for gantry-only system) were recorded. Twenty different trials were performed over a 0.02–1.0-Hz range for the pan-only system, and 12 for the gantry-only system with $z^* = 0.25$ m and $f = 633$ pixels. These experimentally constructed Bode plots (circles) correspond closely the Matlab computer generated Bode plots (solid), thus giving confidence in the mathematical modeling. The plots are shown in Fig. 9.

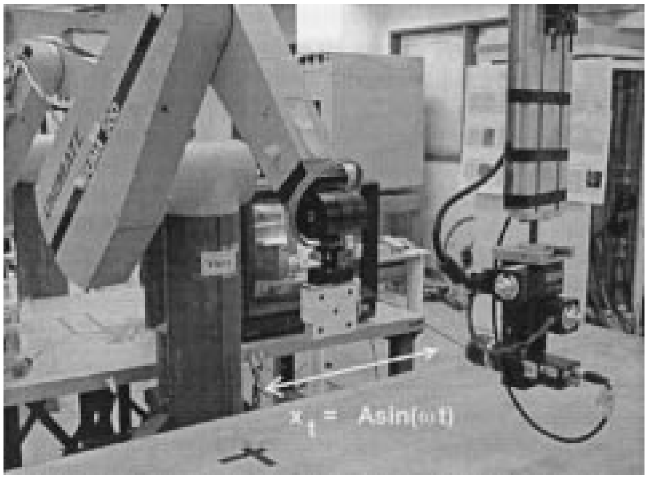


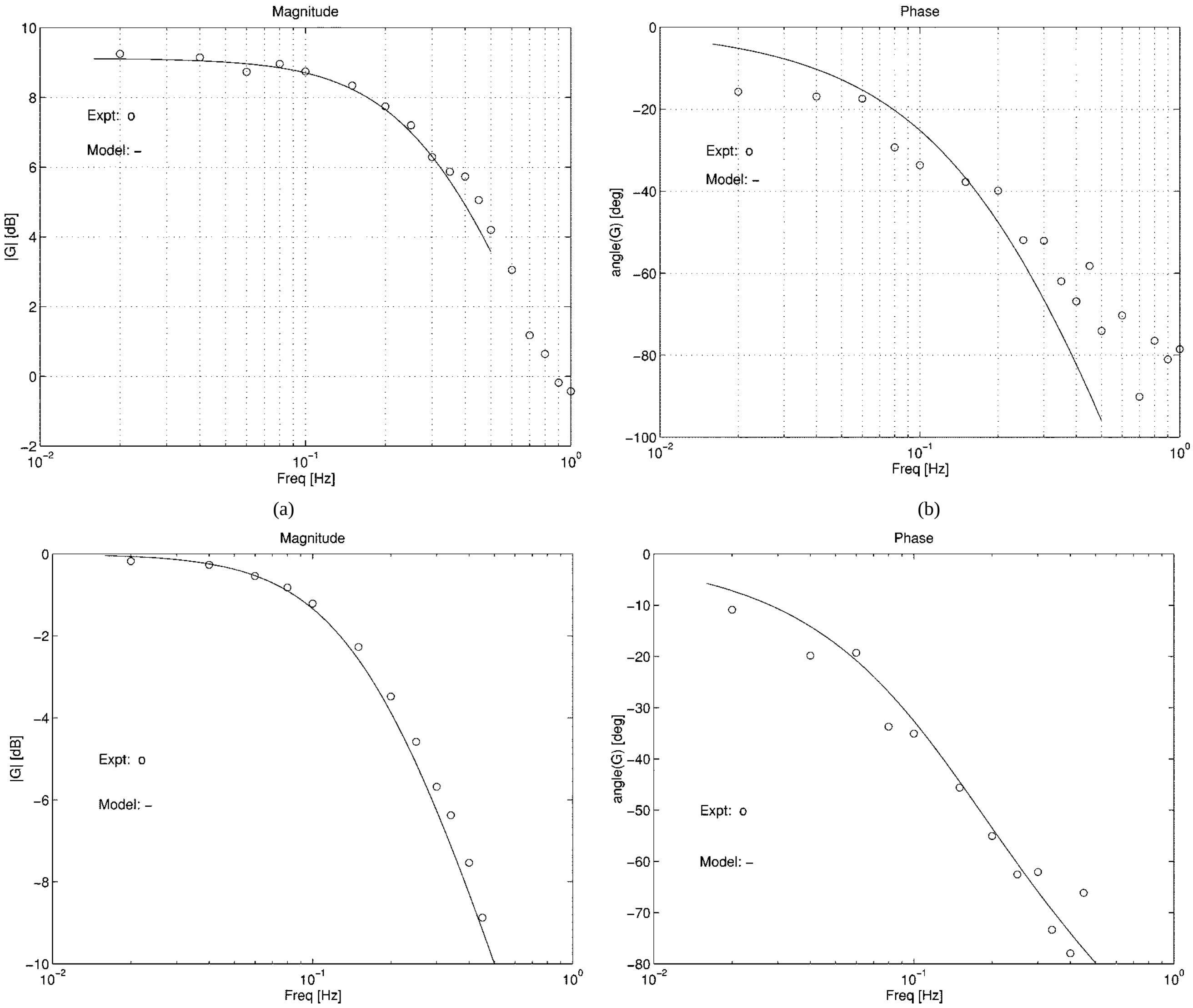
Fig. 8. The Puma translates the block target horizontally back and forth at a user prescribed frequency and amplitude. Bode plots can then be constructed by recording the joint response.

The pan-only Bode plot [Fig. 9(a) and (b)] reveals a bandwidth of ≈0.3 Hz (1.9 rad/s) with a phase angle of ≈−70° (−1.2 rad) and 1.13-s rise time. The closed-loop transfer function model (8) with $\lambda_p = 1.0$ yields a dominant pole at $z = 0.7822$ on the real z-axis. Since $z = e^{sT}$ then for $T = 0.12$ s, the natural frequency is $\omega_n = 2.05$ rad/s. The gantry-only Bode plots [Fig. 9(c) and (d)] indicate a bandwidth of ≈ 0.1668 Hz (1.048 rad/s) and phase angle of $\approx -46.76°$ and 2.08 s rise time. Its closed-loop transfer function model (9) with $\lambda_g = 1.0$ and $T = 40$ ms yields a dominant closed-loop pole at $z = 0.956$ or a natural frequency of $\omega_n = 1.12$ rad/s.

The gains, $\lambda_p = \lambda_g = 1.0$, were selected in light of known stepper motor start/stop frequency ranges and at these values there is no stall. Root locus plots (not shown), derived from the transfer function models, can be used to tune performance (rise time and overshoot behaviors).

1) Design Insights: Tracking necessitates keeping the target in the camera's field-of-view and large bandwidth DOF should be employed to afford fast camera servoing. As such, the frequency responses reveal that PTU DOF are more suited for visual servoing than gantry DOF.

To reiterate our end goal, we want to design a controller that combines the advantages that each DOF offers by exploiting



(a) (b)

(c) (d)

Fig. 9. Bode magnitude and phase plots for (a), (b) pan-only and (c), (d) gantry-only tracking. Both experimental (dots) and modeled (solid) plots are shown.

its kinematic and dynamic attributes. The modeling and the dynamic attributes of visually-servoing camera rotations and camera translations reveal design insights. First, pan-only tracking performance depends on small camera-to-target bearing angles, but its bandwidth affords fast camera motions. Second, although gantry-only tracking is sluggish as indicated by its smaller bandwidth, there is very little attenuation.

These insights suggest that better tracking, through camera rotations, can be achieved by cutting the bearing angle between the camera and target. This requires translating the camera. One way to achieve both camera rotation and translation is to define an underlying joint-coupling between the rotational and translational DOF in a control law. Its synthesis would exploit the PTU's large bandwidth by visual servoing rotational DOF (pan and tilt). This keeps the camera's image plane centered over the target. The pan and tilt angles are then used to kinematically servo the gantry to exploit its ability to translate and cut down the camera-target bearing angle. The net effect is a multi-input multi-output control architecture, we call *partitioning* [14], [15], that improves tracking performance.

