Proceedings of the 1998 IEEE
International Conference on Robotics & Automation
Leuven, Belgium • May 1998

Design of a Partitioned Visual Feedback Controller

Paul Y. Oh and Peter K. Allen
Dept. of Computer Science, Columbia University, New York, NY 10027 *

Abstract

Image-based servoing systems are often used to track moving targets and their underlying control architecture is a regulation of the image. This regulation is a function of rigid camera-to-target geometric constraints. Satisfying such constraints requires that the robot motors have sufficient velocity bandwidths, and often these bandwidths are limited. This paper lays down the foundation for a partitioned controller. Such a controller would coordinate a camera's DOF into a synergistic move to overcome bandwidth limitations. Tracking experiments are shown on a custom designed 5 DOF gantry robot which highlight the limitations of regulator-based control, as well as show how partitioning can be used to achieve more robust control.

1 Introduction

Closed loop control can increase the performance of robotic tasks. Controllers can be designed to handle external disturbances or uncertainties in the environment. However, closed loop control requires sensory feedback, and vision is one way to provide this feedback. The visual task often is to robotically servo the camera to maintain a desired visual pose of the moving target. Typically an image Jacobian is used [1], [4], [5],[8], [9], [10], [13], [14], and [16]. An excellent review of visual servoing is given in [6]. In essence, these systems are image regulators.

There are two major limitations for regulator-based tracking systems. The first is motor speed bandwidth. That is, for fast moving targets, the motors may not have the speed capability to keep the target's image features in view. Tracking will then fail since there is nothing to regulate with. The second is the robustness of vision-based algorithms to image changes. For example, in the case of tracking, target accelerations may appear as discrete jumps in pixel information. Filters often have to be implemented in software to distinguish such jumps from noise [17], [3]. This paper introduces a partitioned controller design that addresses these limitations.

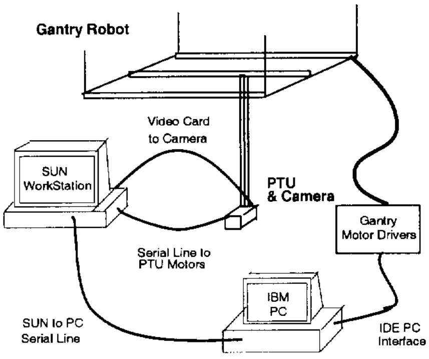

Figure 1: Robot and Local Network Configuration

Motor	Precision	Range	Max. Velocity
X	7.50×10^{-6} m/step	1.94 m	≈ 0.7 m/s
Y	7.50×10^{-6} m/step	4.43 m	≈ 0.5 m/s
Z	3.75×10^{-6} m/step	0.68 m	≈ 0.5 m/s
Pan	0.0514 deg/step	±150 deg	≈ 128 deg/s ec
Tilt	0.0514 deg/step	-47 to +31 deg	≈ 113 deg/s ec

Table 1: Joint Motor Properties

2 Monitoring a Robotic Workcell

Our interest in visual control is in building a robotic system that can monitor an entire assembly workcell. Our workcell contains 2 Puma 560 robots and we have custom built a 5-DOF robot to position a camera anywhere in the assembly workcell to be able to monitor the operation of the Puma's. The robot has three translational DOF which configure it as a Cartesian Gantry robot and can cover a workspace of $3.6 \times 6.4 \times 1$ cubic meters. At the gantry's end-effector is a 2-DOF pan-tilt unit (PTU) made by Directed Perception on which a camera is mounted. The robot's 3 translational DOF's are controlled by stepper motors with a IBM PC bus interface. The PTU has its own serially interfaced stepper motor control system. Table 1 gives the relevant joint motor properties. The net effect of this is a 5-DOF hybrid robot that can monitor the workspace (see Figure 1).

In using this robot for visual monitoring tasks, a number of problems must be addressed. First, the entire mass of the gantry's links may need to be moved as it

*This work was supported in part by an ONR/DARPA MURI award ONR N00014-95-1-0601, DARPA AASERT awards DAAHO4-93-G-0245 and DAAH04-95-1-0492, and NSF grants CDA-96-25374 and IRI-93-11877.

