Bernard D. Adelstein
NASA Ames Research Center,
MS 262-2
Moffett Field, CA 94035
and
University of California
Berkeley, CA 94720

Eric R. Johnston
and
Stephen R. Ellis
NASA Ames Research Center
MS 262-2
Moffett Field, CA 94035

Dynamic Response of Electromagnetic Spatial Displacement Trackers

Abstract

Overall system latency—the elapsed time from input human motion until the immediate consequences of that input are available in the display—is one of the most frequently cited shortcoming of current virtual environment (VE) technology. Given that spatial displacement trackers are employed to monitor head and hand position and orientation in many VE applications, the dynamic response intrinsic to these devices is an unavoidable contributor to overall system latency. In this paper, we describe a testbed and method for measurement of tracker dynamic response that use a motorized rotary swing arm to sinusoidally displace the VE sensor at a number of frequencies spanning the bandwidth of volitional human movement. During the tests, actual swing arm angle and VE sensor reports are collected and time stamped. By calibrating the time stamping technique, the tracker's internal transduction and processing time are separated from data transfer and host computer software execution latencies. We have used this testbed to examine several VE sensors—most recently to compare latency, gain, and noise characteristics of two commercially available electromagnetic trackers: Ascension Technology Corp.'s Flock of Birds™ and Polhemus Inc.'s Fastrak™.

1 Introduction

If virtual environment (VE) technology is to be employed effectively and reliably in experimental human factors research, scientific data visualization, or human operator training, it must be subjected to the same rigorous performance characterization procedures as any other engineering or scientific apparatus. Objective quantitative understanding of VE system characteristics would permit assessment of the quality of experimental and analytic results, and prediction of the expected benefits of training implemented through VE techniques. For developers of VE hardware and software, detailed quantitative descriptions of VE systems and components will enable optimization to maximize performance. All consumers of VE hardware and software stand to profit from critical objective data on how well one component or configuration rates against another.

To date, characterizations have been developed for both spatial and temporal properties of a number of VE systems and components. The rational first approach to assessing VE performance has been to treat temporal and spatial properties as being separable—i.e., spatial characteristics are considered time-invariant, and dynamic characteristics are assumed to produce purely temporal phenomena that have no spatial element. For example, distortions in the static

Presence, Vol. 5, No. 3, Summer 1996, 302–318

spatial characteristics of commercially available VE displacement sensors, commonly referred to as *trackers,* have been examined by Hirose, Kijima, Sato, and Ishii (1990), Burdea, Dunn, Immendorf, and Mallik (1991), Harwin (1991), Bryson (1992), and Holloway (1995). Timing analyses of VE *systems* have been reported by Hirose et al. (1990), Bryson and Fisher (1990), Liang, Shaw, and Green (1991), Mine (1993), Wloka (1995), and Holloway (1995). In several of these studies, the latency component associated solely with the spatial tracker is claimed to have been isolated. Additionally, Adelstein, Johnston, and Ellis (1992) and Krieg (1993) described equipment and procedures for measuring tracker latency directly.

A review article by Meyer, Applewhite, and Biocca (1992), listing widely disparate tracker latency values for the Polhemus Isotrak®, demonstrates the inconclusiveness of the reported data. Furthermore, with one exception, none of the original sources referenced by Meyer et al. (1992) describe the methods used to determine tracker latency. The controversy over latency claims continues in a recent National Research Council review of tracker technology (Durlach & Mavor, 1995) in which 20–30 msec values attributed to an uncited independent source call into question the 4 msec latency advertised by Polhemus, Inc. for the Fastrak™.

Overall system latency—the time elapsed from motion of the user's instrumented hand or head-mounted display until representation of that movement in the display—remains one of the more widely acknowledged shortcomings of current VE technology. For the human interacting with a VE, excessive system latency can trigger "simulator sickness" due to sensory mismatch or discrepancy between signals in ocular and vestibular pathways (Sheridan & Furness, 1992). Excessive time delay prevents adaptation between mismatched visual input and motor output when performing manual tasks (Held, Efsathiou, & Greene, 1966) and is postulated to correspond to a reduction in a subjective sense of "presence" (Held & Durlach, 1991). Sheridan and Ferrell (1963) demonstrated that human subjects slow their manual response time by adopting a "move-and-wait" strategy to preserve final point accuracy when visual feedback is delayed. It is also well known from control engineering principles that time lag reduces the stability and effective bandwidth of feedback systems.