IV. Partitioning

A paradigm one notices in the visual-servoing literature are control laws that exclusively use image data to command camera motions. Kinematic data, like joint encoder positions are sometimes added in feedback or feedforward loops in dynamic image-based look and move structures [11] to achieve faster response. This hints that joint and image data combinations can afford novel controllers for visual servoing. For example, in a unique approach [2] modeled visual compliance after well-understood force compliance techniques.

One control strategy, when multiple sensors (like camera and joint encoders) provide possible command inputs, is to define an underlying joint-coupling in the servoing law. Observing the behaviors people display when visually tracking also suggests that we use an underlying joint-coupling. One behavior is that the eyes and neck typically pan in the same direction when tracking. Another is that the eyes also lead (i.e., start panning before) the neck; as the eyes reach their kinematic joint limits, neck pan commences. A possible explanation for such behaviors is we

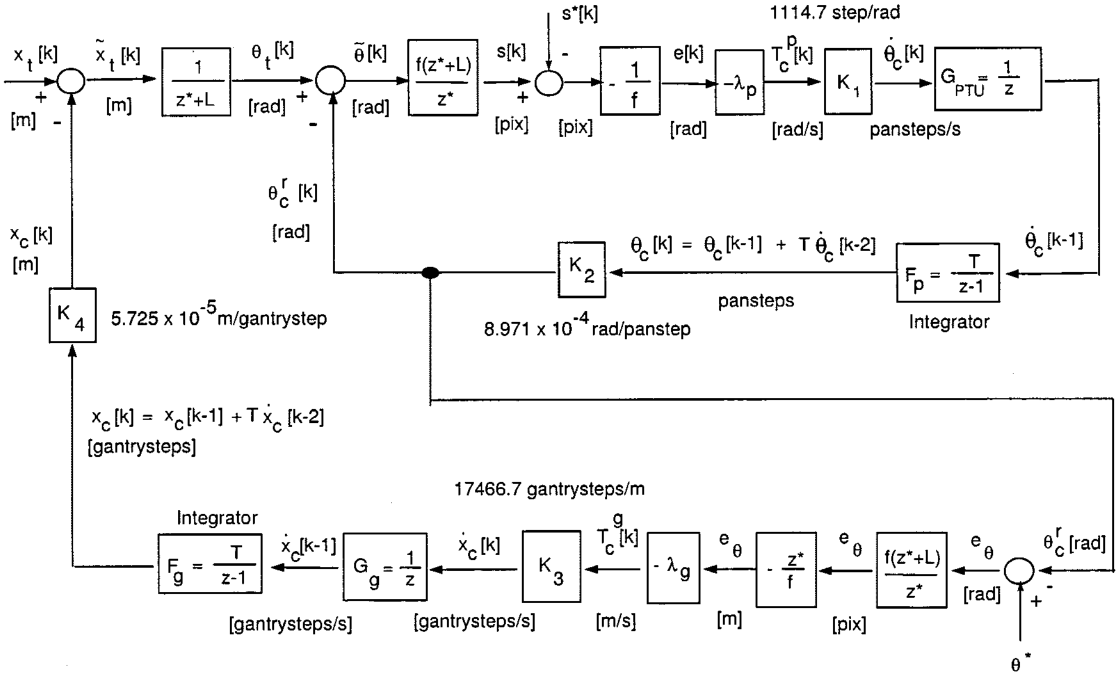


Fig. 10. Coupled pan-gantry block diagram. Notice the similarity with Figs. 5 and 7.

use both image and kinematic data when coordinating our DOF. Visually servoing DOF with fast response times (like the eyes) physically serve as lead compensators (add phase) for joints that are slower to react. *Skeet shooting* is a good physical example of lead compensation. Often a marksman will add phase by aiming ahead of the target to compensate for relative motion dynamics and reaction time (latencies).

The explanation for the observed human tracking behaviors is conjecture but joint-coupling can be implemented in the control law and is illustrated in Section IV-A. Step response experiments tracking a robot hand are also featured. Bode plots and peak-to-peak pixel error measurements (that give a clearer representation of phase lags) are given in Section IV-B.

A. Coupled Pan-Gantry Tracking Model

A 2-DOF joint-coupled system is realized as a block diagram in Fig. 10 and is structurally similar to a combination of Figs. 5 and 7. The net camera motion we wish to achieve is a coupled one; pan always centers the camera over the target and gantry translation is achieved through a joint-coupling, described as follows.

The block diagram has two distinct feedback loops: the first embeds a linearized pan-only *steering* design. Here, camera pan velocity $\dot{\theta}_c$ is visually servoed using the pixel difference $s[k] - s^*$ and proportional gain λ_p. The second embeds a gantry-only *steering* element that uses camera pan angle θ_c^r (superscript r for radians). The pan angle difference $\theta_c^r - \theta^*$ and proportional gain λ_g servo the camera's translational velocity $\dot{x}_c$. θ^* is a defined set-point angle and is the desired camera orientation. A human analogy of θ^* would be the desired eye orientation, which generally points forward when relaxed. Without loss of generality, θ^* is set to zero. The closed-loop input–output relationships can be deduced using standard block-reduction techniques. A target translational input x_t [m] yields the following:

$$^cG_\theta(z) = \frac{\theta_c \text{ [rad]}}{x_t \text{ [m]}} = \frac{\lambda_p T z^2 - \lambda_p T z}{\text{CE}} \tag{10}$$

$$^cG_g(z) = \frac{x_c \text{ [m]}}{x_t \text{ [m]}} = \frac{(z^* + L)T^2 \lambda_g \lambda_p}{\text{CE}} \tag{11}$$

with

$$\begin{aligned} \text{CE} = {} & z^* z^4 - 2z^* z^3 + (z^* + (z^* + L)\lambda_p T)z^2 \\ & - (z^* + L)\lambda_p T z + (z^* + L)T^2 \lambda_p \lambda_g. \end{aligned}$$

The superscript c in $^cG_\theta$ and ^cG_g is used to denote "coupled" and the subscripts θ and g describe pan and gantry, respectively. As in Section III-A, z^* is the lens-to-target distance, L is the camera's radius of rotation, and T is the sampling time.

1) Robot Hand Tracking—Step Response Experiment: Experiments tracking a target (robot hand) were implemented on our hybrid robot. Fig. 11 shows three sequential image stills captured while videotaping experiments. A single 40×40 SSD was used to acquire the center of gravity (u, v) pixel coordinates of the robot hand's image centroid (left-most). The coupled pan-gantry step response was achieved by first translating the robot hand by 0.1 m (middle) then letting the pan and gantry DOF servo the camera (right-most). The gridded background highlights the resulting pan and gantry motions.

The pan angles and the gantry positions were recorded during the above experiment (dashed lines) and are plotted in Fig. 12. The results correspond closely to computer simulations (solid lines) using the transfer functions (10) and (11). Gains $\lambda_p = 1.0$ and $\lambda_g = 0.5$ were selected using root locus plots (Fig. 13) derived from these transfer functions and yield closed-loop pole positions that correspond to a damping ratio of $\zeta = 0.95$. In-

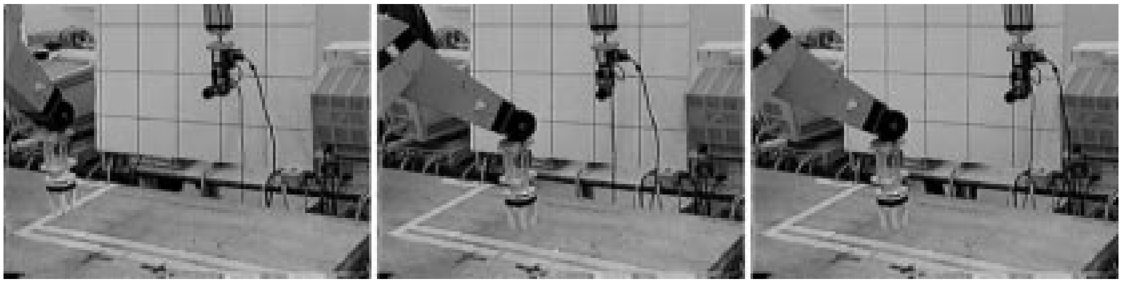


Fig. 11. Three image stills taken from a videotape of the coupled pan-gantry system step response.

