Slaving head and eye movements for visual telepresence

Jason J Heuring and David W Murray
Department of Engineering Science, University of Oxford
Parks Road, Oxford, OX1 3PJ, UK
Email [jace,dwm]@robots.ox.ac.uk

Abstract

This paper describes a system for visual telepresence that copies the motion of an operator's head onto the neck axes of an electromechanical stereo head/eye platform. Three or more point features on the operator's head are detected using an external static camera and tracked over time to recover the pose of the operator's head The process of capturing images, tracking features, recovering pose, and generating demands runs at 50Hz with a latency of 23ms, with the controller and platform typically taking another 20ms to satisfy a demand. Experiments show that the system, including head platform, is able to copy head movements of up to $400^{\circ}s^{-1}$. The paper goes on to consider the possibility of copying individual eye movements onto the individual camera axes of the head, but simulation studies suggest that delays in the feedback will lead to unsmooth eye response.

1 Introduction

There is a commercial demand for delicate, non-routine and risky tasks requiring a high degree of sensori-motor coordination to be undertaken in environments which are hostile and remote. Until robotic sensing and perception develop sufficient maturity to enable unsupervised performance of such tasks, teleoperation appears to be a way both of satisfying the demand, and also of learning what will be required for fully autonomous operation.

In this paper we first describe a system for slaving an operator's head motions onto a remote electro-mechanical stereo camera platform from which images are returned to the operator. A variety of methods have been proposed for head tracking in the literature including mechanical [6]; magnetic [8]; visual inside-out [1, 9], where a camera attached to the operator's head views the static environment; and visual outside-in [2, 3], where a static camera views the operator. Here we use the last method. Although inherently somewhat more difficult than the others, this method appears to be the only one which is completely non-intrusive and, more importantly, has scope for considerable extension — to include, for example, the recognition of facial and limb gestures.

We then explore the possibility of slaving individual eye movements onto the individual cameras, finding in simulation that delays would cause undesirable saccadic movements in the operator eyes.

2 Recovery of pose

The estimation of the pose of the master human head is based on the recovery of at least three image point features corresponding to 3D points on the operator's head whose relative positions are known. As long as three or more points remain visible, any motion of the operator's head can be recovered. The details of the algorithm are given in [5].

Consider the rotation $\mathtt{R}$ and translation $\mathbf{t}$ that transform a 3D point on the operator's head at $\mathbf{p}_i$, referred to in a local object-based frame, into a position $\mathbf{X}_i$ in

BMVC 1996 doi:10.5244/C.10.22

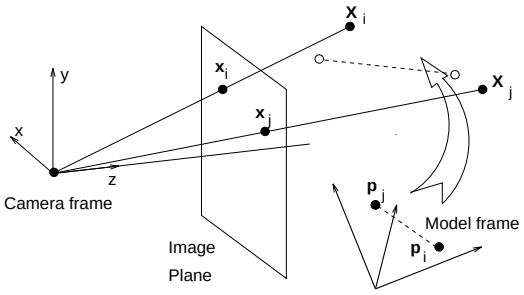

Figure 1: Points $\mathbf{p}$ are transformed to $\mathbf{X}$ in the camera frame and imaged at $\mathbf{x}$. Our constraint on rotation demands only that the rotated vector $\mathtt{R}(\mathbf{p}_i - \mathbf{p}_j)$ lies in the plane formed by the optic centre, $\mathbf{x}_i$ and $\mathbf{x}_j$.

