Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Feedforward Control for Human-in-the-Loop Camera Systems

Rares Stanciu and Paul Y. Oh
Drexel University, Philadelphia PA
Email: ris22@drexel.edu, paul@coe.drexel.edu

Abstract

Cameras are often mounted on platforms that can move like rovers, booms, gantries and aircraft. People operate such platforms to capture desired views of a scene or a target. To avoid collisions with the environment and occlusions, such platforms often possess redundant degrees-of-freedom. As a result, manual manipulating of such platforms demands much skill. Visual-servoing some degrees-of-freedom may reduce operator burden and improve tracking performance. This concept, which we call human-in-the-loop visual-servoing, is demonstrated in this paper and applies a $\alpha - \beta - \gamma$ filter and feedforward controller to a broadcast camera boom.

1 Introduction

Human-in-the-loop systems involve an operator who manipulates a device for desired tasks based on feedback from the device and environment. For example, devices like rovers, gantries, and aircraft possess a video camera where the task is to maneuver the vehicle and position the camera to obtain desired fields-of-view. Such tasks have applications in areas like broadcasting, inspection and exploration. Such device-mounted camera systems often possess many degrees of freedom (DOF) because it is important to capture as many fields-of-view as possible. To overcome joint limits, avoid collisions and ensure occlusion-free views, these devices are typically equipped with redundant DOF. Tracking moving subjects with such systems is a challenging task because it requires a well skilled operator who must manually coordinate multiple joints. Tracking performance becomes limited to how quickly the operator can manipulate redundant DOF. Figure 1 for example, shows a typical broadcast boom and pan-tilt camera head. Here, the operator can push and steer the dolly, as well as boom, pan and tilt the camera. Our particular interest is to apply *visual-servoing* to augment an operator's ability to track moving targets; computer vision is used to control some DOF so that the operator has fewer DOF to manipulate.

The prototype shown in Figure 1 was constructed to capture data, implement controllers and assess performance. Hardware includes a 266 MHz PC, a pan-tilt DC motor controller and quadrature encoders. The vehicle is a four wheeled dolly with gimbaled broadcast boom, a motorized pan-tilt head, color camera, wireless video transmitter and framegrabber. The boom pivots on the steerable dolly to sweep the camera horizontally and vertically. Both proportional [6] [7] and partitioned [4] [5] controllers were designed that visually servo the pan-tilt motors to keep a moving target centered in the camera's field-of-view despite boom or dolly motions. Sample image stills acquired from videotaping tracking experiments are shown in Figure 2. The net effect is what we call *human-in-the-loop visual servoing* – the operator just focuses on safely manipulating the boom and dolly while computer-control automatically servos the pan-tilt camera.

A challenge underlined in [6] was the system's stability, especially when the target and the boom move 180 degrees out of phase. If boom motion data is not included, camera pose cannot be determined explicitly because there are redundant degrees-of-freedom. As a result, the system could track a slow moving target rather well, but would be unstable when the target or boom moves quickly. In this paper a feedforward controller is employed to improve stability. Section 2 models the pan-tilt motors. The feedforward controller is presented in Section 3. Several experiments were performed to assess the performance of this controller. The results as well as some conclusions and a map of future work are presented in Sections 4 and 5 respectively.

2 Modeling the PTU

As shown in Figure 1, the camera is mounted on a 2 degree-of-freedom pan-tilt unit (PTU). Two DC mo-

0-7803-8232-3/04/$17.00 ©2004 IEEE

Figure 2: Three sequential images from videotaping a tracking experiment. Camera field-of-view (top row) shows the target is kept centered in the image plane despite boom motions executed by the operator. Such motions are illustrated by the middle and bottom rows' images; two cameras placed in the room were used to record the experiment.

tors are driven by a motion card installed in a PC. Like many commercial motion cards, the PID control gains are factory set, balancing transient response with minimal overshoot. Using a standard DC motor transfer function, one has

$$G_m(s) = \frac{\dot{\theta}_m(s)}{E_a(s)} = \frac{K_t}{K_v K_t + (sJ_m + D_m)(R_a + sL_a)} \quad (1)$$

where $\dot{\theta}_m$ is motor speed, E_a is the applied voltage, K_t is the motor torque constant, K_v is the back EMF constant, R_a and L_a are the rotor resistance and inductance respectively and D_m is the armature viscous damping. Values for these parameters are given in Table 1. J_m is the motor shaft's moment of inertia.