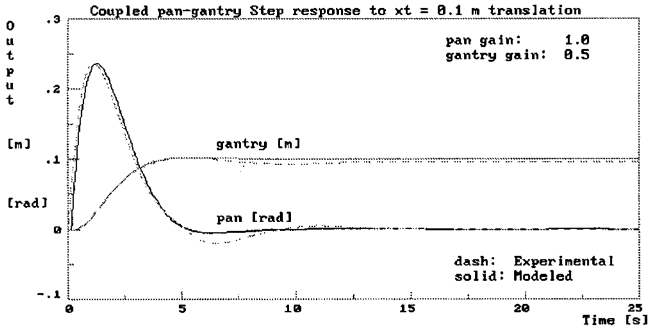


Fig. 12. Coupled pan-gantry step response. Experimental (dash) and simulations (solid) shown.

cluding the ζ lines in the root locus plots reveal necessary gains to yield faster response while ensuring stability. For example, gains can be tuned for a $\zeta = 0.707$ to yield a good compromise in both stability and performance.

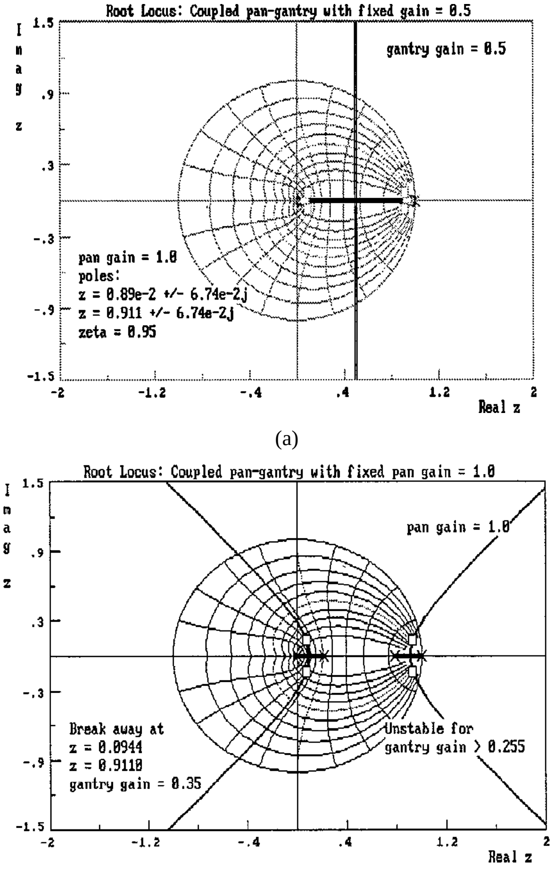
The x_t (target position) step input causes a rapid camera pan acceleration and simultaneously invokes gantry translation due to the $\theta^* - \theta_c^r$ (with $\theta^* \doteq 0$) coupling. This gantry action consequently brings the camera closer to the target and thereby cuts the camera-to-target bearing angle, thus θ_c^r decreases. The net effect is that the final camera's image plane is centered over and parallel to the target.

B. Frequency Response and Peak-to-Peak Pixel Errors

The frequency domain offers a clearer explanation of joint-coupling effects. As mentioned previously, the visually-servoed pan DOF physically acts as a lead compensator for the kinematically servoed gantry DOF. Compensators can be mathematically designed using feedforward or feedback techniques [6] to place or cancel system poles and zeros and improve lag. Beyond these mathematical abstractions, however, compensation can also be physically achieved with joint-coupled partitioning as seen in the following sinusoidal input response.

In one experiment, Fig. 14, the robot hand oscillated horizontally at 0.1 Hz (period $T = 10$ s) and its position $x_t(t)$ over time t is $-0.1 \sin(2\pi t/T)$ [m] (solid thick line). The pan and gantry joint responses were recorded (dashed lines). Gantry *lag* is quite obvious with its peak following the input peak. The pan *leads* the gantry as evidenced by the pan peak coming before the gantry peak. Simulations using (10) and (11) are also shown (solid lines) and correspond closely to experimental results.

1) Partitioned Joint-Coupling Bode Plots: The pan DOF acts like a *lead compensator*. This can be seen from the



(a)

(b)

Fig. 13. (a) Fixed $\lambda_g = 0.5$ and varying λ_p root locus with $z^* = 0.25$ m, $L = 0.1$ m and $T = 0.12$ s yields $G_{\lambda_p}^{RL} = (0.042z^2 - 0.042z + 0.002\,52)/(0.25z^4 - 0.50z^3 + 0.25z^2)$. (b) Fixed $\lambda_p = 1.0$ and varying λ_g root locus yields $G_{\lambda_g}^{RL} = 0.005\,04/(0.25z^4 - 0.5z^3 + 0.292z^2 - 0.042z)$. Closed-loop pole positions (boxes) lie on damping ratio line $\zeta = 0.95$.

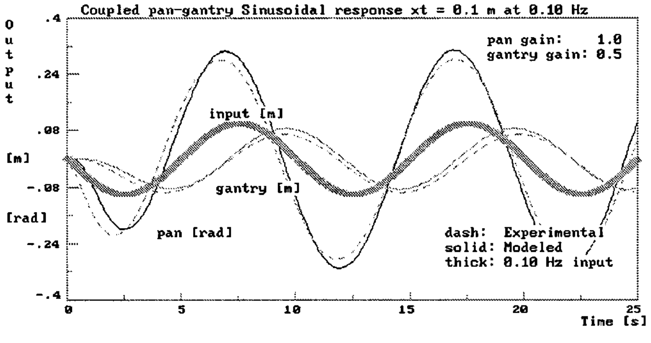
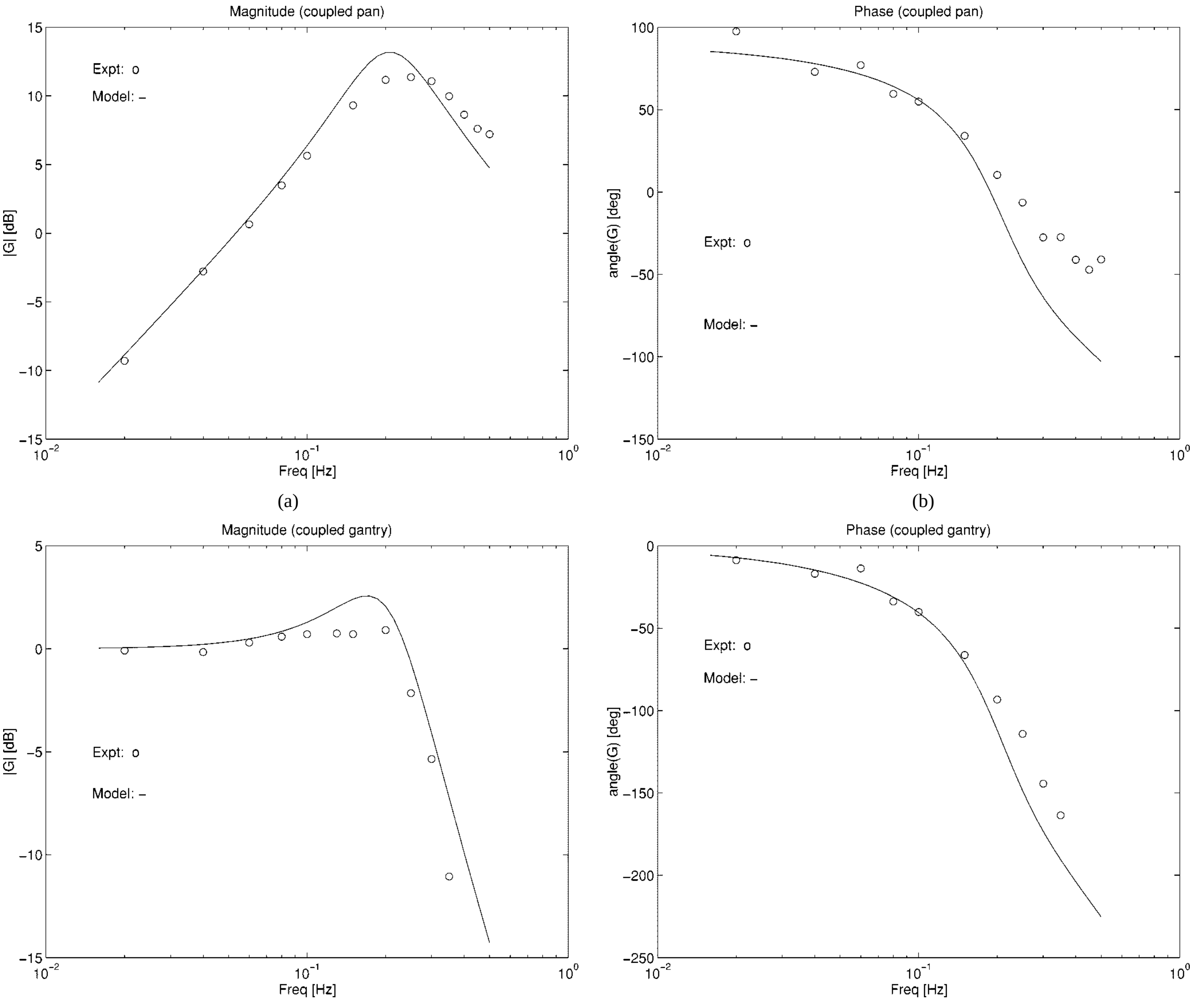


