Analysis of Partitioned Joint-coupling in Visually Servoed Tracking

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

Abstract

Humans have degrees-of-freedom (dof) of varying bandwidths. One can casually observe that they coordinate their joints while visually tracking and suggests that joint interplay effects tracking performance. In a control scheme we call partitioning, both image and kinematic data are used to visually-servo a 5-dof robot by defining a joint-coupling among rotational and translational dof. Performance analysis will illustrate how joint-coupling can be used to improve visual-servoed tracking. Simulations and experiments highlight results.

1 Introduction

People display interesting behaviors when visually tracking moving targets. First, we engage several joints (eye pan, neck, torso etc) when tracking and their motions suggest a kinematic joint coupling. For example, both eyes and neck typically pan in the same direction. Second, when tracking, the eyes lead (i.e. start panning before) the neck. A possible explanation for such behavior is that joints with faster response times serve as compensators for joints that are slower to react. With these behaviors, people track targets quite well despite large variations in target motions. This hints that integration of such behaviors into a visually servoed robot may improve its tracking performance.

Compensators are often added to the control law to makeup for performance deficiencies. For example, a common objective in eye-in-hand tracking is to robotically servo the camera such that it remains centered over the moving target. But, joint bandwidth and saturation limits not permitting, the camera will lag behind the target. The control engineer can then mathematically design feedforward [4] or lead compensators to add phase and cancel out camera lag. *Skeet shooting* is a good example of lead compensation. Often a marksmen will aim ahead of the target, taking its flight path, speed and distance into account. He thereby physically adds phase to the target's bearing angle. Beyond the mathematical abstractions of compensator design, joints with fast bandwidths can physically act as lead compensators when their kinematic encoder data is added to the visual-servoing loop. This will be illustrated in this paper.

Our particular vision interests are in visually monitoring a large $3.6 \times 6.4 \times 1$ m^3 assembly workcell with a 5-dof robot (Fig. 1). This robot is a custom-built 3-dof Cartesian gantry mounted with a Directed Perceptions 2-dof pan-tilt-unit (PTU) and camera on its end-effector. For our initial tracking efforts, a pose regulator [7], [11], [2] was designed. The control law in such designs rely exclusively on image data to update camera motions using an image Jacobian to maintain a desired camera-to-target pose. This initial effort met with several limitations associated with joint bandwidth and saturation. Target translations forced gantry motions which are marked by large inertial loads and slow responses. Oftentimes the target would accelerate past the camera's field-of-view (FOV) before the gantry got up to speed thus losing visual contact. By contrast, the small, lightweight PTU responds quickly and easily. Another problem was abrupt target starts and stops forced large gantry accelerations that generated camera end-point vibrations. Kalman filtering [12], [1] was implemented to improve robustness but only works when the target's motion dynamics can be specified *a priori*.

To overcome these limitations a joint-coupled system called *partitioning* was designed [9], [10]. In the control law both image and kinematic joint data are used. The pan and tilt degrees-of-freedom (dof) are visually servoed using image data while gantry translations are kinematically servoed using pan and tilt angle data. The purpose of this paper is to present an analysis of partitioning to illustrate joint interplay effects in visually-servoed tracking. It was found that fast bandwidths joints physically display lead compensation characteristics much like the skeet shooting anal-

*This work was supported in part by an ONR/DARPA MURI award ONR N00014-95-1-0601 and NSF grants CDA-96-25374.

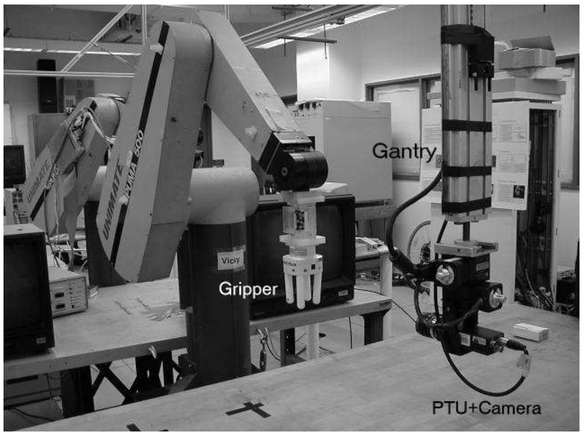

Figure 1: Pan-tilt unit and camera mounted on XYZ gantry end-effector tracks targets like grippers in workcell

ogy. This finding is highlighted by comparing three designs, namely a pan-only (section 2). gantry-only (section 3) and coupled pan-gantry (section 4) tracking systems. Their comparison is done by analyzing (root locus and Bode plots) their linearized models and highlighted with simulations and experiments in time and frequency-domain.