$$J_m = J_a + J_L(\frac{N_1}{N_2})^2 \quad (2)$$

where, J_L is load moment of inertia, J_a is the rotor moment of inertia and $\frac{N_1}{N_2}$ is the gear ratio. The PTU's gear ratio and D_m are both small and were set to zero. As such, Equation(1) with values from Table 1 results in

$$G_m(s) = \frac{\dot{\theta}_m(s)}{E_a(s)} = \frac{5500}{0.001862s^2 + 1.295s + 31.9} \quad (3)$$

Using a zero-order-hold to model a digital-to-analog converter, the discrete form of the transfer function

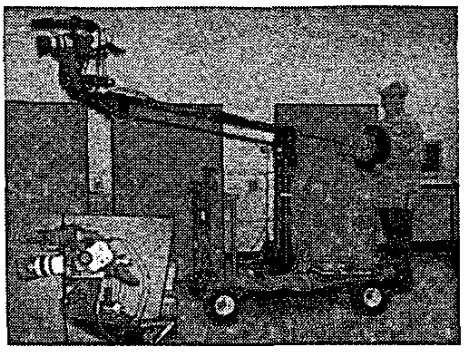

Figure 1: An operator can boom the arm horizontally and vertically to position the camera. The pan-tilt (lower left inset) head provides additional degrees-of-freedom.

Motor Parameters	Value and Units
R_a, rotor resistance	1.15Ω
L_a, rotor inductance	$1.4mH$
K_t, torque constant	$0.055Nm/A$
K_v, back EMF constant	$5.8V/krpm$
J_a rotor moment of inertia	$1.33 \cdot 10^{-5}kgm^2$

Table 1: The parameters of the PTU motors.

can be calculated. Figure 3 gives the plant block diagram that combines the motor and the PID controller $D(z)$. Here v_{ref} is the command reference velocity, E is the error between the command and actual motor velocities and $K_e = 2000$ counts/rev is the encoder constant. The sampling time T was set at 1.25 msec. $D(z)$ is the factory tuned PID controller with proportional, integral and derivative gains set at $K_P = 15000$, $K_I = 40$ and $K_D = 20000$ respectively for the PTU pan motor. PID gains for the tilt motor were factory set at $K_P = 15000$, $K_I = 20$ and $K_D = 32000$. With $G_m(s)$ given by Equation(3), the plant discrete transfer function $G_P(z)$ relating the command and actual velocities is given as

$$G_p(z) = \frac{0.704 - 0.787z^{-1} + 0.439z^{-2} - 0.055z^{-3} + 0.035z^{-4}}{1409.3 - 1575.36z^{-1} + 878z^{-2} - 11.02z^{-3} + 70z^{-4}} \quad (4)$$

Equation(4) is validated experimentally as described in Section 4.

3 Feedforward Controller

As mentioned in Section 1, the boom-camera system under proportional control [6] becomes unstable when tracking a fast moving target. The boom and PTU

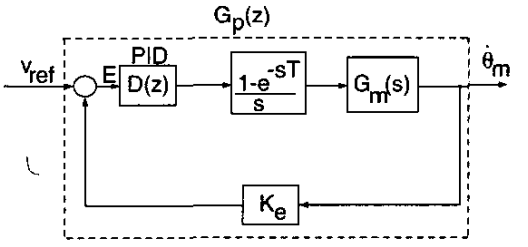

Figure 3: The PTU Controller Block Diagram

are redundant rotational DOF that at high frequencies can become 180 degrees out-of-phase. The net result is the boom and PTU rotations conflict rather than cooperate and tracking fails. To overcome such instabilities, a feedforward controller can be designed which provides target motion estimation [1]. Figure 5 depicts a block diagram with transfer function

$$\frac{{}^iX(z)}{X_t(z)} = \frac{V(z)(1 - G_p(z) \cdot D_F(z))}{1 + V(z) \cdot G_p(z) \cdot D(z)} \quad (5)$$

where ${}^iX(z)$ is the position of the target in the image, $X_t(z)$ is target position, $V(z)$ and $G_p(z)$, are respectively the transfer functions for the vision system and PTU. $D_F(z)$ and $D(z)$ are respectively the transfer functions for the feedforward and feedback controllers.

Clearly if $D_F(z) = G_p^{-1}(z)$ the tracking error will be zero, but this requires knowledge of the target position which is not directly measurable. Consequently the target position and velocity are estimated. For a horizontally translating target, its centroid in the image plane is given by the relative angle between the camera and the target

$${}^iX(z) = K_{lens}(X_t(z) - X_r(z)) \quad (6)$$

where ${}^iX(z)$ and $X_t(z)$ are the target position in the image plane and world frame respectively. $X_r(z)$ is the position of the point which is in camera's focus (due to the booming and camera rotation) and K_{lens} is the lens zoom value. The target position prediction can be obtained from the boom and PTU as seen in Figure 4. Rearranging this equation yields

$$\hat{X}_t(z) = \frac{{}^i\tilde{X}(z)}{K_{lens}} + X_r(z) \quad (7)$$

where $\hat{X}_t$ is predicted target position.