0-7803-4300-x-5/98 $10.00 © 1998 IEEE

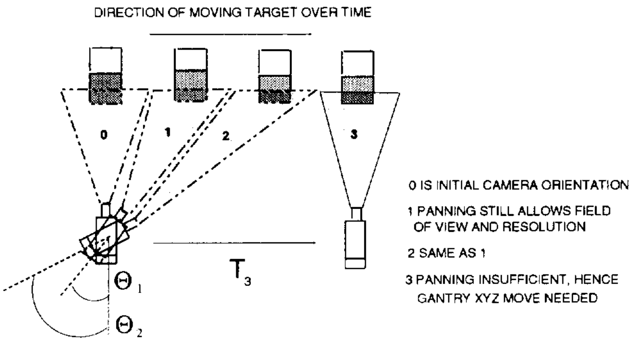
DIRECTION OF MOVING TARGET OVER TIME
0
1
2
3
T3
Θ1
Θ2
0 IS INITIAL CAMERA ORIENTATION
1 PANNING STILL ALLOWS FIELD OF VIEW AND RESOLUTION
2 SAME AS 1
3 PANNING INSUFFICIENT, HENCE GANTRY XYZ MOVE NEEDED

Figure 2: Monitoring with partitioned DOF's

translates. Second, the gantry motors' velocity bandwidths are limited, and are much less than the velocity bandwidths of the PTU's motors. The net effect of this is that performance for tracking moving targets with this robot is related to which degrees of freedom 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 pan-tilt is limited, and arbitrary pose configurations of the camera-to-target cannot be satisfied. If we allow all 5-DOF to be used, we then limit our tracking velocities. This paper describes a hybrid approach to solving the above problems, by building a partitioned controller that can be used to track objects using temporal as well as spatial constraints.

In our specific case, we have additional problems. Large accelerations on the gantry's translational DOF generate end-point vibrations. These vibrations are a disturbance and adversely affect image feature acquisition. Because of these vibrations, joint accelerations have to be limited below their mechanical capabilities. This again limits the ability to track fast moving targets.

Tracking with pose constraints alone may be overly restrictive. We can redefine the tracking task with *temporal* constraints as follows. The camera is constrained to keep the target in its general field of view. A more rigid camera-target pose constraint is maintained at certain *critical* times. For example, cameras are often used to track a part in a pick-and-place process. When the part is in-flight, the camera is actuated to keep the part in general visual contact. The part's pose information is most important during pick-up and place-down. It is at these critical times that the camera is positioned to maintain a desired camera-to-target pose. Figure 2 shows how the different DOF may be used to accomplish a monitoring task for a moving object.

Redefining the tracking task with temporal constraints may overcome limitations inherent in regulator-based tracking systems. For instance, some of a robot's joint motors have faster response times than others. The response time is dependent on torque loading. For example, a robot's base motor joint incurs a high startup cost due to the inertial load of all the robot's links. End-effector joints, with less link mass to actuate, accelerate quicker. This suggests a natural *partitioning* of a robot's DOF into fast and slow response time groups. The former group's motors actuate the camera quickly, keeping the target in a general field of view during which time the latter group's motors get up to speed. Additionally, end-effector joints tend to have smoother performance. For eye-in-hand configurations, this suggests less camera jerk and a more robust image capture thus curbing end-point vibrations.

A partitioned controller requires a fundamentally different controller architecture. However the image-based methods used in traditional tracking systems lend to its development. For example, the same filters (namely Kalman-based ones) that are used for robust image capture, can be used for target trajectory prediction. The development of these methods will be introduced in this paper and highlighted with experiments.

3 Image Based Tracking

Image-based visual servoing methods directly express an error function in terms of image features. They are less sensitive to calibration errors and computationally faster [12] than position-based "look-and-move" methods [16].

The image Jacobian (called the *Interaction Matrix* L^T in [8]) maps differential rates of change of image features in a camera's image space to the world task space

$$\frac{d\vec{f}}{dt} = L^T \frac{d\vec{r}}{dt} \tag{1}$$

Typically in visual tracking, one is interested in the inverse mapping. Since L^T may be under or over determined, a suitable pseudo-inverse must be calculated.