Fig. 14. Coupled pan-gantry response to a 0.1-Hz, 0.1-m amplitude sinusoidal target translation x_t (thick solid line). Both experimental (dash) and simulated (solid) responses shown.



(a) (b)

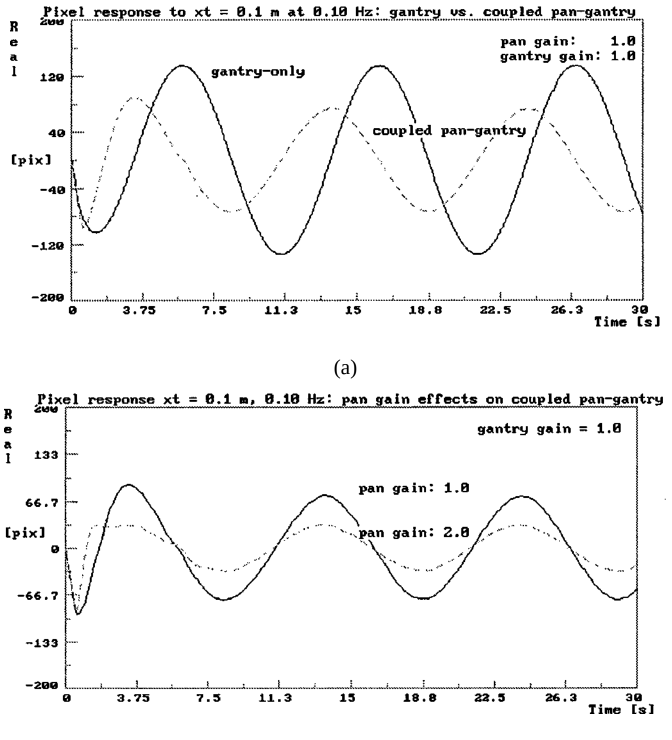
(c) (d)

Fig. 15. Bode plots of coupled pan-gantry: (a), (b) pan magnitude and phase, (c), (d) gantry magnitude and phase for $\lambda_p = \lambda_g = 1.0$. Both Matlab-generated (solid) and experimental (dot) results are shown. The pan's magnitude peaks at 0.189 Hz with a 12.95 dB gain and 0° phase angle. The gantry 3 dB down occurs at 0.293 Hz and −166° phase angle.

Bode phase and magnitude plots which were obtained in the same manner as in Section III-B. Both λ_p and λ_g were set to one for this plot. Fig. 15 shows the resulting pan and gantry Bode plots, respectively. Also shown are the Matlab generated Bode curves (solid) using the transfer function models (10) and (11).

The pan's Bode plot [Fig. 15(a) and (b)] has a resonant peak at 0.189 Hz (12.95 dB). Below this value, there is phase lead with amplified gain. At higher frequencies, gain is attenuated and hence filters high-frequency signals. Fig. 14 showed the pan phase lead and at a 0.1-Hz target frequency input, the Bode plots shows this lead to be +55.88°. The pan coupling improves gantry lag and bandwidth (−40.6°, 0.283 Hz) versus −52.8° and 0.166-Hz values when there is no coupling. The net effects can be summarized as follows: at low target frequencies (e.g., 0.020 Hz) the gantry handles tracking more dominantly, at high target frequencies (e.g., 0.80 Hz) the pan manages tracking more dominantly, and at medium target frequencies (e.g., 0.189 Hz) both pan and gantry share in the tracking task.

2) Peak-to-Peak Pixel Errors: Peak-to-peak pixel error, in response to a sinusoidal input, is a reliable measure of lag performance. A zero error signifies that there is no lag and means that the target is always centered in the camera's image plane. There will be nonzero errors for several reasons. First, the gantry has sluggish response (as measured by its limited bandwidth) and will lag behind a fast moving target (as measured by the Bode phase plot). At high enough target frequencies, the pan will also lag behind the target. Second, both the pan-only and gantry-only closed-loop systems are Type 1, which means that they only have a single integrator, $(z-1)$, in the characteristic equation. Type 1 systems always have tracking errors in response to a sinusoidal input. Partitioning however exploits both the pan's faster rise time (to compensate for the gantry's slower response) and gantry translation (to cut down camera-target bearing angle).



(a)

(b)

Fig. 16. (a) Pixel errors for both gantry-only ($\lambda_g = 1.0$) and coupled pan-gantry ($\lambda_g = \lambda_p = 1.0$) systems to a 0.1 Hz, 0.1 m amplitude target translation x_t. Note how coupling improves phase and has less peak-to-peak pixel error. (b) Increasing the pan gain decreases the pixel error but can yield gantry end-point vibrations due to larger accelerations.

Fig. 16(a) plots pixel error, $s[k] - s^*$, to a 0.1 Hz, 0.1 m amplitude target translation x_t. With $z^* = 0.25\,\text{m}, L = 0.10\,\text{m}$, and $f = 633$ pixels, the gantry-only system ($\lambda_g = 1.0$) has a peak error of 133 pixels. The net effect is that gantry's phase properties keep the target more than 5 cm off the image plane center. The coupled pan-gantry plot ($\lambda_g = \lambda_p = 1.0$) reveals both improved phase and a peak error of 73 pixels (less than 3 cm off-center).

Fig. 16(b) illustrates reduced pixel error as the pan's gain λ_p is increased. For $\lambda_p = 2.0$, the peak error is 33 pixels (1.3 cm off-center). Smaller peak errors can be achieved by increasing pan gain; however, the coupling also increases gantry start-up accelerations which increase end-point vibrations as seen in the first few seconds of the figure.

V. Partitioning Applications

The framework and synthesis of a partitioned controller, developed in the previous sections, were extended by coupling the tilt DOF to the gantry's vertical DOF (q_3 in Fig. 2). Visually servoing pan and tilt can center the camera's image plane over a target traveling in a general trajectory. Tilt coupling adds vertical gantry action and translates the camera up and down. The resulting system was used to track a person and robot hands.

A. People Tracking

An 80×80 pixel SSD tracker was initialized over the person's head and the partitioned system tracked a person casually walking, cornering, and ducking around the workcell perimeter. The person's motions were not planned but the overhead view in Fig. 17 gives the person's general trajectory. The 15 numbers give approximate positions of the person and correspond to the fifteen sequential image stills in Figs. 18 and 19. The image stills were acquired from two videotapes. In Fig. 18, a handheld video camera recorded the person and the scene, and in Fig. 19 the robot camera taped its field-of-view.