2 Pan-only Tracking

To keep a horizontally translating target centered in the camera's FOV one dof is required, hence a visually-servoed pan-only system was designed. Automatic tracking behavior can be achieved by controlling pan velocity in proportional to target position differences. This can be modeled mathematically with a block diagram (Fig. 2) and linearization (Fig. 3). Tracking in this manner is known as *steering* [4] or *piloting* [3].

Target translation x_t results in a bearing angle θ_t with respect to the camera. If the target is not centered in the camera's image plane, then a non-zero angle difference $\tilde{\theta}$ exists between θ_t and camera angle θ_c. Assuming small angles, $\tilde{\theta}$ can be approximated by

$$\tilde{\theta} = \theta_t - \theta_c \approx \frac{x_t}{z^* + L} - \frac{x_c^{'}}{z^* + L} \qquad (1)$$

z^* and L are the initial camera-to-target distance and camera's radius of rotation respectively. ϕ is the angle the target makes with respect to the lens center and using perspective

$$\phi \approx \frac{x_t - x_c^{'}}{z^*} \equiv \frac{u}{f} \qquad (2)$$

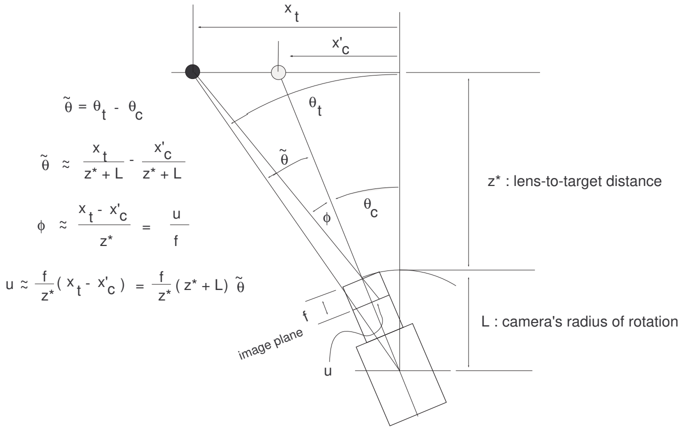

Figure 3: Relevant notation used for linearization

where u is the target's horizontal pixel coordinate in the camera's image plane and f is the lens focal length (633 pixels). Combining these two equations yields

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

with the following input-output relationships

$$\frac{x_t}{\theta_t} = \frac{1}{z^* + L} \quad \text{and} \quad \frac{\tilde{\theta}}{u} = \frac{f(z^* + L)}{z^*} \qquad (4)$$

The $\frac{f(z^*+L)}{z^*}$ term is a camera-to-target distance dependent gain and models a camera's lens (with focal length f). u and v (vertical pixel) define the 2-D target position $\vec{s} = (u, v)$ in the camera's image plane.

Fig. 2 shows the two input-output relationships as block elements before and after the $\tilde{\theta}$ summer. The frame-grabber digitizes the analog image position s and generates its sample $s[k]$ in pixels. $s^* = s^*(u^*, v^*)$ is a set point image location. Before tracking begins camera pose is initialized so that its image plane is centered (i.e. $s^* = (0, 0)$) over the target. As the target moves, the current image position $s[k]$ is compared to the set point s^*. The difference is the target's position relative to the image center. Since the target only translates horizontally, perspective yields

$$\dot{s} = \dot{u} = -\frac{1}{f}\omega_y \qquad (5)$$

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

$$e[k] = -\frac{1}{f}\left(s[k]\right) \qquad (6)$$

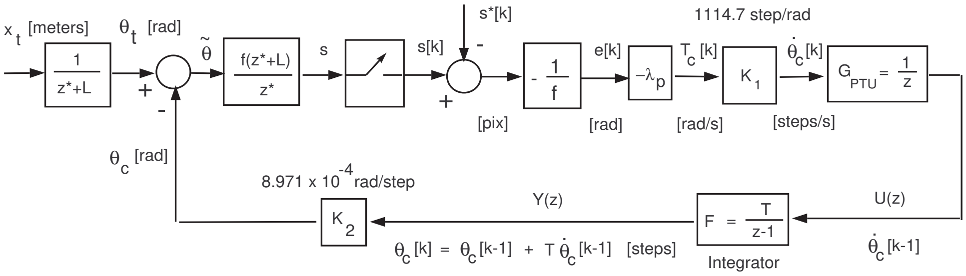

Figure 2: Block diagram for steering design

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) with λ_g and K_1 as the gain and unit conversion constants respectively. Within a stepper's start/stop (pull-in) velocity range, it accelerates instantaneously and modeled, G_{PTU}, as a unit delay. Forward-rectangular integration at sample time T [sec]