Since the image Jacobian depends on the image features selected, it is not unique. It is a design variable. Espiau et al. [8] prescribed possible image features for planar, cylindrical and spherical targets. Castano et al. [4] presented an interesting method of selecting an image set based upon visual compliance as a vision analog of force compliance. Allen et al. [1] and [2] demonstrated that planar motion exhibits elliptical optic flow patterns that can be used to quickly estimate the image Jacobian. Feddema et al. [9] avoided the pseudo-inverse computation by designing a square Jacobian matrix. The resulting image space to task space mapping is non-unique but used a best guess method for determining target pose. Papanikolopoulos et al. [14] used a model reference adaptive control method where the actual image Jacobian does not have to be determined. The captured image is referenced to a desired model of the mapping and the state parameters are updated on-line.

As mentioned in the Section 1, these are regulator-based trackers. In essence, this technique operates as follows. First, a reference image is captured and a $(n \times 1)$ reference vector s^* is defined from this image. This vector is a set of n image features that geometrically describe the desired camera pose relative to the target. Second, as the target moves, a new image is acquired and the image features are monitored. These image features form the elements of a $(n \times 1)$ vector s. An error function, (called the *task function* in [8]) is defined. This is a function of the difference between the acquired and reference image. This function describes the geometrical relationship the camera should have with respect to the target. Traditionally a *rigid* constraint is used as follows. For differential changes, the task function is defined as

$$e(\vec{r}(t), t) \doteq C\left(s(\vec{r}(t), t) - s^*\right) \tag{2}$$

$\vec{r}(t)$ is a 6×1 vector description at time t of the position and orientation of the target with respect to the camera. C is the inverse of the image Jacobian. Thus the *rigid* constraint $s(\vec{r}(t)) = s^*$ is perfectly achieved if $e = 0$. Since $e = e(\vec{r}(t), t)$ then

$$\frac{de}{dt} = \frac{\partial e}{\partial \vec{r}} T_c + \frac{\partial e}{\partial t} \tag{3}$$

where we define $T_c \doteq d\vec{r}/dt$. T_c will be our control law vector. It will be the camera's velocity in response to the rates of change in e. From (3) we have

$$T_c = \left(\frac{\partial e}{\partial \vec{r}}\right)^{-1} \left(\frac{de}{dt} - \frac{\partial e}{\partial t}\right) = \frac{de}{dt} - \frac{\partial e}{\partial t} \tag{4}$$

where $\partial e/\partial \vec{r} = CL^T = I$, can be set to the identity matrix under certain circumstances [15]. If we design a *desired* evolution of de/dt we can *pilot* the camera's velocity response. For example, we could design e to converge asymptotically by setting

$$\frac{de}{dt} \doteq -\lambda e \tag{5}$$

with $\lambda > 0$. Consequently, our control law becomes

$$T_c \doteq -\lambda e - \frac{\partial e}{\partial t} \tag{6}$$

The last term of (6) $\partial e/\partial t$ is the target's velocity. That is, from (3), and $\partial e/\partial \vec{r} = I$ we have

$$\frac{de}{dt} = T_c - \frac{\partial e}{\partial t} \tag{7}$$

If camera's velocity is identical to the target's velocity, then $s(\vec{r}(t)) \equiv s^*$ and de/dt would be 0. Thus,

$$T_c = \frac{\partial e}{\partial t} = T_{target} \tag{8}$$

Since the target's velocity is not known a priori, some estimate of $\partial e/\partial t$ will be necessary for perfect tracking. Kalman filtering can be used for estimating the target velocity [3]

4 Tracking Implementation

Figure 3 shows a photo of the target. This target is the Toshiba FMA gripper. It is a pneumatic-driven four-fingered gripper. Each finger is made up of 3 stretchable flexible chambers and can be positioned into a variety of different grasps by changing the pressure in each chamber. Because this gripper is devoid of sensors, we are interested in using vision to monitor its pose during assembly tasks.