the camera frame. The point in the camera frame must lie along the backprojected ray from the image position $\mathbf{x}_i$, so that

$$\mathtt{R}\mathbf{p}_i + \mathbf{t} = \zeta_i \mathbf{x}_i,$$

where ζ_i is unknown. It might be thought that the translation can be eliminated by using pairs of points

$$\mathtt{R}(\mathbf{p}_i - \mathbf{p}_j) = \zeta_i \mathbf{x}_i - \zeta_j \mathbf{x}_j .$$

where $|\mathbf{p}_i - \mathbf{p}_j| = |\zeta_i \mathbf{x}_i - \zeta_j \mathbf{x}_j|$, but this leaves translation encoded in the ζ. Instead we generate a weaker but more robust constraint by confining $\mathtt{R}(\mathbf{p}_i - \mathbf{p}_j)$ merely to lie *in the plane* containing $\mathbf{x}_i$, $\mathbf{x}_j$ and the optic center, as illustrated by the unfilled points in Figure 1. The normal to this plane is $(\mathbf{x}_i \times \mathbf{x}_j)$ and since it passes through the origin, $(\mathbf{x}_i \times \mathbf{x}_j) \cdot \mathbf{r} = 0$ for any vector $\mathbf{r}$ in the plane. Thus for n points visible from the model the constraints can be written as

$$(\mathbf{x}_i \times \mathbf{x}_j) \cdot \mathtt{R}(\mathbf{p}_i - \mathbf{p}_j) = 0, \text{ for} i = 1 \,..\, n-1, j = i+1 \,..\, n \quad .$$

To find the optimal rotation we minimize

$$\min_{\mathtt{R}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} ((\mathbf{x}_i \times \mathbf{x}_j) \cdot \mathtt{R}(\mathbf{p}_i - \mathbf{p}_j))^2.$$

The rotation is represented as a quaternion $\mathring{\mathbf{q}} = (q_0, q_1, q_2, q_3)^\top = (q_0, \mathbf{q}^\top)^\top$ and a linear approximation to the rotation matrix derived as

$$\mathtt{R} \approx \begin{bmatrix} q_0 & -2q_3 & 2q_2 \\ 2q_3 & q_0 & -2q_1 \\ -2q_2 & 2q_1 & q_0 \end{bmatrix}$$

The approximation improves on that of Harris [4], who set $R_{ii} = 1$. Applying this rotation, a structure point $\mathbf{p}$ after rotation becomes

$$\mathbf{p}' \approx q_0 \mathbf{p} + 2\mathbf{q} \times \mathbf{p},$$

and the minimization is therefore

$$\min_{\mathring{\mathbf{q}}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} ((\mathbf{x}_i \times \mathbf{x}_j) \cdot (q_0(\mathbf{p}_i - \mathbf{p}_j) + 2\mathbf{q} \times (\mathbf{p}_i - \mathbf{p}_j)))^2.$$

This in turn can be rewritten [5] as

$$\min_{\mathring{\mathbf{q}}} \mathring{\mathbf{q}}^\top \mathbb{N} \mathring{\mathbf{q}} \quad ,$$

minimized when $\mathring{\mathbf{q}}$ is the unit eigenvector of $\mathbb{N}$ corresponding to the smallest eigenvalue.

Although the approximation to the rotation matrix may not be a pure rotation, the quaternion recovered using the eigenvalue method is guaranteed to represent an orthogonal transformation, a requirement which is difficult to maintain this property when dealing with 3×3 rotation matrices. Even using quaternions, when the model structure is rotated thousands of times, distortion can occur due to the finite precision of computer arithmetic. We avoid this problem by obtaining the updated structure directly from the original structure at each step. At step n, the quaternion defining the rotation of the original structure, $\mathring{\mathbf{q}}_n$, and the structure points, $\mathbf{p}_n^i$, are updated as follows:

$$\mathring{\mathbf{q}}_n = \mathring{\mathbf{q}}_\theta \mathring{\mathbf{q}}_{n-1}; \quad \text{and} \quad \mathbf{p}_n^i = \mathring{\mathbf{q}}_n \mathbf{p}_{orig}^i \mathring{\mathbf{q}}^*{}_n$$

where $\mathring{\mathbf{q}}_\theta$ is the incremental rotation from the last computed structure position, and $\mathbf{p}_{orig}^i$ are the original structure points.