Predicting target velocity requires a tracking filter. Oftentimes a Kalman filter is used but is computationally expensive. Since Kalman gains often converge

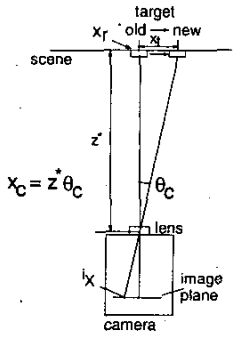

Figure 4: A schematic of camera-scene.

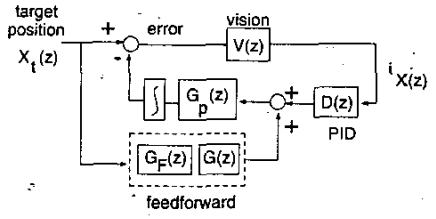

Figure 5: The Feedforward Controller with Feedback Compensation.

to constants, the simpler $\alpha - \beta - \gamma$ tracking filter can be employed which tracks both position and velocity without steady-state errors [3].

Tracking involves two step process. The first step is to predict target position and velocity

$$x_p(k+1) = x_s(k) + Tv_s(k) + T^2 a_s(k)/2 \quad (8)$$

$$v_p(k+1) = v_s(k) + Ta_s(k) \quad (9)$$

where T is the sample time and $x_p(k+1)$ and $v_p(k+1)$ are respectively the predictions for position and velocity at iteration $k+1$. $x_s(k)$, $v_s(k)$ and $a_s(s)$ are the corrected values of iteration k for position, velocity and acceleration respectively.

3.1 $\alpha - \beta - \gamma$ Filter

The second step is to make corrections

$$x_s(k) = x_p(k) + \alpha(x_o(k) - x_p(k)) \quad (10)$$

$$v_s(k) = v_p(k) + (\beta/T)(x_o(k) - x_p(k)) \quad (11)$$

$$a_s(k) = a_p(k-1) + (\gamma/2T^2)(x_o(k) - x_p(k)) \quad (12)$$

where $x_o(k)$ is the observed (sampled) position at iteration k. The appropriate selection of gains α, β and

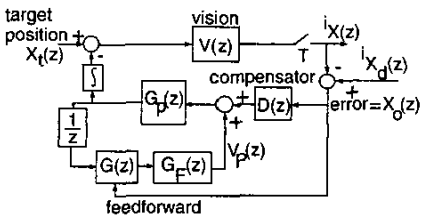

Figure 6: The Feedforward Controller with Feedback Compensation as it was implemented.

γ will determine the performance and stability of the filter [8].

An $\alpha - \beta - \gamma$ filter was implemented to predict target velocity in the image plane with gains set at $\alpha = 0.75$, $\beta = 0.8$ and $\gamma = 0.25$. This velocity was then used in the feedforward algorithm as shown in Figure 6. Image processing in the camera system can be modeled as a $1/z$ unit delay which affects camera position x_r, and estimates of target position. In Figure 6, the block $G_F(z)$ represents the transfer function of the $\alpha - \beta - \gamma$ filter, with the observed position as input and the predicted velocity as output. $X_d(z)$ represents target's desired position in the image plane and its value is 320 *pixels*.

The constant, K_{lens}, represents the ratio between the target dimensions in the image plane (in pixels) and in meters. K_{lens} was set a constant value and assumes a pinhole camera model that maps the image plane and world coordinates. This constant was experimentally determined by comparing known lengths in world coordinates to their projections in the camera's image plane.

Taking the Z transform of the $\alpha - \beta - \gamma$ filter yields its discrete-time transfer function $G_F(z) = \frac{V_P(z)}{X_o(z)}$ or

$$G_F(z) = \frac{2.4z^4 - 8.47z^3 + 11.09z^2 - 3.81z}{z^5 - 3.3875z^4 + 4.2375z^3 - 2.1632z^2 - 0.7325z + 0.75} \quad (13)$$

where $V_P(z)$ is the predicted velocity and $X_o(z)$ is target's observed position.