To uniquely determine its pose, typically four coplanar fiducial marks are selected. The centroid of each mark has a vertical, v, and horizontal u, pixel associated with it. Assuming perspective projection it can be shown [12] that the following equation

$$\begin{bmatrix} \frac{du}{dt} \\ \frac{dv}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{f}{z} & 0 & \frac{u}{z} & \frac{uv}{f} & \frac{-f^2-u^2}{f} & v \\ 0 & -\frac{f}{z} & \frac{v}{z} & \frac{f^2+v^2}{f} & -\frac{uv}{f} & -u \end{bmatrix} T_c \tag{9}$$

where

$$T_c = \begin{bmatrix} \frac{dx_c}{dt} & \frac{dy_c}{dt} & \frac{dz_c}{dt} & \omega_{x_c} & \omega_{y_c} & \omega_{z_c} \end{bmatrix}^T \tag{10}$$

leads toward the form of the image Jacobian. The above (2×6) matrix maps the velocity of a point (x, y, z) in the $\Re^3$ task space to a velocity of a point (u, v) in the $\Re^2$ (camera) image space. Here, f is the camera lens focal length. The subscript c denotes that the variable is with respect to the camera frame. The frame's origin is at the camera's lens. z_c is along the optic axis and points towards the target.

Using four fiducial points will lead to an (8×6) matrix (see [5] for full form). In tracking, one needs to compute T_c given $d\vec{f}/dt$. This requires calculating the pseudo-inverse of L^T. Furthermore, necessary robot joint velocities must be computed using T_c and the robot's manipulator Jacobian J:

$$\frac{d\vec{q}}{dt} = [J]^+ T_c \tag{11}$$

where the (5×6) matrix $[J]^+$ is the pseudo-inverse of J. The net effect is two expensive matrix inversion computations which limit the motor command update rate, additionally limiting tracking performance.

4.1 Visual & Kinematic Servoing

Kinematic-servoing refers to the use of a robot's kinematics (joint positions and velocities through encoders) as feedback for end-effector positioning. It is much quicker than visual-servoing but requires accurate calibration. On the flip side visual-servoing is tolerant to calibration errors, but is computational expensive and requires image feature robustness. Crowley, [7] suggests a duality existing between these two servoing methods.

Our partitioned tracking system uses a hybrid control law composed of two different servoing commands. The first is a visually-servoed command. Here, a single SSD is used to command the robot's pan and tilt joint velocities as follows:

Visual-servoing:

$$\begin{bmatrix}\frac{du}{dt}\\ \frac{dv}{dt}\end{bmatrix} = \begin{bmatrix}\frac{uv}{f} & -\frac{f^2+u^2}{f}\\ \frac{f^2+v^2}{f} & -\frac{uv}{f}\end{bmatrix} T_c \text{ where } T_c = \begin{bmatrix}\omega_{x_c}\\ \omega_{y_c}\end{bmatrix} \tag{12}$$

where ω_{x_c} and ω_{y_c} are the image feature's rotational velocities with respect to the camera's frame. These are mapped to the pan and tilt joint velocities, $\dot{q}_{\text{pan}}$ and $\dot{q}_{\text{tilt}}$ respectively using a (2×2) manipulator Jacobian. This yields:

$$\begin{bmatrix}\frac{dq_{\text{pan}}}{dt}\\ \frac{dq_{\text{tilt}}}{dt}\end{bmatrix} = \begin{bmatrix}-\frac{\omega_y}{\cos q_{\text{tilt}}}\\ \omega_x\end{bmatrix} \tag{13}$$

The result of this command keeps the target centered in the camera's field-of-view. The second command is a kinematically-servoed command. Here pan and tilt angle position feedback are used to proportionally command the gantry's cartesian joint velocities:

Kinematic-servoing:

$$\begin{bmatrix}\frac{dq_{\text{horiz}}}{dt}\\ \frac{dq_{\text{vert}}}{dt}\end{bmatrix} = K_g \begin{bmatrix}q_{\text{pan}} - q^*_{\text{pan}}\\ q_{\text{tilt}} - q^*_{\text{tilt}}\end{bmatrix} \tag{14}$$

where K_g is a proportionality gain constant. q^*_{pan} and q^*_{tilt} are reference setpoint pan and tilt angles respectively. The result of this command translates the camera in the direction of pan and tilt.

We note that since only a (2×2) image Jacobian and (2×2) pan-tilt manipulator Jacobian need to be inverted, motor update rate is quickly achieved. This combined with both the PTU's quick response times and image robustness of a single SSD affords tracking of higher target accelerations.

5 Experimental Tests in Tracking

Two sets of experiments were conducted to highlight both motor bandwidth limitations using a traditional tracking system and increased tracking performance using the partitioned controller.