Once the rotation is known, the translation can easily be found. The equations for perspective projection for our structure can be written as

$$f(p_x^i + t_x)(p_z^i + t_z)^{-1} = u^i \quad \text{and} \quad f(p_y^i + t_y)(p_z^i + t_z)^{-1} = v^i.$$

From these comes the matrix equation

$$\begin{bmatrix} & \vdots & \\ f & 0 & -u^i \\ 0 & f & -v^i \\ & \vdots & \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} \vdots \\ u^1 p_z^i - f p_x^i \\ v^1 p_z^i - f p_y^i \\ \vdots \end{bmatrix}$$

and the least squares solution $(t_x, t_y, t_z)^\top$ can be found using a singular value decomposition.

3 System Architecture and Implementation

The overall layout of the system in shown in Figure 2(a). A single camera views the master head, and image fields are captured into the memory space of a network of C40 DSPs. The 2D image positions of features on the master head are detected and tracked, and the pose (both orientation and translation) is recovered. The present implementation uses a small number of point features and all processing is carried out at 50Hz on a single C40.

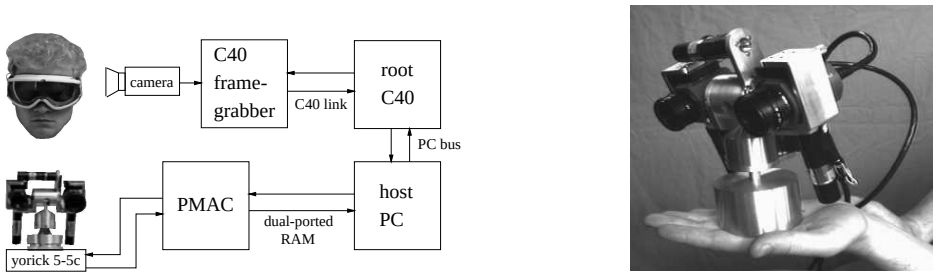

Figure 2: (a) The system for slaving Yorick to a user's head motions. The root C40 is used for communication only. (b) The Yorick 5-5C head platform.

The pose is mapped onto the pan and elevation degrees of freedom of the slave head platform, and angular demands are written into an area of memory in the host PC which is dual-ported and readable by the slave head controller. The controller is a Delta Tau PMAC, a DSP-based board for performing multi-axis servo control. The board drives the slave head platform via a power amplifier and obtains positional feedback information from encoders on the motors.

The stereo head platform, Yorick 5-5C, is one of a series designed and built in our laboratory. Yorick, shown in Figure 2(b), weighs less than 2kg, including cameras, and has an interocular baseline of 110mm. Each axis is driven by a DC motor with a Harmonic Drive geared transmission and capable of maximum accelerations in the range of $20{,}000 - 25{,}000°s^{-2}$, and maximum velocities of some $600°s^{-1}$. In the present work only the neck axes — that is, pan and elevation axes — are driven from neck pan and elevation movements of the master human head.

For robustness, features are supplied using IR emitting diodes attached to a visor worn by the operator. An example image is shown at the end of Figure 3. The algorithm assumes knowledge of which point in the structure has produced each point in the image. These correspondences must be established at initialization. The automatic initialization proceeds in two steps. First, the image is scanned and all connected regions above an intensity threshold are detected. The intensity threshold is constant and requires no adjustment. If there are not six of these regions, the scan is repeated until there are. These points are then ordered to match the order of the points in the structure model. The method of ordering is such that correct correspondences are guaranteed as long as all six points are visible and head cyclotorsion does not exceed 90°. Once running, the tracking algorithm proceeds by scanning a small region (presently 40 pixels square) centered on the previous position of each point. The centroid of all pixels above the intensity threshold is taken as the new point position.

When the user rotates his head to extreme angles, some of the LEDs may rotate out of view. As long as at least three of the six LEDs remain in view, we can continue estimating the pose, and predict the positions of the missing points. Specifically, whenever the region surrounding the previous position of a point contains no values above threshold, the predicted position of that point is defined as

$$\tilde{u}^i = f(p_x^i + t_x)(p_z^i + t_z)^{1} \quad \text{and} \quad \tilde{v}^i = f(p_y^i + t_y)(p_z^i + t_z)^{-1}.$$

A flag is set so that these predicted positions are not used in the pose estimation.