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

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 pan-only closed-loop linearized transfer function is

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

Fig. 4 is a simulated step response (solid) using (8) and $\lambda_p = 1.0$ and compares well to actual experimental results (dots). Unless specified differently, $z^* = 0.25$ m, $L = 0.1$ m, $f = 633$ pix for all simulations and experiments. Experiments measured $T = 0.12$ s sampling time and used in (8) to generate the simulation. This long sampling time is due serial communications latency. The PTU's 68HC11 embedded micro sends/receives ASCII characters at 9600 baud and is serially connected to our Unix-based Sparc 20 host computer. Unix context switching adds latency and a constitutes the bulk of the program loop time. To maintain fast image capture, Unix's multitasking capabilities were exploited; the vision-handling process (an XVision [8] 40×40 sum-of-squared-difference (SSD) tracker oversampling at 3.5 ms loop time) was forked. It sends target pixel coordinates to the pan servoing process when requested, via an interprocess pipe. The net effect is that the long serial latency does not hinder image tracking but its value limits how quickly new PTU velocities can be issued.

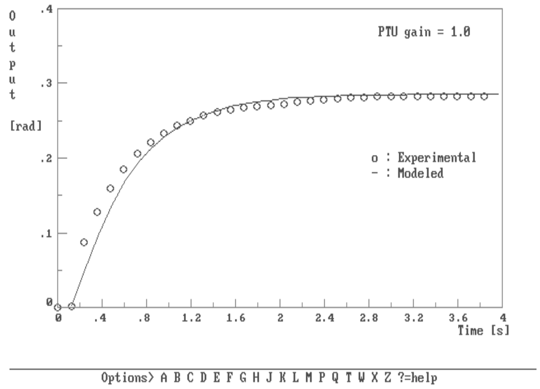

Figure 4: Camera pan step-input ($x_t = 0.1$ m) response θ_c for $\lambda_p = 1.0$ results in a 0.28 rad bearing angle

2.1 Pan-only Root Locus

Pan-only tracking performance depends on the proportional gain λ_p value. Increasing λ_p moves the poles and zeros of the closed-loop system (8) and results in faster response time but possible overshoot. To visualize gain effects, a root locus plot is used. Fig. 5 is such a plot where (8) is re-expressed into root locus form (9). The system has poles at $z = 0$ and 1. For $\lambda_p \geq 1.49$, the loci breaks away vertically at $z = 0.5$ and the pan will have a fast rise time but overshoot. Instability (loci lies outside the unit circle) sets in for $\lambda_p \geq 5.95$. $\lambda_p = 2.0$ is a good compromise between fast response time and overshoot and yields a $\zeta = 0.707$ damping ratio. This places the poles slightly above and below the real z-axis.

$$G_{\theta_{RL}} = \frac{T(z^* + L)}{z^* z(z-1)} \tag{9}$$

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

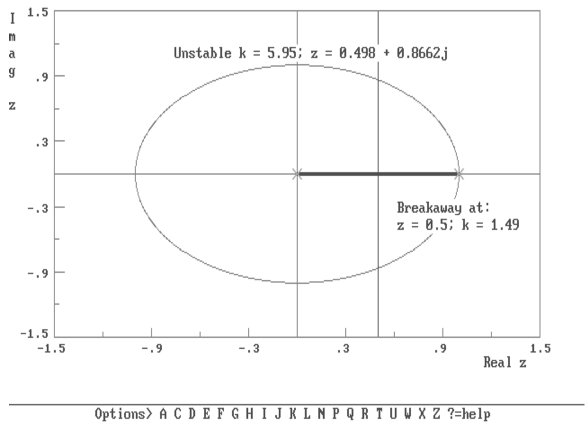

Figure 5: Root locus $G_{\theta_{RL}} = \frac{T(z^*+L)}{z^*z(z-1)} = \frac{0.168}{z(z-1)}$.

2.2 Pan-only Frequency Response

Root locus plots in themselves don't clearly present lag behavior. (8) is known in control theory as a Type I (a single $(z-1)$ term in the denominator) system and have the characteristic that constant x_t step inputs will yield zero steady-state error as seen in Fig. 4. Type I system's yield errors to sinusoidal inputs and result in the camera lagging behind the target. The amount of lag can be determined using a Bode plot.