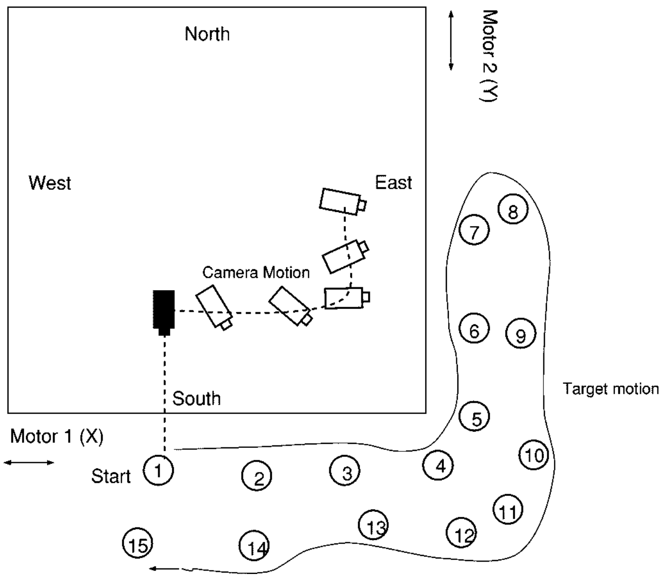


Fig. 17. The person's path around the workcell perimeter.

In tracking a person, the partitioned system illustrates several points. First, a region-based SSD tracker can be used to track geometrically complex targets, like a person's head. An SSD is a standard image processing technique and is simple to implement for the real-time pixel measurements of the image centroid's center-of-gravity. The SSD tracker uses correlation in measuring pixel positions and is quite robust to nondeterministic head motions such as bobs, sways, and turns.

Second, partitioning exploits the dynamic and kinematic attributes in a robot's DOF. The PTU's large bandwidth affords fast camera accelerations and by visually servoing pan and tilt, the image plane can be quickly centered over the target. Kinematically servoing the gantry through joint-coupling exploits its abilities to cut bearing angles and transport the camera throughout the workcell.

Third, by using joint data, additional kinematic servoing rules can be defined to exploit redundancy in a robot's DOF, as well as overcome joint limits. The gantry has two horizontal DOF, q_1 and q_2 (see Fig. 2). The horizontal DOF to couple with the pan can be determined by monitoring pan angle quadrant. As the person corners, coupling can be handed off from one horizontal DOF to the other. Another advantage is that joint limits can be handled. For example, in one people tracking experiment [15], the pan hits its physical joint limit while the person travels in the south-west corner (Fig. 17). Partitioning however shares the tracking task among the robot's DOF, and when panning is no longer possible, the gantry keeps tracking the target.

B. Robot Hand Tracking

The system was used to also track a robot hand which translates in a triangular trajectory seen in Fig. 20. The hand

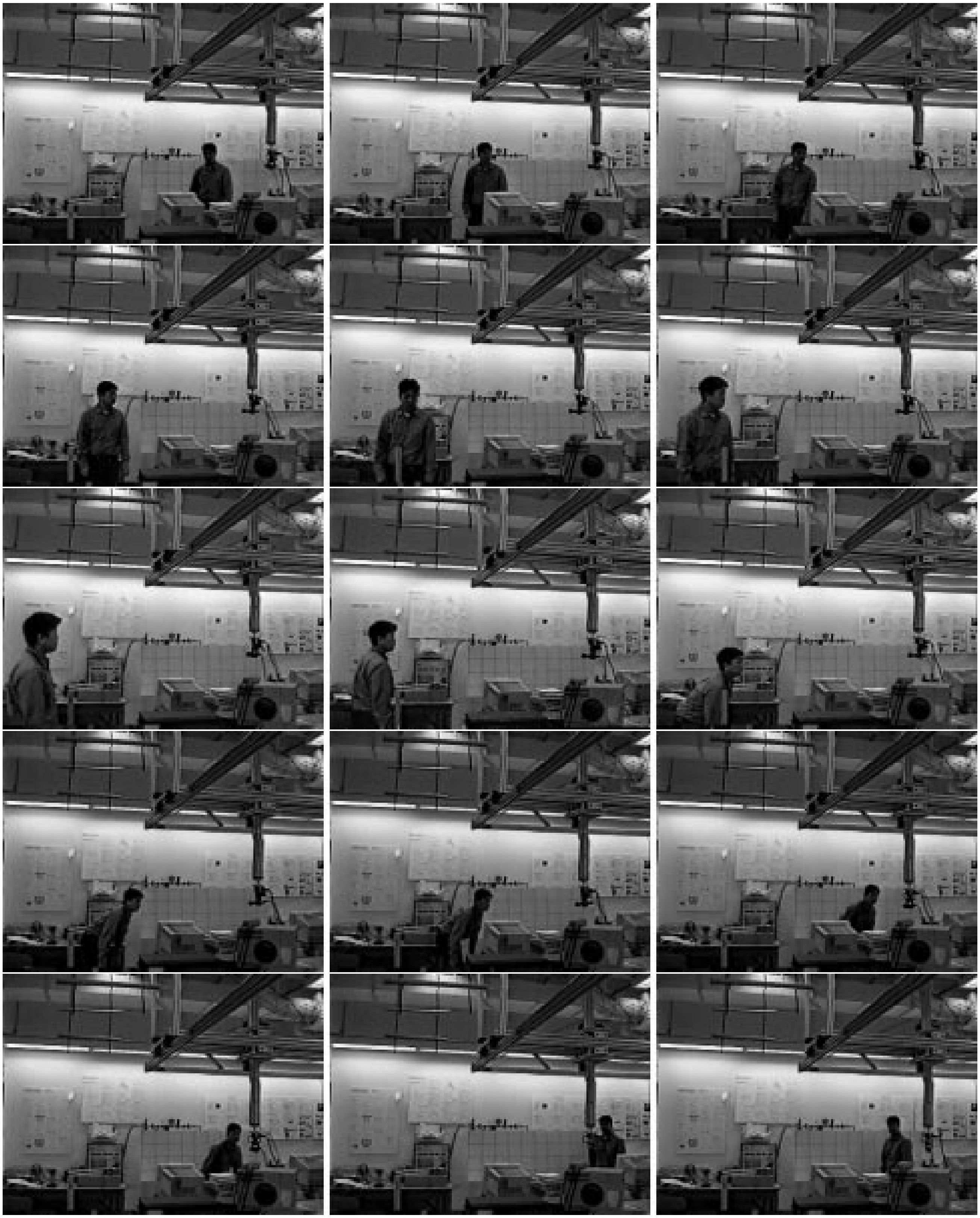


Fig. 18. Fifteen sequential images stills (top left to bottom right) captured by a handheld video camera.

first translates diagonally, 25 cm in Y and 35 cm in X, and stops. It then moves up 20 cm along Z, pauses, then returns to its home position traveling both down and diagonally forward. This hand trajectory (solid) and resulting partitioned camera position (dash) are shown in Fig. 21. SSD scale data were used to invoke the gantry DOF along the Y-axis. As the