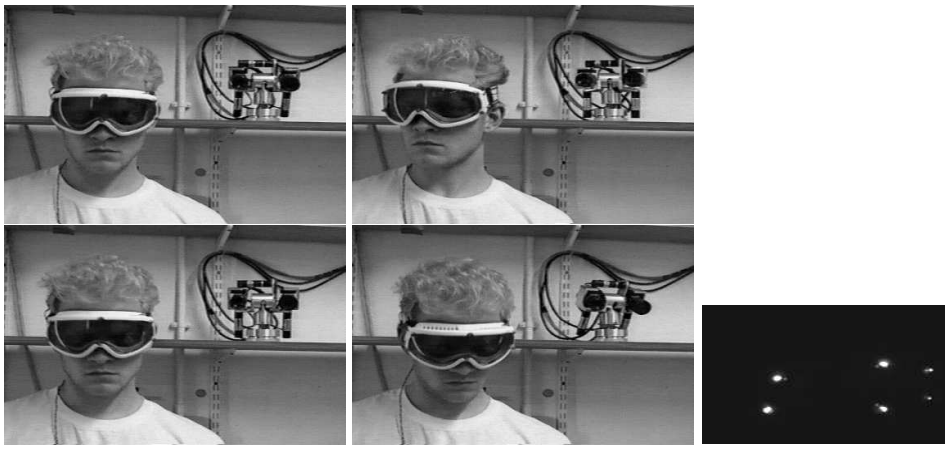

Figure 3: Frame from a short clip of 50Hz video of the slave head (right) being driven by the master (left). The final figure shows the IR leds on the visor.

When the point reappears and enters the search region it is acquired and the flag is reset.

Because the method is incremental, the estimate may drift slightly after many thousands of iterations due to roundoff error. This problem is most noticeable when the user returns his head to its original position and the estimate is not equal to the original one. We correct this error by using the original structure whenever the computed absolute rotation is less than some small angle arbitrarily set at five degrees.

4 Head slaving results and performance

Figure 3 illustrates the overall performance of the operator/slave head system using a sequence of images cut from a video made during use. The inter-image spacing is 150ms.

As noted earlier, all the processing takes place on a single C40 based framegrabber and proceeds at 50Hz. The pose estimation algorithm requires approximately 3ms. Because of the fixed 20ms clocking-out time of our CCD camera, this results in an overall latency of 23ms between the movement of the user's head and the corresponding motion demand being sent to the PMAC.

The bandwidth of the video tracker is limited only by the frequency with which new fields are received. Our system is presently limited (by the cameras) to 50Hz, resulting in a tracker bandwidth of 25Hz.

The pixel slew rate S of the tracker is limited by the size of the region checked for the new position of a particular point in the image. In the present implementation, this translates to a slew rate of 20 pixels per field, or 1000 pixels per second. The maximum trackable rotational velocity of the operator's head is then given by

$$\omega_{max} = \frac{S}{k}\left[\frac{fd\cos\theta}{z - d\cos\theta} - \frac{f(d\sin\theta)^2}{(z - d\cos\theta)^2}\right]^{-1}$$

where f is the focal length, k is the resolution in pixels/mm, d is the distance from the point being tracked to the relevant axis of rotation, θ is the angle of

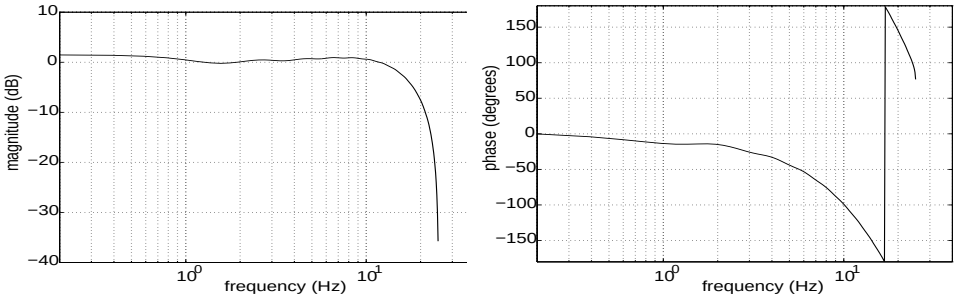

Figure 4: Bode plot of the response of the pan axis on the slave system.