A Puma-mounted target was translated with a sinusoidal acceleration and tracked. Both the target position, x_t [m], and resulting pan response, θ_c [rad] were recorded. Twenty different trials were performed over a 0.02 to 1.0 Hz range with $\lambda_p \doteq 1.0$. Fig. 6 shows the resulting Bode magnitude and phase plots (dotted values). A Matlab Bode plot of linearized model (8) shows (solid) close correspondence to the ones obtained experimentally.

The 3 dB down point in Fig. 6 occurs at ≈ 0.3 Hz (1.9 rad/s) with a phase angle of $\approx -70^o$ (-1.2 rad). At $\lambda_p = 1.0$ a root locus reveals a dominant pole (one closest to $z = 1$) occurring at $z = 0.7822$ on the real z-axis. Since $z = e^{sT}$ and $T = 0.12$ s, $s = -2.05$ or a natural frequency of $w_{\mathrm{n}} = 2.05$ rad/s and closely matches the ≈ 1.9 rad/sec seen in the Bode magnitude plot. Thus, for target frequencies near 0.3 Hz, we can expect camera lag of $\approx 70^o$. The time response in Fig. 7 shows this lag (solid) quite clearly.

3 Gantry-only Tracking

A second gantry-only visually-servoed tracking system was designed using only one gantry dof. As mentioned in the introduction, this dof is marked by large inertial loads and one can expect more camera lag than with a pan-only system. Steering was also implemented on this gantry-only system to track a target traveling horizontally and parallel to the camera's image plane. Fig. 8 is the block diagram and (10) is its closed-loop transfer function. Notationally it is similar to the pan-only system. Perspective $u = fx_t/z^*$ yields a linear relationship and no bearing angle is necessary.

$$G_g(z) = \frac{x_c}{x_t} = \frac{T\lambda_g}{z(z-1) + T\lambda_g} \tag{10}$$

A 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* every 30 ms. The host Sparc 20 issues velocity commands (in ASCII) to the PC via a 38400 baud serial line. With low-level PC programming the serial latency was about 6 ms and thus the sampling time T was set at 40 ms.

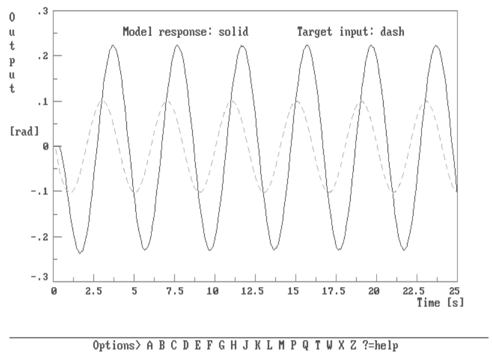

Figure 7: Modeled θ_c response (solid) to 0.25 Hz accelerating x_t translation (dash). The lag between the two curves is $\approx 70^o$

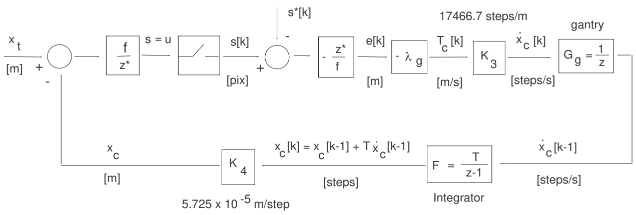

Figure 8: Detailed block diagram for gantry-only tracking

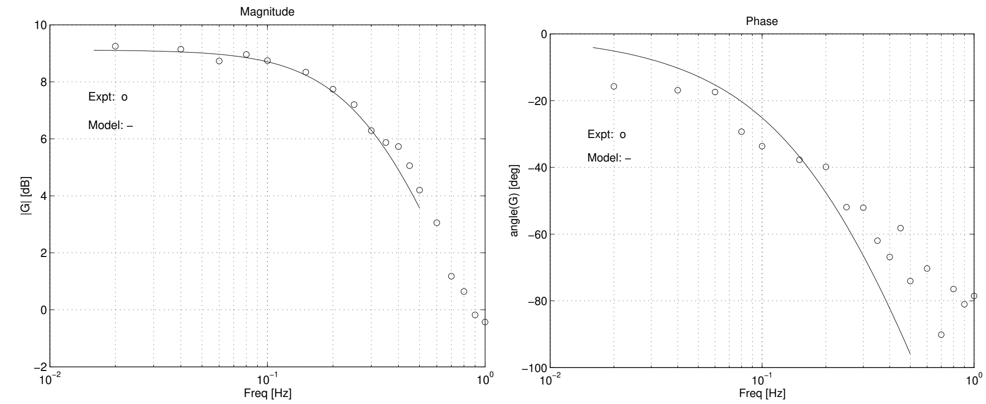

Figure 6: (Left) Pan-only Bode magnitude (Right) and phase plots.