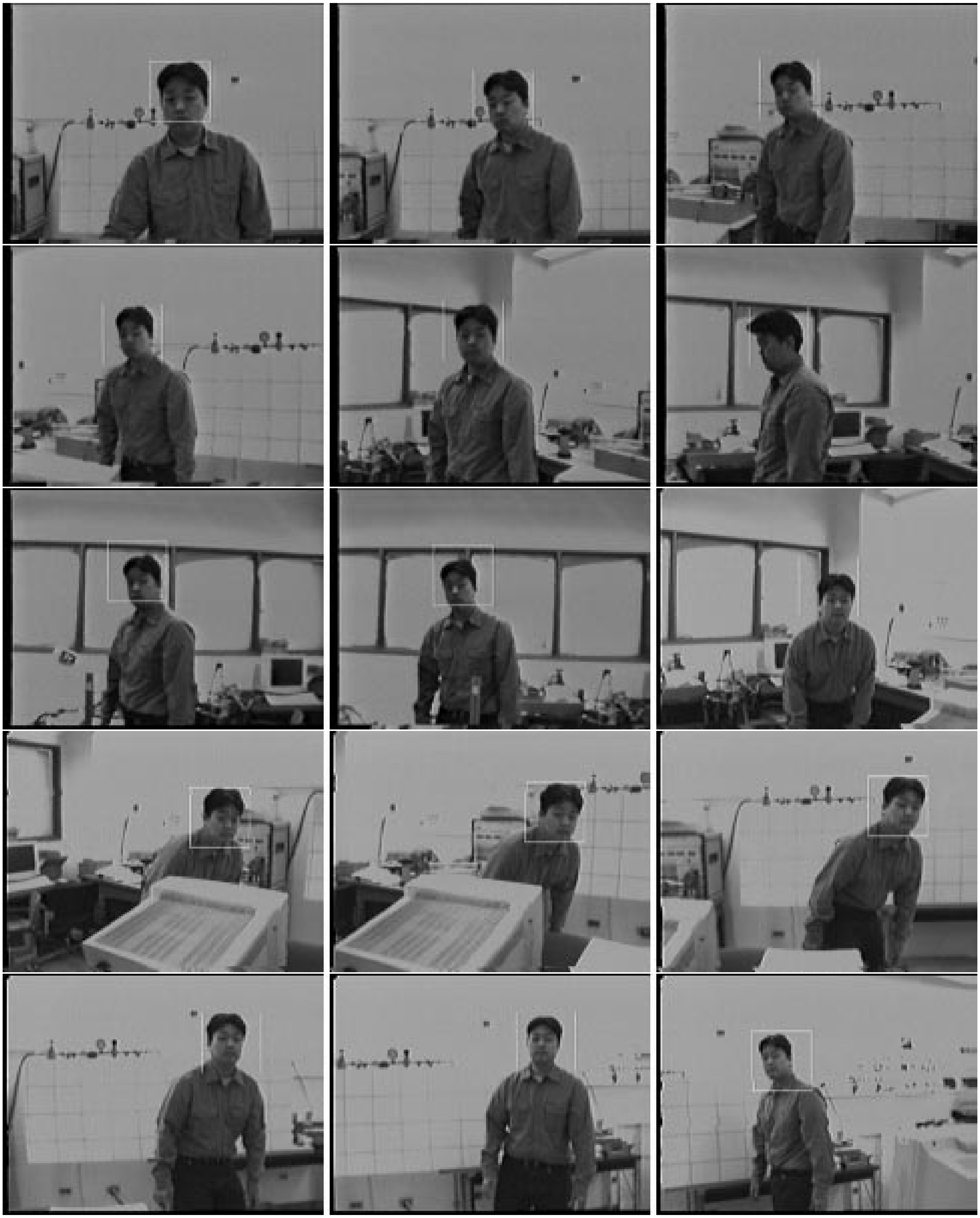


Fig. 19. Fifteen sequential image stills (top left to bottom right) captured by the robotic camera.

scale changes, the gantry velocity in this direction is proportionally servoed and thus adds depth regulation. Six sequential image stills taken while tracking the hand are shown in Fig. 22.

C. *Regulator Retrofit*

This application shows how the partitioning control law can be retrofitted to an existing visual servoing law, like a pose regulator. Our vision interest includes monitoring targets like

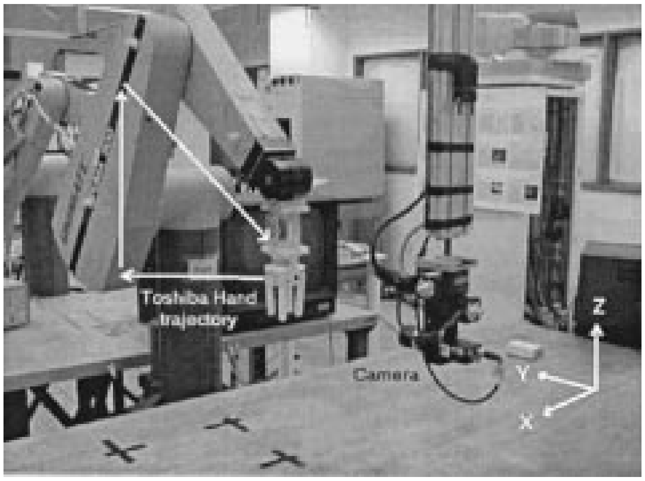


Fig. 20. The Toshiba Hand moves in the triangular trajectory shown. Pan and tilt couple gantry translations along X and Z axis using partitioned control. Scale data regulates camera-to-target distance along Y.

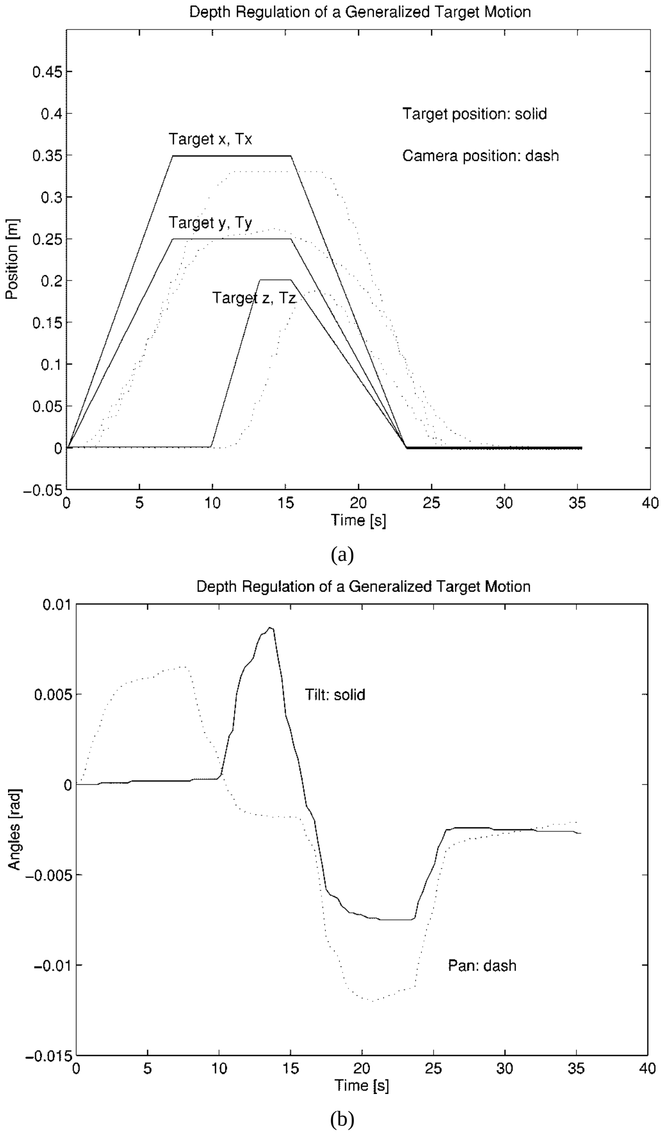


Fig. 21. (a) The target moves in a triangular trajectory (solid Target x, Target y and Target z lines). The gantry positions the camera under partitioned control (dashed T_x, T_z lines). Scale data regulates the camera-to-hand distance (dashed T_y line) and converges whenever the target stops. (b) the pan and tilt response.

tools and workpieces that move in an assembly workcell. As such, there are critical times like during tool alignment and part manipulation operations, when a desired camera-to-target pose needs to be established and maintained. Target motions are typically slow compared to when the tool or workpiece undergoes transport operations. Since such targets motions are slow, pose regulating a camera is possible.

In revisiting the block tracking experiment, we retrofitted the partitioned controller to the pose regulator and defined *hard* and *soft* constraints. In the hard constraint case, the camera is servoed under regulator control and requires estimating target pose using a full image Jacobian (four co-planar points). In the soft constraint case, the partitioning control law actuates camera motions using only one image feature (one of the four co-planar points). The net effect is that the target can be tracked using partitioning when its motions are fast and pose can be regulated when target motions are slow. Tracking in this manner is highlighted experimentally tracking a block target in Fig. 23.

The block's sidelengths are known and four SSD trackers are placed at each corner. The block translates at 10 cm/s, slowly curves at 2 cm/s and then stops. As mentioned in Section II, pose regulation fails at fast target speeds due to the gantry's sluggish response. But by retrofitting partitioned control to the regulator, the fast target translation can be handled. The camera-to-target pose was reestablished under regulator control when the block moved slowly as seen by the similarity in the initial and final pose (Fig. 23 top left and bottom right images, respectively). Fig. 24 shows the gantry and PTU position and velocity responses. The dashed line (added) is when camera servoing switches from partition to regulator control. Asymptotic convergence can be observed as pose is reestablished under regulator control.