the operator's head with respect to the forward direction, and z is the distance from the optic centre to the axis of rotation. The point has maximum velocity in the image when $\theta = 0°$. Typical values ($f = 8.5$mm, $k = 120$pixels/mm, $d = 150$mm, and $z = 1000$mm) yield a value of approximately $320°s^{-1}$ for ω_{max} in this situation. This limit can be raised by increasing the area of the search region (using another of the C40s in the network if necessary) or implementing a prediction scheme.

Another limit to the slew rate is the use of the small angle approximation. A value of $\omega_{max} = 500°s^{-1}$ is equivalent to a rotation of 10° per field. For this rotation, the small angle approximation has an error of less than 0.08°, smaller that the angular resolution of the entire system.

The frequency response of Yorick 5-5c is shown in Figure 4. The phase response corresponds roughly to a constant delay of 20ms. The phase response of the entire system is double this, corresponding to a constant delay of 40ms. The magnitude response can effectively be considered the magnitude response of the entire system, since the tracker's bandwidth is 25Hz. The bandwidth of the human head is similar, with only the most violent motions exceeding this range. As mentioned in the introduction, the slew rate of the head platform is some $600°s^{-1}$, which defines the maximum slew rate achievable by the system.

4.1 Accuracy

The accuracy of both the rotational and translational parts of the pose recovery has been tested using a 3D head model moved known amounts by a robot arm. Typical results of the rotational pose estimation experiments are shown for both pan and elevation axes in Figure 5(a). The accuracy on both axes is always within 1°. These measurements were made with the test device stopping momentarily at each demanded position, so that the results are not affected by the latency of the system.

Figure 5(b and c) shows data recorded during a dynamic test where the robot carrying the face template made simultaneous translation, pan, and elevation movements. Figure 5(b) shows the pan and elevation commands compared to the measurements made by the pose estimation algorithm. A comparison of translation commands with measurements recovered by the pose estimation algorithm is shown in Figure 5(c). These recovered values remain within 3% of the actual values throughout the test. In current work we are exploiting the ability to recover translation to control other actuators and degrees of freedom.

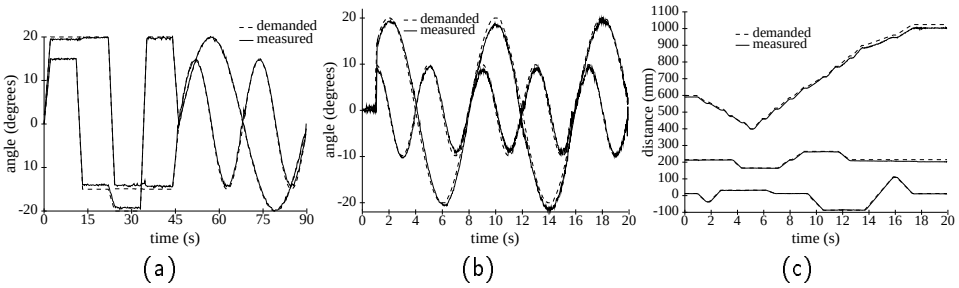

Figure 5: (a) *Static accuracy*: A comparison of commanded rotation of a 3D head model with the measurements of the pose estimation algorithm. The pair of traces at the top on the left of the graph represents the pan angle, the other pair elevation. (b & c) *Results of a dynamic test using translation, pan, and elevation*: (b) Another rotation comparison. The pair of traces with the greater amplitude represents the pan angle, the other pair elevation. (c) A similar comparison for translation commands. The top pair of traces gives distance along the optic axis $\mathbf{z}$, and the middle and bottom pairs along $\mathbf{y}$ and $\mathbf{x}$ respectively. The $\mathbf{y}$ measurements are shifted up by 200mm for readability.