4 Experimental Results

Experiments to validate the dynamic models and to compare the performance of feedforward and proportional control in *human-in-the-loop visual-servoing* were performed. A condensation algorithm [2] was implemented to capture the target's position in image

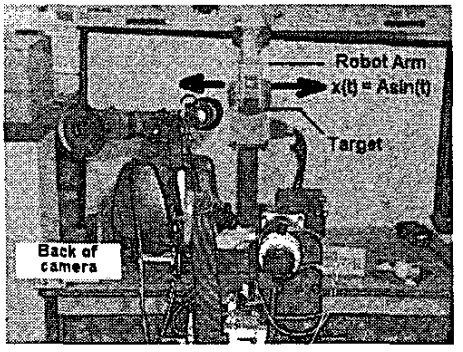

Figure 7: A wooden block target was mounted in the end-effector of a Mitsubishi robot arm (background). The boom-camera system (foreground) attempts to keep the target's image centered in the camera's field-of-view.

space. A $8.9 \times 8.25\ cm^2$ wooden block was mounted in the end-effector of a 7-DOF Mitsubishi robot arm, Figure 7. The camera-to-target distance was $3.15\ m$. The value of K_{lens} was set to $700\ pixels/m$. In every sampling period the target was succesfully identified. However, the condensation algorithm is a little bit noisy. To minimize this effect, the target image should be kept small. In this case the target dimensions were $34 \times 32\ pixels$. Using this value, from Figure 4 the focal length is then

$$f = \frac{{}^{i}d \cdot z^{*}}{D} \tag{14}$$

where ${}^{i}d = 34\ pixels$ is the target width in the image plane, z^{*} is the distance between the lens and the target and $D = 8.9\ cm$ is the target width. Then the focal length is $f = 1200\ pixels$.

To validate the dynamic model, Equation(4), a Bode plot was generated. Here, the input would be an oscillating target and the output would be the resulting PTU angle. As such, the robot arm oscillated the block horizontally over a range of frequencies and PTU output angles were recorded. As shown in Figure 8, the resulting magnitude and phase plots (top two) match well with a Matlab simulation on Equation 4 (bottom). Figure 9 shows the results tracking the target which oscillated at $0.08\ Hz$ from -0.58 m to $+0.49$ m (top plot). While the controllers attempted to track the target, the boom was manually moved over from -15 to $+25$ degrees (second plot from top). The bottom two plots depict tracking errors resulting from such *human-in-the-loop visual-servoing*. Feedforward based control has a ± 100 pixel peak-to-peak tracking error (bottom-most plot) compared to ± 300 pixel errors in proportional-only control. At 300 *pixels* with a camera-to-target distance at $3.15\ m$ and focal length of $f = 1200\ pixels$ the target was at the boundaries of the camera's field of view while at 100 pixels, the otained error was about $0.35\ m$. Zero peak-to-peak pixel error reflects perfect tracking such that the target image always remains centered in the camera's field-of-view. As such, the results suggest that a feedforward strategy performs better than proportional control for *human-in-the-loop visual-servoing*.

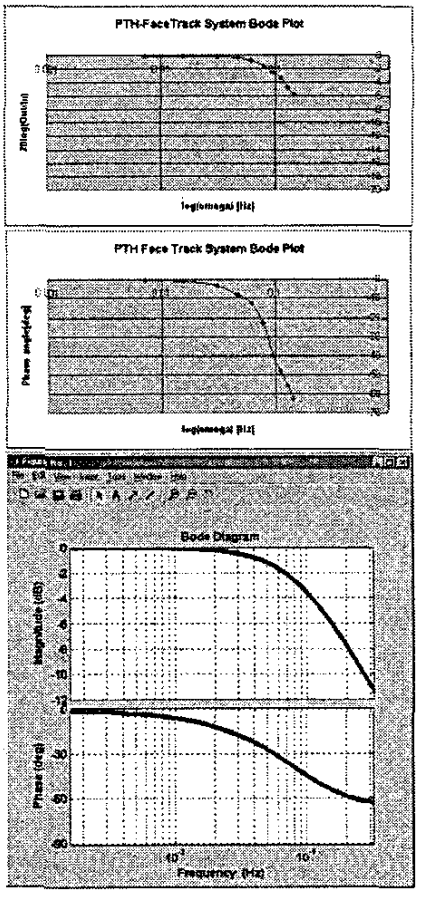

Figure 8: The PTU Bode magnitude (top) and phase (bottom) plots

5 Conclusions and Future Work

This paper integrates visual-servoing for augmenting the tracking performance of camera teleoperators. By reducing the number of DOF that need to be manually manipulated, the operator can concentrate on coarse