The root locus and Bode plots of (10) are graphically similar to the pan-only system and not shown. It also has poles at $z = 0$ and 1 and breaks away at $z = 0.5$ ($\lambda_g = 6.25$). The gantry's large inertial load, reduces its stepper start/stop (pull-in) range and hence maximum gain (saturation limits). Experimentally the gantry would stall when instantaneously accelerating from stand-still at $\lambda_g = 0.85$. The gantry-only Bode plot revealed a -3 dB value at 0.1668 Hz and -46.76^o phase angle. As expected, the gantry's slow response yields lag at much lower frequency, having about half the PTU's bandwidth.

4 Pan-Gantry Coupled Tracking

This paper began by highlighting certain human behaviors while tracking that suggests kinematic coupling between their various joints. Partitioning integrates a joint-coupling into the control law using both image and kinematic encoder data. In this third tracking design, pan is visually-servoed and its angle data is use to kinematically servo gantry translations. Fig. 9 is its block diagram and is structurally similar to a combination of Figs. 2 and 8. Here, camera pan velocity $\dot{\theta}_c$, is visually servoed using pixel difference $s[k] - s^*$ and gain λ_p and the gantry translates using camera pan angle difference $\theta_c^r - \theta^*$ and gain λ_g. θ^* is a desired setpoint camera orientation. A human analogy of it would be a desired eye orientation, which look forward. Without loss of generality, θ^* is set to zero.

Target translation input x_t [m] yields the following closed-loop transfer functions:

$$ {}^cG_\theta(z) = \frac{\theta_c \text{ [rad]}}{x_t \text{ [m]}} = \frac{\lambda_p T z^2 - \lambda_p T z}{\text{CE}} \qquad (11) $$

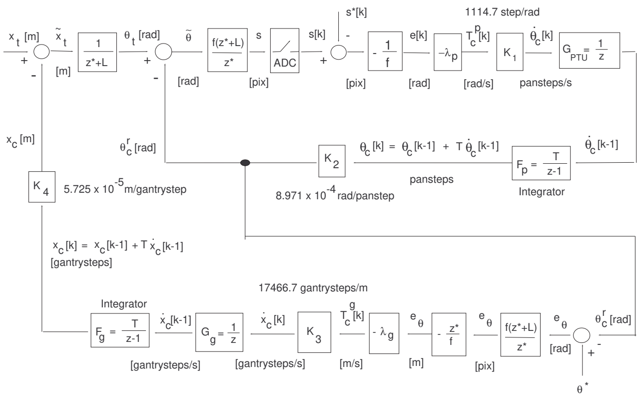

Figure 9: Coupled pan-gantry block diagram. Notice the similarity with figures 2 and 8.

$$ {}^cG_g(z) = \frac{x_c \text{ [m]}}{x_t \text{ [m]}} = \frac{(z^* + L)T^2 \lambda_g \lambda_p}{\text{CE}} \qquad (12) $$

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

Superscript c in ${}^cG_\theta$ and ${}^cG_g$ denotes "coupled" and subscripts θ and g describe pan and gantry respectively. The net effect of the coupling in Fig. 9 is that tracking behavior (i.e pan and gantry servo action) depend on two gains λ_p and λ_g. This is because both ${}^cG_\theta$ and ${}^cG_g$ have the same characteristic equation (CE) and the closed-loop pole locations depend on the product $\lambda_p \lambda_g$. Tracking behavior can also be seen by the fact that if $\lambda_g = 0$ then there is no camera translation (since ${}^cG\ _g = 0$) and all tracking is done exclusively by panning. The gain dependence on pan