5 On slaving camera to eye

If head movements can be slaved onto the neck axes of the electromechanical head, could not eye movements be slaved onto the vergence axes of the individual cameras? Certainly the eyes could be tracked — IR limbus tracking is one method which gives sufficient accuracy and appears practical within an enclosed head-mounted display. However, when considering the feedback loop for eye tracking, as a key difficulty emerges. Using an eye to control the camera in turn requires

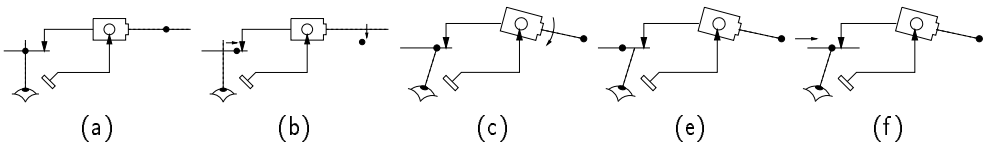

Figure 6: The need to move the display is illustrated in this step by step representation of events when a target moves.

the display viewed to be moved synchronously in front of the eye, either physically or, perhaps more simply, electronically. Figure 6(a) shows the camera and eye stabilized on a target. In (b) the target moves, so the image moves on the display. In (c) the eye moves to fixate the target, and so therefore, via eye tracking, does the camera. But, as (d) shows, if the display as a whole remains still, the target's image would return to the centre of the display. Diagram (f) shows that to achieve a stable fixation point, the display must move by the same angle as, and synchronously with, the camera.

The control of such a system presents three significant hurdles, viz frequency response, where a range of DC–100 Hz is required; resolution, where the eye tracker must have resolution of order 10^{-3}rad, and the display a movement resolution of 2×10^{-2}mm; and time delays in the control loop. Even if the first two are

cleared, our simulation suggests that the third is unsurmountable within existing technology.

5.1 Discrete-time simulation

In a discrete time simulation we model both the imaging of the scene and the sensing of eye movements as sample and hold processes. We also allow the eye controller to be a finite state machine, switching between tracking, saccading and a recovery period after a saccade.

There are delays associated with almost every component in the loop. It is obvious from the time stepped diagram that if the delay between the head movement and display movement is too great, then the system can be unstable. The delays considered in the simulation below are (i) Imaging delays Δ_{ci} — the time taken between something happening in the scene, and its being shown on the image; (ii) Head-to-display delays Δ_{hd} — the delay between the head making a move, and its being echoed by movement of the active area of the display; (iii) Eye delay Δ_{ie} — the delay between something appearing on the screen and muscular response; and Eye tracking delay Δ_{ef} — the delay between actual eye movement and demands being sent to the head.

At time step t, $t = 0, 1, \ldots$, let the scene be at angle $\theta_s(t)$ and let the head be at $\theta_h(t)$, where the latter has been derived from the previous iteration. The scene is imaged at angle $\theta_c(t) = \theta_s(t) - \theta_h(t)$. We model the vision (here, just image capture) and display process as one of sample and hold followed by delay, so that $\theta_i(t) = \theta_c{}^*(t - \Delta_{ci})$ where $\theta_c{}^*$ is θ_c after sampling, and Δ_{ci} is the delay of around 100ms.

The active area of the display is moved in response to camera movements, so that θ_d should mirror θ_d, but delayed by an amount Δ_{hd} which should ideally be identical with Δ_{ci}. Ie, $\theta_d = \theta_h(t - \Delta_{hd})$, and $\Delta_{hd} = 100$ms.