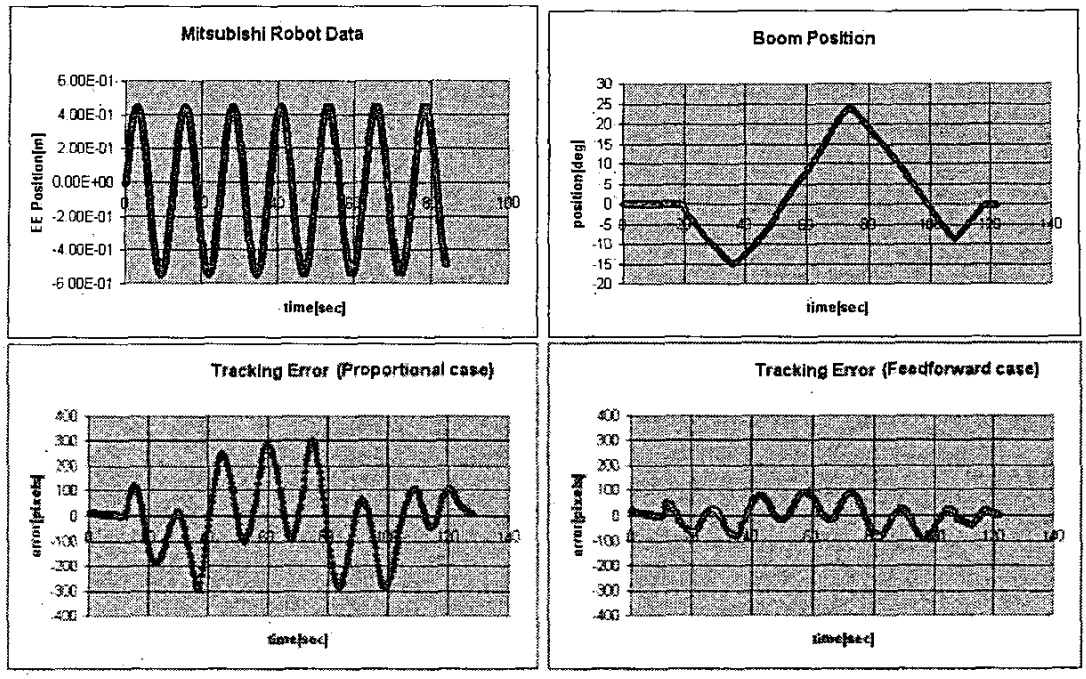

Figure 9: Tracking errors comparing feedforward and proportional control in *human-in-the-loop visual-servoing.*

camera motion. Using a broadcast boom system as an experimental platform, the dynamics of a camera pan-tilt-unit were derived and validated experimentally. A feedforward controller with an $\alpha - \beta - \gamma$ filter was formulated and implemented experimentally. Results comparing proportional and feedforward controllers were illustrated. Feedforward control yielded lower peak-to-peak pixel errors which suggest that estimating target position improves tracking performance despite human-in-the-loop disturbances. Future work will look at increasing the bandwidth under which the boom-camera system can track stably. A multivariable controller approach is being considered.

References

[1] Corke P.I., Good M.C."Dynamic Effects in Visual Closed-Loop Systems" *IEEE Trans on Robotics and Automation* V. 12 N. 5 Oct. 1996

[2] Isard, M., Blake, A., "CONDENSATION – Conditional Density Propagation for Visual Tracking", *Int. J. Computer Vision*, V29, N1, pp. 5-28, 1998.

[3] Kalata, P.R., Murphy, K.M., '$\alpha-\beta$ Target Tracking with Track Rate Variations", *Proc. of the Twenty-Ninth Southeastern Symposium on System Theory*, pp. 70-74, Mar. 1997.

[4] Oh, P.Y., Allen, P.K., "Visual Servoing by Partitioning Degrees of Freedom", *IEEE Trans. on Robotics Aut.* V17 N1, pp. 1-17, Feb. 2001.

[5] Oh, P.Y, "Biologically Inspired Visual-Servoing using a Macro/Micro Actuator Approach", *Int. Conference on Imaging Science, Systems and Tech. - CISST 2002*, Las Vegas CA, June 2002

[6] Stanciu, R., Oh, P.Y., "Designing Visually Servoed Tracking to Augment Camera Teleoperators" *IEEE Intelligent Robots and System (IROS)*, Lausanne, Switzerland, V1, pp. 342-347, 2002.

[7] Stanciu, R., Oh, P.Y., "Human-in-the-loop Visually Servoed Tracking" *Int. Conference on Computer, Communication and Control Tech. (CCCT)*, V5, pp. 318-323, Orlando, FL, July 2003.

[8] Tenne, D., Singh, T., "Optimal Design of $\alpha - \beta - (\gamma)$ Filters", *Proc. of the American Control Conference*, V6 pp. 4348-4352, June 2000.