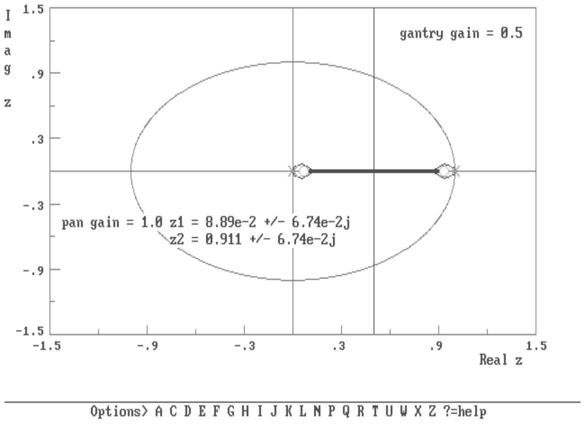

Figure 10: $G_{\lambda_g}^{RL}$ root locus with $\lambda_p = 1.0$

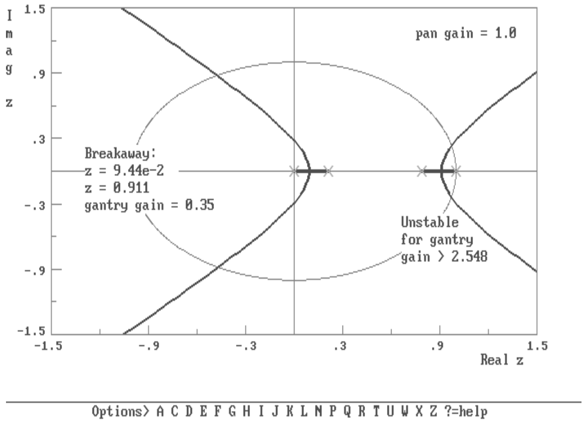

Figure 11: $G_{\lambda_p}^{RL}$ root locus with $\lambda_g = 0.5$

and gantry responses can be seen more clearly in the ensuing root locus diagrams and step responses.

4.1 Root Locus & Time Response

Reexpressing (11) and (12) into root locus forms yields two plots (Figs. 10 and 11), for varying λ_p and fixed $\lambda_g = 0.5$ (left) and varying λ_g and fixed $\lambda_p = 1.0$ (right). T was set aat 0.12 s in light of the PTU's serial latency. New gantry velocity updates are issued at this loop time and its serial latency (6 ms) is comparatively small. These two plots complement each other. The left reveals double poles at both $z = 0$ and 1, and a pair of zeros at 0.0641 and 0.936. The loci breaks away at $z = 0.5$ and pan response will be more oscillatory as λ_p increases above 1.958 with instability at $\lambda_p > 6.242$. The right reveals yields poles at $z = 0$, 1, 0.786 and 0.214 and no zeros. Gantry overshoot will occur for $\lambda_g > 0.35$ and settling time will increase. As λ_g is increased, oscillations will increase with unstable response occurring for $\lambda_g > 2.548$.

The net effect of kinematic coupling pan and gantry dof together is graphically shown in Fig. 12. Fig. 13 is the experimentally obtained (dash) step input responses for θ_c and x_c with $\lambda_p = 1.0, \lambda_g = 0.5$. Simulation results (solid) using equations (11) and (12) match these results closely.

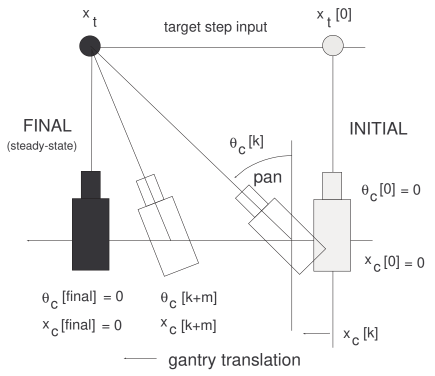

Figure 12: Graphical illustration of the coupled pan-gantry step response.

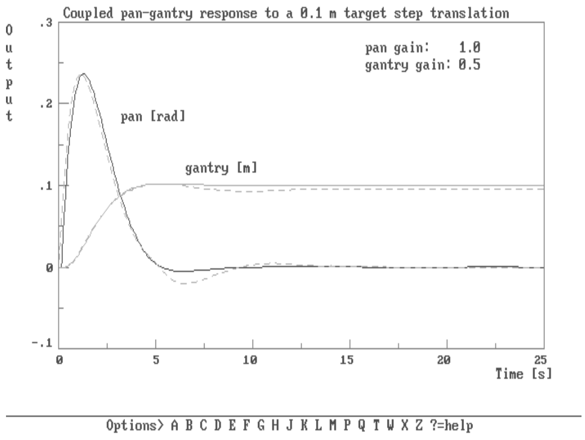

Figure 13: Coupled pan-gantry step response. Experimental (dash) and simulations (solid) shown. $\lambda_p = 1.0$ and $\lambda_g = 0.5$. Pan accelerations result in gantry translations.