In the first experiment, the Toshiba gripper was attached to a Puma and was visually tracked using four co-planar fiducial marks. The distance between each fiducial mark was known. In tracking tests, rigid constraints on the camera-gripper pose were used. The pose required that (a) the gripper's image was centered in the camera's field of view; (b) a predefinable camera-to-gripper distaince z^* was maintained; and (c) the camera's image plane remained parallel to the gripper. Each fiducial mark was tracked with a (40×40) window using Hager's X Vision sum-of-square-difference (SSD) trackers [11]. This allowed us to do tracking in real-time at the 30 fps video rate. In all tests, the lens focal length, f, was 12.5 *mm.* and z^* was set to 25 *cm.*

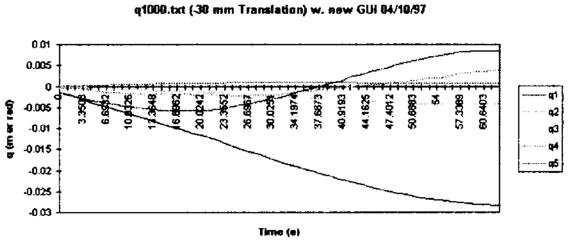

Figure 4: Step Input Joint Position Response

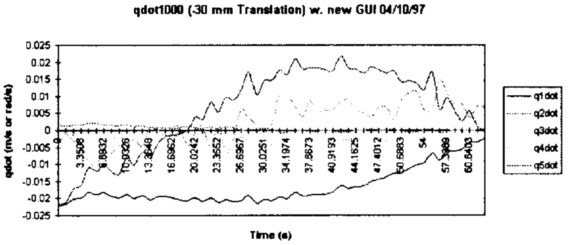

Figure 5: Step Input Joint Velocity Response

The step input was carried out as follows. First, a reference image was defined (Figure 3-Left) by placing a SSD tracker on each fiducial mark. The gripper was then displaced 3 cm along a line parallel to the camera's horizontal axis i.e. the gantry's $-z_0$ axis. Motor actuation was not enabled while the gripper was moved. Once the gripper was in place (see Figure 3-Middle), the camera was then allowed to move. Figure 3-Right shows the image after the camera moved. The regulator has successfully moved the gantry holding the camera to recover the reference image. Figures 4 and 5 show the position and velocity response of the five motors when the controller gain, λ, was set to 0.1. $\partial e/\partial t$ in equation (6) represents the target's absolute velocity. Since the gripper is not in motion once the camera is allowed to move this term was set to zero.

We note several points in figures 4 and 5. First, there is asymptotic convergence, and e indeed approaches zero. This suggests that the control law is stable. Second, since translation was parallel to the camera's image plane, the corresponding joint, q_0 approaches -3 cm, hence the image Jacobian design appears to be dependable. Third, there is some initial pan and tilt actuation from 0 to 37 *sec*, and some reverse correction from 37 to 61 *sec.*

This last point highlights the different motor speed bandwidths. Ideally if all motors had the same bandwidth only joint q_0 would actuate because the tar-

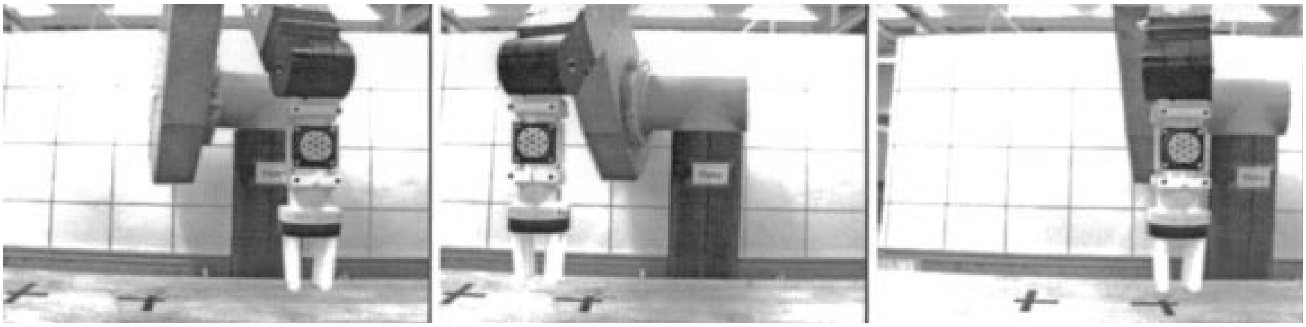

Figure 3: (Left) Reference Image (Middle) Gripper in position (Right) Final image after camera movement. The regulator has correctly moved the camera on the gantry to recover the reference image

get was translated parallel to this axis only. In reality, the PTU motors have a faster response time than the gantry's translational motors. The PTU motors thus react to the step input quicker than the gantry can translate. Once the gantry does translate toward its commanded position, the PTU motors compensate for their initial reaction. This underscores how partitioning may naturally be taken advantage of to track fast moving targets. The control gain λ was then increased to take advantage of the PTU's faster response times. However, this led to poorer results. The gantry's translational DOF would overshoot and led to oscillatory behavior.