For slow phase movements, we model the eye by a neurovisual transfer function $I(s) = 4/(s + 4)$ which describes neuronal activity in response to an off-centred target [7]. In the slow phase, the muscular part of the eye plant (modelled as a critically damped second-order system with maximum acceleration of $30,000°s^{-2}$, and maximum velocity of $600°s^{-1}$ so that the transfer function is $E(s) = 50^2/(s + 50)^2$ can be safely ignored as it is much faster than the controller). In addition we insert a delay of $\Delta_{ie} = 100$ms. Using time steps of h, the update rule for the eye is therefore

$$\theta_e(t) = [4\theta_d(t) + \theta_e(t - 1)/h]\,[4 + 1/h]^{-1} \quad .$$

If the difference between the eye output and display input angles is greater than a threshold (here, 3°), the controller enters the saccade state. The current display input is stored and used a input throughout the entire saccade, as visual input is blurred and effectively useless. The eye is moved at its maximum velocity (here $500°s^{-1}$) to reduce the error. When within 0.5° of the held display input, the controller switches to the recovery phase which is similar to the slow tracking phase, but in which it is not possible to make a saccade for a period of 100 ms.

If the current position of the eye is $\theta_e(t)$, then the error in eye position is $(\theta_d + \theta_i - \theta_e)$, and this drives the eye plant. We assume that the eye tracker output produces a sampled, held and delayed version of the eye position $\theta_f(t) = \theta_e{}^*(t - \Delta_{ef})$ which is used to drive the vergence axis of the head, modelled as a critically damped second-order system with transfer function $R(s) = 120^2/(s + 120)^2$. This yields a discrete update equation for the camera axis on the head of

$$\theta_h(t + 1) = \left[\theta_f(t) - \left(\frac{M}{h^2} - \frac{f}{2h}\right)\theta_h(t - 1) - \left(k - \frac{2M}{h^2}\right)\theta_h(t)\right]\left[\frac{M}{h^2} + \frac{f}{2h}\right]^{-1}$$

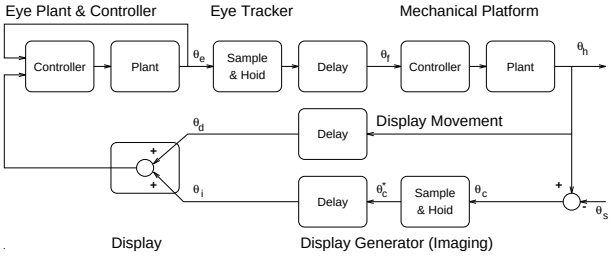

Figure 7: The discrete time system.

where we use $M = 1/120^2$ s^2, $f = 1/60$ s and $k = 1$. The simulations are carried out with $h = 0.001$s.

5.2 Simulation Results

We choose to show two results which together raise questions about the feasibility of the eye tracking approach. Figure 8(a) shows a 20° angular step change made by a target in the scene. Figure 8(b) shows overall position θ_d of the target relative to the resting direction of the eye. The important feature is the 25Hz saw-tooth oscillation, a result of sampling and holding by the imaging process during periods of fast camera motion.

If we consider just the slow phase tracking response of the eye, the eye position (c) shows the oscillations low-pass filtered, and their overall effect on the slaved head axis appears minor (d). However, when fast phase responses are introduced, a saccade is triggered in the eye in response to the oscillations, as Figure 8(e) show. The eye and the mechanical axis now overshoot and approach the target from the other side. and (f) show.

Without real experiment, we can only surmise that such effects would tire an operator and degrade degrade overall task performance.

6 Discussion

In this paper we have presented a system which uses visual "outside-in" tracking of features to slave the motion of a mechanical head to that of the operator's head. The overall dynamic performance is such that head motions of $\sim 400°\text{s}^{-1}$ can be followed.

The method of pose estimation recovers rotation to drive the head, but also recovers translation, which could be used to control further degrees of freedom. This advantage, combined with a potentially unlimited working volume provided by multiple cameras and the potential to extract other limb or facial gestures, gives this outside-in visual method enormous potential for operator sensing.