5 Coupling Effects

Rise time is an often used performance metric and is defined as the time it takes the system to go from 10 to 90 percent of its steady-state value. The rise times of the pan-only, gantry-only and coupled-gantry systems are given in table 1:

System	rise time (s)
Pan-only	1.13
Gantry-only	4.29
Coupled pan-gantry	2.50

Table 1: Rise time characteristics: $\lambda_p = 1$, $\lambda_g = 0.5$

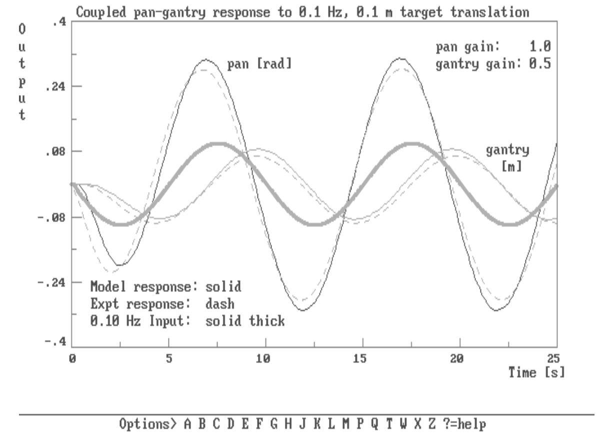

Figure 15: Coupled pan-gantry response to a 0.1 Hz, 0.1 m amplitude sinusoidal target translation x_t (thick solid line). Both experimental (dash) and simulated (solid) responses shown. $\lambda_p = 1.0$ and $\lambda_g = 0.5$.

The gantry-only system has very sluggish response mainly because of large inertial loads force using a low gain. Partitioning introduces coupling and takes advantage of the pan's fast rise time to translate the gantry quickly. As mentioned in the introduction, the pan dof serves as a lead compensator and is analogous to skeet shooting. The pan adds positive phase when aiming at the target. This can be seen analytically from the Bode phase and magnitude plots of (11) and (12) given in Fig. 14.

The pan's Bode plot reveals 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, filtering high-frequency signals. At low frequencies (e.g. 0.02 Hz) there is little pan and larger gantry motion. In other words most of the tracking action is done by the gantry. At higher frequencies (e.g. 0.80 Hz) the reverse is true with pan motions being more prevalent. In mid-frequencies, both pan and gantry motions contribute towards tracking. The net effect is that the pan dof introduces phase lead to compensate for the lower gantry bandwidth. Fig. 15 shows this pan phase lead ($+55.88^o$ and 6.37 dB gain) for a 0.1 Hz sinusoidal input.

System	Peak error [pix]
Gantry-only $\lambda_g = 1.0$	133 (5.3 cm)
Coupled pan-gantry $\lambda_g = 1.0$, $\lambda_p = 1.0$	73 (2.9 cm)
Coupled pan-gantry $\lambda_g = 1.0$, $\lambda_p = 2.0$	33 (1.3 cm)

Table 2: Peak pixel errors

Peak-to-peak pixel error in response to a sinusoidal input is another performance metric. A small pixel error means that the target is centered in the camera's image plane. Fig. 16 illustrates the improved phase and peak-to-peak pixel error that joint-coupling offers. By increasing λ_p, even smaller pixel errors are possible (Fig. 17). Pixel error results are summarized in Table 2.

6 Conclusion

Casual observance reveals that humans coordinate their dof while tracking and suggests a joint-coupling. The partitioned scheme defines a joint-coupling in the control law that takes advantage of both image and kinematic data. Three visually-servoed tracking systems, pan-only, gantry-only and coupled pan-gantry (partitioned) were modeled and designed to compare their tracking performance (response time, lag and pixel errors) through simulations and experiments in time and frequency domains. The caparisons reveal that joint-coupling provides faster response and overcomes joint bandwidth and saturation limitations. Like skeet shooting, the partitioned pan dof adds phase to the gantry's dynamics due to its faster response and physically acts like a lead compensator. This suggests that lag performance can be improved by taking advantage of the joints in a robot that have fast bandwidth by kinematically coupling them to slower joints under a partitioned scheme. Coupled pan-gantry performance depends on two gains which can be tuned using root locus techniques. We are currently exploring linear quadratic cost functions to tune these gains. This will optimally prescribe the amount of pan versus gantry control efforts while tracking ensuring both stability and maximized performance.