We also note that motion tracking tests were conducted. We found that tracking was unreliable at target speeds greater than 2 cm/s. The tracking system would lose the four SSDs whenever the target was suddenly accelerated. A Kalman-filter estimator was implemented. However there is a startup time associated with error covariance matrix convergence. We found that at fast accelerations, the target may leave the camera's field-of-view during this convergence time.

For the second set of experiments, the partitioned control law (Eqns 12-14) was implemented and pose constraints were relaxed. Here, the reference setpoint pan and tilt angles, q^*_{pan} and q^*_{tilt} were set to 90 deg and 0 deg respectively. These setpoint angles were the starting positions of the pan and tilt joints upon tracking initialization. The target was commanded to move 0.4 m at 8 cm/s parallel to the camera's image plane. Figures 6, 7, 8 and 9 show the resulting motor velocities and positions. Figure 6 shows the gantry's horizontal velocity ramping up in response to the increase in the pan velocity (Figure 7). At approximately 5.0 $secs$, when the target has stopped moving, we see the pan velocity passes 0 deg/s and goes through a sign reversal. From approximately 5.0 to 12.0 sec we see a consequent gantry deceleration. Figures 8 and 9 confirm that camera has translated 0.4 m and its pan and tilt angles are reconfigured to their setpoint positions.

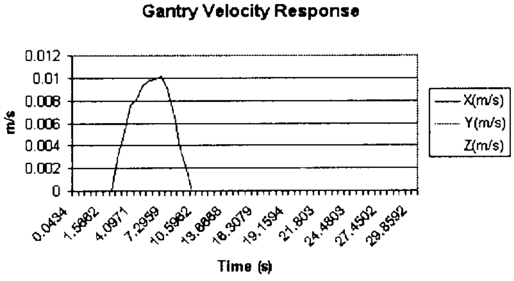

Figure 6: Partitioned Cartesian Velocity Response

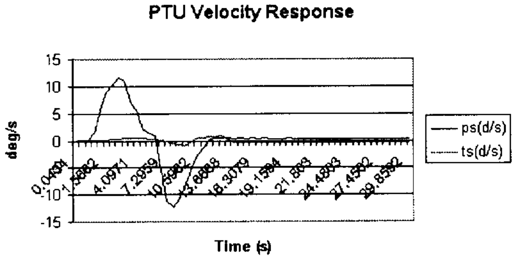

Figure 7: Partitioned PTU Velocity Response

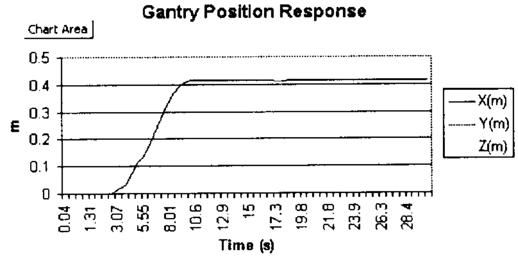

Figure 8: Partitioned Cartesian Position Response

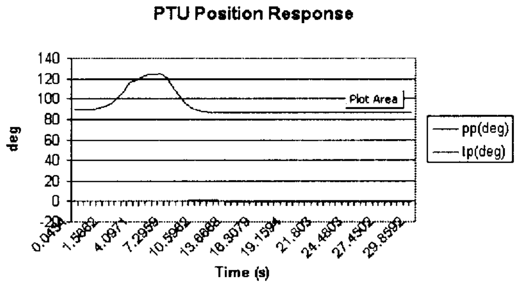

Figure 9: Partitioned PTU Position Response

6 Conclusion and Future Research

The paper emphasized the problems associated with regulator-based tracking systems using rigid geometrical constraints. Experiments showed that problems exist that limit the ability for a machine-vision system to track fast moving targets. These problems are both mechanical and image processing related. Namely, robot joints motors have limited velocity bandwidths and image feature detection is sensitive to noise. Despite these problems, we have been able to effect a stable control of the different degrees-of-freedom by partitioning them and understanding the limitations of each controller. By redefining the tracking task with temporal constraints we can overcome some of the problems involved and have tracked targets moving at 20 cm/s.