VI. Conclusions and Potential Extentions

Ideally one would like to simply mount a vision system on a robot to easily perform vision-based tasks. Implementation however is confounded by many design factors. Robots with multiple DOF, disparate servo update rates, and the required task performance, are factors that convolute design decisions. When visual servoing every robot DOF is not necessary, design questions arise, such as which DOF to choose, and what roles, if any, do remaining DOF play in the task. Without any analytical framework, the visual-servoing designer resorts to *ad hoc* gain tuning and try-and-see implementations.

An analytical framework however, gives quantitative measures of a robot's DOF kinematic and dynamic attributes, and potential insights, for visual servoing design synthesis. In our frequency-based framework, we designed a multi-input multi-output controller we called partitioning using insight obtained from system identification of individual DOF. Partitioning exploits each DOF's attributes by defining an underlying joint-coupling in the control law. By taking advantage of the PTU's large bandwidth, visually servoing pan and tilt keeps the camera's image plane centered over the moving target. By exercising the gantry's ability to transport the camera throughout the workcell, tracking improves because of the reduced camera-target bearing angle. Improved lag was seen by

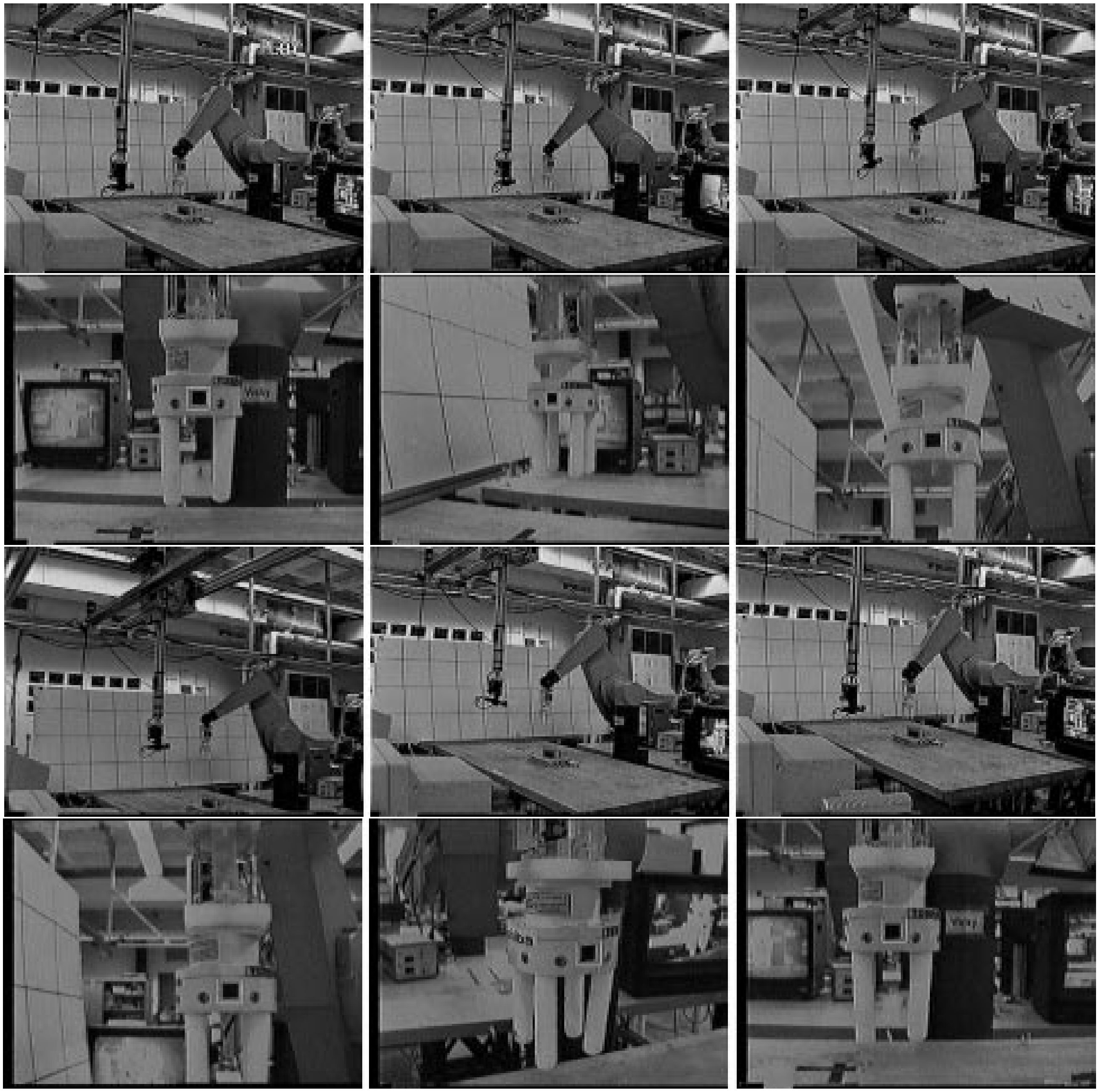


Fig. 22. Six sequential snapshots while regulating camera-to-hand distance. Rows 1 and 3 are image stills from a video camera and Rows 2 and 4 are image stills from the robot camera. Top left to bottom right: (top left) from home the hand moves diagonally away from and to the left of the camera and results in camera pan and side translation X. The hand moves up and results in a camera tilt and upward translation Z. The hand then moves both downward and toward the camera, returning to its home position (bottom right). The final camera position has the target centered and parallel to its image plane, with the desired camera-to-target distance.

the coupled pan-gantry Bode plots and peak-to-peak pixel error measurements. Partitioning was then applied to track targets like people and robot hands without *a priori* knowledge of their motion trajectories. Partitioning can also be retrofitted with other control laws to regulate pose [15]. The net effect is that with partitioning, we meet our end goal of visually tracking geometrically complex targets like grippers, parts, and tools that move in a large assembly workcell.

A paradigm one notices in the visual-servoing literature is that only image data is used to actuate a robot's DOF to effect camera pose. Most robots however come readily equipped with additional sensors such as joint encoders. Our system suggests sensor fusion; image and kinematic data are combined in a multi-input control strategy that defines an underlying joint-coupling and achieves improved visually-servoed performance. Analysis of the resulting phase characteristics reveals that large bandwidth DOF, which are visually servoed, physically act as lead compensators for DOF with slow response times. One extension is to incorporate additional large bandwidth sensors, such as accelerometers, with vision. Combined with fast actuators, one can use frequency response to synthesize a system that mimics human oculomotor lead/lag compensation. Such a system would afford quick tracking and would be robust to end-point vibrations.

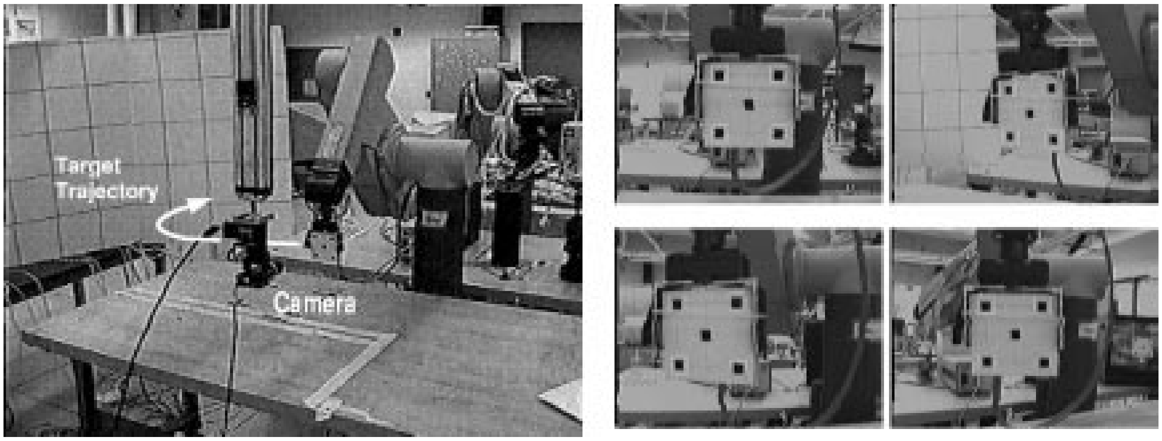


Fig. 23. Hard and soft constraints: A Puma moves the block in a curvilinear trajectory (white arrow in the large left photo). The block translates at 10 cm/s, curves slowly at 2 cm/s, then stops. The block's sidelengths are known and 4 SSDs track each corner. The 4 smaller photos on the right are sequential image stills from a videotape while recording the camera's field-of-view. The top left image is the desired camera-to-target. pose. Partitioned control tracks the fast moving block (top right). As the block slowly curves and stops, regulator control begins (bottom left) and establishes the desired pose (bottom right).