References

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

[2] Chaumette, F., Rives, P., Espiau, B., "Positioning of a Robot with respect to an Object, Tracking it and

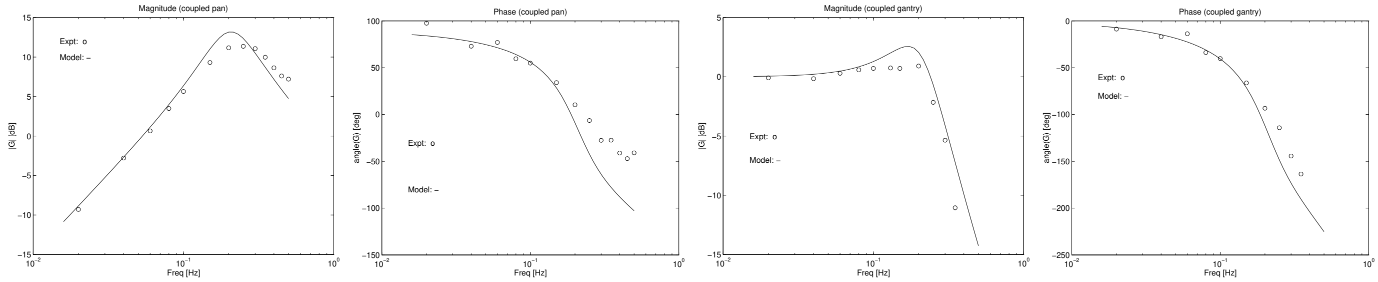

Figure 14: Bode plots of coupled pan-gantry (left to right): pan magnitude, pan phase, gantry magnitude and gantry phase for $\lambda_p = \lambda_g = 1.0$. Both Matlab-generated (solid) curves and experimental (dot) data shown. The pan's magnitude peaks at 0.189 Hz with a 12.95 dB gain and 0^o phase angle. The gantry 3 dB down occurs at 0.293 Hz with a -166^o phase angle.

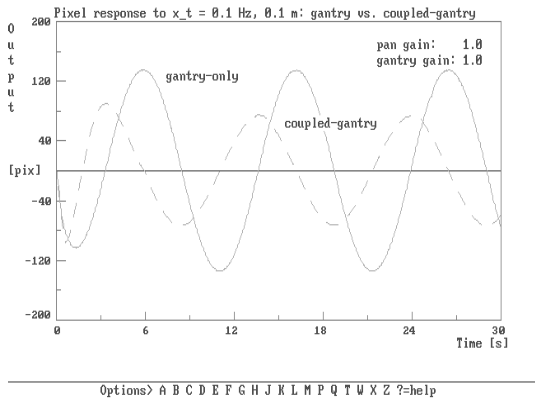

Figure 16: Pixel errors for both gantry-only ($\lambda_g = 1.0$) and coupled pan-gantry ($\lambda_g = \lambda_p = 1.0$) systems to a 0.1 Hz, 0.1 m amplitude target translation x_t. Note how coupling improves phase and has less peak-to-peak pixel error.

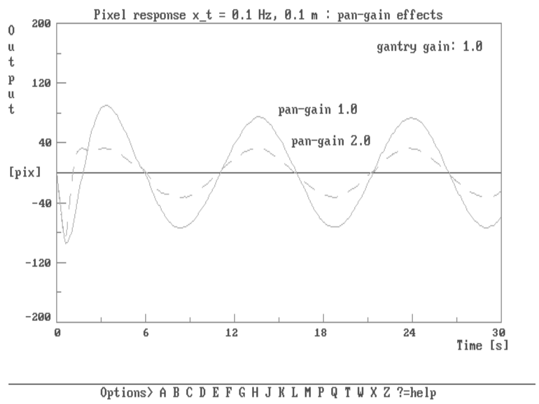

Figure 17: Coupled pan-gantry system pixel error response. As the pan gain λ_p increases, the pixel error decreases but can yield gantry end-point vibrations due to larger accelerations

Estimating its Velocity by Visual Servoing", *IEEE Int Conf Robotics & Automation,*, p 2248, 1991.

[3] Chaumette, F., Santos, A., "Tracking a Moving Object by Visual Servoing", *Proc. of 12th World Congress IFAC*, Vol 9 pp 409-414, Sydney, July 1993.

[4] Corke, P. I., "Dynamics of Visual Control", in *IEEE Int Conf Robotics & Automation*, Workshop M-5, San Diego, 5/94.

[5] Corke, P., "Design, Delay and Performance in Gaze Control: Engineering and Biological Approaches", in *The Confluence of Vision and Control*, Kriegman et al, Eds. Springer-Verlag pp. 146-158, 1998.

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

[7] 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.

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

[9] Oh, P.Y., Allen, P.K., "Design of a Partitioned Visual Feedback Controller", *IEEE Int Conf Robotics & Automation,* Leuven, Belgium, pp 1360 5/98.

[10] Oh, P.Y., Allen, P.K., "Performance of a Partitioned Visual Feedback Controller", *IEEE Int Conf Robotics & Automation,* Detroit, Michigan 5/99.

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

[12] 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