Using a single SSD however, poses two limitations, namely a loss of target-camera depth and orientation. The partitioned motor responses (Figures 6- 9) suggest that we can determine when the target has stopped moving or changed direction. This occurs when the gantry has reached its maximum velocity or when the PTU reverses direction. When the target is motionless, its position can become a reference point for triangulation and depth can then be computed. The gantry's cartesian DOFs can then maneuver the camera and reestablish depth constraints.

Target orientation determination poses a more significant problem. We are presently examining motor responses as a result of the coupling of visual and kinematic servoing. We are currently implementing this approach for monitoring our workcell via visual servoing.

References

[1] Allen, P.K., et al "Real-time Visual Servoing", *IEEE Int Conf on Robotics & Automation* p 851 4/91.

[2] Allen, P.K., et al "Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System", *IEEE Trans Robotics & Automation,* V9 N2, p 152, 4/93.

[3] Bensalah, F., Chaumette, F., "Compensation of Abrupt Motion Changes in Target Tracking by Visual Servoing", *Proc IROS '95* p 181, Pittsburgh, 8/95.

[4] Castano, A., Hutchinson, S., "Visual Compliance: Task-Directed Visual Servo Control", *IEEE Trans Robotics & Automation,* V10, N3, p 334, 6/94.

[5] Chaumette, F., Rives, P., Espiau, B., "Positioning of a Robot with respect to an Object, Tracking it and Estimating its Velocity by Visual Servoing", *IEEE Int Conf Robotics & Automation,,* p 2248, 1991.

[6] Corke, P. I., "Visual control of robot manipulators - A review," in *Visual Servoing*, K. Hashimoto, Ed. Singapore: World Scientific, 1993, pp 1.

[7] Crowley, J.L., Mesrabi, M., Chaumette, F., "Comparison of Kinematic and Visual Servoing for Fixation", *IEEE Int Conf Robotics & Automation,* pp 335, 1995.

[8] Espiau, B., Chaumette, F., Rives, P., "A New Approach to Visual Servoing in Robotics", *IEEE Trans Robotics & Automation*, V8 N3 pp 313, 6/92

[9] Feddema, J.T., Lee, G.C.S., "Weighted Selection of Image Features for Resolved Rate Visual Feedback Control", *IEEE Trans Robotics & Automation,* V7 N1 2/91.

[10] Ghosh, B.K., "Nonlinear Estimation Schemes for Visual Servoing", *IEEE Int Conf Robotics & Automation,* Workshop M-5, San Diego, 5/94.

[11] Hager, G.D., Toyama, K., "X vision: A Portable Substrate for Real-Time Vision Applications", Tech Report, Dept of Computer Science, Yale Univ, 1995.

[12] Hutchinson, S., Hager, G., Corke, P., "A Tutorial on Visual Servo Control", *IEEE Trans Robotics & Automation,,* V12 N5, 10/96.

[13] Koivo, A.J., "Real-time Vision Feedback for Servoing Robotic Manipulator with Self-Tuning Control", *IEEE Trans Syst Man Cybern* V21 N1, 1/91.

[14] Papanikolopoulous, N.P., Khosla, P.K., "Adaptive Robotic Visual Tracking: Theory and Experiments", *IEEE Trans Automatic Control"* V38 N3 3/93.

[15] Samson, C., Espiau, B., Le Borgne, M., "Robot Control: The Task Function Approach", Oxford University Press, 1990.

[16] Weiss, L.E., Sanderson, A.C., "Dynamic Sensor-Based Control of Robots with Visual Feedback", *IEEE J of Robotics & Automation* V RA-3 N5 p 404, 10/87.

[17] Wilson, W.J., "Visual Servo Control of Robots Using Kalman Filter Estimates of Robot Pose Relative to Work-pieces", in Visual Servoing, World Scientific, Koichi Hashimoto Editor, 1993