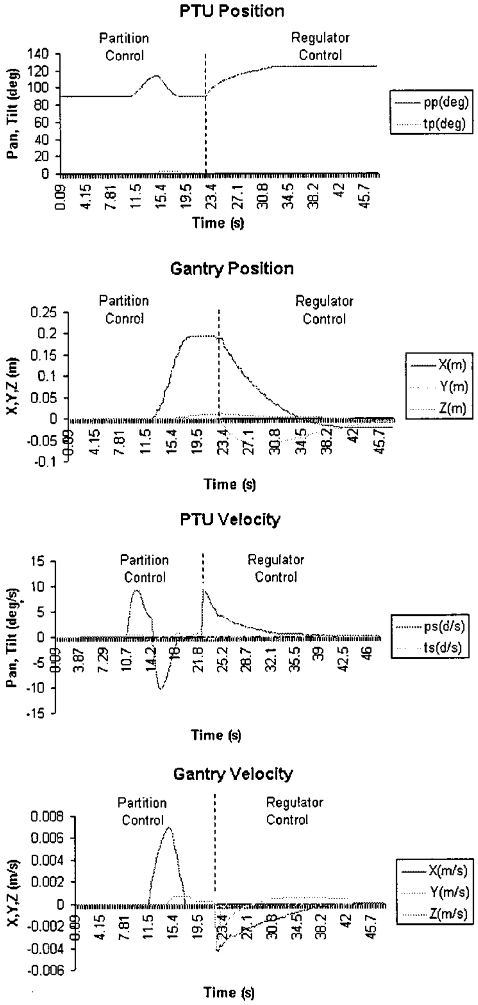


Fig. 24. PTU and gantry position and velocity responses. The dashed vertical line was added to emphasize the time when the robot switches from partition to regulator control.

Motion and sensor planning designers take advantage of a robot's attributes when servoing end-effector trajectories. A second potential extension to our framework is to quantify and analyze volumes swept by the robot's individual DOF. This can give further insight on synthesizing a robot-vision solution. Combined with dynamic attributes such as bandwidth, a performance cost can be defined that weights the servoing of individual DOF. A prototype linear quadratic regulator for partitioned tracking was designed in [16].

There are limitations in our system, some of which can be handled through better hardware and software. The PTU's serial latencies increase program loop time and thereby reduce the rate at which camera velocities can be updated. Access to the PTU's microcontroller interrupts and a real-time operating system would define a precise timing budget and overcome this limitation.

Our tracking tasks only requires simple image processing and SSD region-based trackers were used to measure the image centroid's center of gravity pixel positions. Often one wants to regulate the camera-to-target distance for desired image resolution. Measuring depth while both target and monocular camera are moving in *a priori* unknown trajectories is an open problem. We used SSD scale data for limited depth regulation, but under gross changes in target pose or when image features are completely occluded, our system will fail.

Image understanding and processing are integral aspects in designing a robot-vision system. However, in the big picture of designing "eyes" for robots, this paper points to considering "eyeball" design. In other words, visually-servoed tasks should considering the mechanisms, like PTU's, that servo the camera. Our framework and resulting joint-coupled controller underscore the advantages of such considerations and their potential in synthesizing solutions.

References

[1] F. Bensalah and F. Chaumette, "Compensation of abrupt motion changes in target tracking by visual servoing," in *Proc. IROS'95*, Pittsburgh, PA, Aug. 1995, pp. 181–187.

[2] A. Castano and S. Hutchinson, "Visual compliance: Task-directed visual servo control," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 334–342, July 1994.

[3] F. Chaumette, P. Rives, and B. Espiau, "Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1991, pp. 2248–2253.

[4] F. Chaumette and A. Santos, “Tracking a moving object by visual servoing,” in *Proc. 12th World Congress IFAC*, vol. 9, Sydney, Australia, July 1993, pp. 409–414.

[5] P. I. Corke, “Dynamics of visual control,” in *Proc. IEEE Int. Conf. Robotics and Automation*, Workshop M-5, San Diego, CA, May 1994.

[6] P. Corke *et al.*, “Design, delay and performance in Gaze control: Engineering and biological approaches,” in *The Confluence of Vision and Control*, Kriegman *et al.*, Eds. New York: Springer-Verlag, 1998, pp. 146–158.

[7] J. L. Crowley, M. Mesrabi, and F. Chaumette, “Comparison of kinematic and visual servoing for fixation,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1995, pp. 335–341.

[8] J. T. Feddema and G. C. S. Lee, “Weighted selection of image features for resolved rate visual feedback control,” *IEEE Trans. Robot. Automat.*, vol. 7, pp. 31–47, Feb. 1991.

[9] B. K. Ghosh, “Nonlinear estimation schemes for visual servoing,” in *Proc. IEEE Int. Conf. Robotics and Automation*, , Workshop M-5, San Diego, CA, May 1994.

[10] G. D. Hager and K. Toyama, “X vision: A portable substrate for real-time vision applications,” Dept. Comput. Sci., Yale Univ., Tech. Rep., 1995.

[11] S. Hutchinson, G. Hager, and P. Corke, “A tutorial on visual servo control,” *IEEE Trans. Robot. Automat.*, vol. 12, pp. 649–670, Oct. 1996.

[12] K. Hashimoto, Ed., *Visual Servoing, World Scientific Series in Robotics and Automated Systems*, 1993, vol. 7.

[13] A. J. Koivo, “Real-time vision feedback for servoing robotic manipulator with self-tuning control,” *IEEE Trans. Syst., Man., Cybern.*, vol. 21, pp. 134–142, Jan. 1991.

[14] P. Y. Oh and P. K. Allen, “Design of a partitioned visual feedback controller,” in *Proc. IEEE Int. Conf. Robotics and Automation*, Leuven, Belgium, May 1998, pp. 1360–1365.

[15] ——, “Performance of a partitioned visual feedback controller,” in *Proc. IEEE Int. Conf. Robotics and Automation*, Detroit, MI, May 1999.

[16] P. Y. Oh, “Integration of joint-coupling for visually servoing a 5-DOF hybrid robot,” Ph.D. dissertation, Dept. Mech. Eng., Columbia Univ., Oct. 1999.

[17] N. P. Papanikolopoulos and P. K. Khosla, “Adaptive robotic visual tracking: Theory and experiments,” *IEEE Trans Automat. Contr.*, vol. 38, pp. 429–445, Mar. 1993.

[18] W. J. Wilson, “Visual servo control of robots using Kalman filter estimates of robot pose relative to work-pieces,” in *Visual Servoing*, K. Hashimoto, Ed: World Scientific, 1993.

Paul Y. Oh (M’95) received the B.Eng. degree (with honors) in mechanical engineering from McGill University, Montreal, PQ, Canada, in 1985. Awarded a Quebec/Korea Ministry of Science and Technology Research Fellowship, he pursued and received the M.Sc. degree from the Mechanical Design and Production Engineering Department at Seoul National University, Seoul, Korea, in 1992. He received the Ph.D. degree in mechanical engineering from Columbia University, New York, in 1999, while pursuing visual-servoing research in the Computer Science Department.

He is currently an Assistant Professor at the Department of Mechanical Engineering and Mechanics, Drexel University, Philadelphia, PA. His research interests include visual-servoing, mechatronics, human augmentation, and sensors.

Peter K. Allen (M’85) received the A.B. degree in mathematics-economics from Brown University, Providence, RI, the M.S. degree in computer science from the University of Oregon, Eugene, and the Ph.D. degree in computer science from the University of Pennsylvania, Philadelphia.

He is a Professor of Computer Science at Columbia University, New York. His current research interests include real-time computer vision, dextrous robotic hands, 3-D modeling, and sensor planning.

In recognition of his work, Prof. Allen has been named a Presidential Young Investigator by the National Science Foundation.