We have explored the potential of eye tracking to control individual camera movements with the mechanical head. Given error-free transfer of angular position in the forward and backward channels, our simulation indicates that an eye slaving system would be stable during slow-phase pursuit. However, introducing an eye controller able to make fast-phase motions introduces instability.

It appears that individual eye movements, principally vergence, will have to be achieved using machine vision. This is a goal of current work, the first practical step of which is to use the recovered translation and cyclotorsion of the operator's

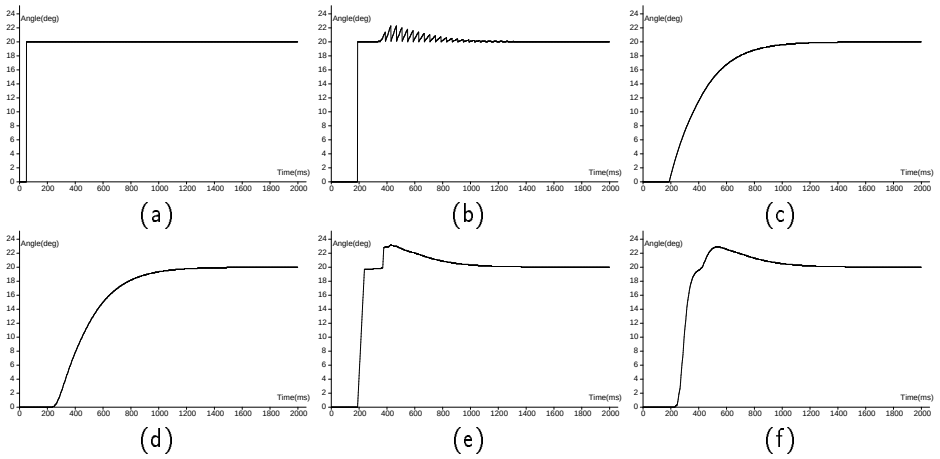

Figure 8: (a) The step move made by the target and (b) the effective angle θ_d of the target in the display with respect to the eye's resting direction. The saw-teeth are explained in the text. (c) is the eye output using only the slow phase controller, and (d) shows the resulting head motion. (e) is the eye output using a controller capable of making saccades, and (f) shows the resulting head motion.

head to drive the corresponding axes of a robot arm carrying Yorick — thereby duplicating all six degrees of freedom of the operator's head motion.

References

[1] R. Azuma and W. Ward. Space resection by collinearity: Mathematics behind the optical ceiling head-tracker. Technical Report TR 91-048, University of North Carolina, 1991.

[2] G. Bishop. *Self-Tracker: A smart optical tracker on Silicon.* PhD thesis, University of North Carolina at Chapel Hill, Department of Computer Science, 1984.

[3] A.M. Cook. The helmet-mounted visual system in flight simulation. In *Proceedings of Flight Simulation: Recent developments in technology and use*, Royal Aeronautical Society, London, April 1988.

[4] C. Harris. Tracking with rigid models. In A. Blake and A. Yuille, editors, *Active Vision*, chapter 4. MIT Press, Cambridge, MA, 1992.

[5] J. J. Heuring and D. W. Murray. Visual head tracking and slaving for visual telepresence. In *Proc. IEEE Int Conf. on Robotics and Automation, Minneapolis, May 1996.* IEEE Computer Society Press, 1996.

[6] R.S. Kalawsky. *The Science of Virtual Reality and Virtual Environments.* Addison-Wesley, Wokingham, 1993.

[7] D. N. Osherson, S. M. Kosslyn, and J. M. Hollerbach. *Visual Cognition and Action, Volume 2.* 1988.

[8] F. Raab, E. Blood, O. Steiner, and H. Jones. Magnetic position and orientation tracking systems. *IEEE Transactions on Aerospace and Electronics Systems*, 15(5):709–717, 1979.

[9] J.F. Wang. *A real-time optical 6-D tracker for Head-Mounted Display systems.* PhD thesis, University of North Carolina at Chapel Hill, Department of Computer Science, 1990.