Overall latency in the typical VE is a sum of hardware and software processing times inherent in the components that form the complete system. These major components include the transducers that sense human input (e.g., manual, voice), the simulation engine that generates the VE and governs human interaction with it, and the rendering equipment (e.g., visual, aural, haptic) that displays the simulation output. Of equal significance is the finite communication time required to transfer information between these components.

Update period (period = 1/rate) is a distinct quantity, different from latency—e.g., consider a 30-Hz NTSC video signal that is tape-delayed by minutes, hours, or days. Various different update periods or "sample and hold" intervals can be present in a VE system. These include the time elapsed between successive transducer samples of human input, the interval between successive transfers of processed transducer measurements to the simulation engine, the cycle time for the simulation engine to recompute various VE model elements, and the scan time during which the output display is rerendered. Since, in general, sampling in different system components does not occur synchronously, the effective update period of a complete VE system can nominally be considered as the time to refresh the slowest component. As with latency, unless specifically controlled, there is no reason to expect a system's update rates to be constant.

Latency and update period, however, are partially related. One component of latency for an individual VE hardware or software element is the time for its final output to be revised as a function of input—i.e., its update period. Another factor in latency is the time to synchronize or trigger that VE element to act—i.e., the time spent by that element waiting for input. Finally, filtering to smooth or condition data adds lag because memory from previous time steps is retained.

The purpose of this paper is to describe and demonstrate apparatus and an engineering method for measuring the components of dynamic response directly attributable to the transduction and processing internal to spatial displacement sensors (i.e., trackers) in the VE

system. The objective behind the method is to develop general quantitative descriptions of how these sensors would perform when driven by normal human motion. By *dynamic response,* we mean not only the timing of sensor output in response to motion input, but, also, how the displacement magnitude reported by the sensor is affected. The goal in demonstrating the apparatus and method is to evaluate the two commercially available electromagnetic trackers currently in widespread use: Polhemus Inc.'s Fastrak and Ascension Technology Corp.'s Flock of Birds™. This paper will not concentrate on update rate since these specifications are readily available—either provided directly by hardware vendors, calculated from software cycle times, or monitored with an oscilloscope.

2 Related Work

As noted above, several other investigators have measured latencies in VE systems. Their measurement techniques and contributions to the assessment of tracker performance are reviewed here in more detail.

Bryson and Fisher (1990) employed video mixer superposition to compare the displacement of a synthetic VE marker driven by an electromagnetic tracker against the video image of the real tracker's receiver. In one series of tests, by knowing the video frame rate, they calculated the time to the start of VE marker motion following a sudden manual displacement of the sensor. In a second series of tests, they measured velocity of the sensor and displacement errors between the tracker and the marker in the VE to estimate time lag.

Liang et al. (1991) analyzed latencies so that they could develop predictive filters to alleviate consequences of VE system lag. In their experiments, readings from an electromagnetic tracker whose receiver was attached to a pendulum and the corresponding time stamps generated by the host computer were stored. Simultaneously, a video camera recorded the pendulum swing along with a computer monitor display of the current time of the clock used to generate the tracker time stamps. They then compared the time stamps in the stored tracker data at zero position crossings with the clock time appearing in the video camera record at the instant when the pendulum was actually seen to pass through the same point.

Hirose et al. (1990) also measured latencies in the process of parameterizing corrective prediction algorithms. They compared the plotted time histories of independently measured point-to-point head yaw trajectories against VE sensor output.

Mine (1993) separated the portion of the delay due to internal tracker processing from overall (i.e., sensor-to-display) VE system latency. As in Liang et al. (1991), the tracker's receiver was affixed to a simple gravity pendulum, but instead of using video image analysis, zero position crossings were marked by the swinging pendulum's optical interruption of an LED-photodiode pair. Tracker latencies were estimated on an oscilloscope screen by comparing the timing of the photodiode's zero-crossing transitions against the on-line D-to-A converted (by a board on the host computer) tracker readings. Additionally, when a zero-crossing was detected in the tracker's output, the computer application toggled a single polygon from black to white (or vice versa) on the system video display. The output from a second photodiode monitoring changes in the polygon's brightness was then compared with the first photodiode's zero crossing report to provide a measure of overall end-to-end system latency.

Polhemus Inc. proposed a theoretical basis comparable to a disk drive's *average* seek time specification for direct estimation of tracker internal transduction and processing latency, separate from data transmission time to the host computer (Polhemus, 1992; Murry, 1993). Krieg (1993) outlined Polhemus Inc.'s experimental apparatus and procedure for verifying tracker latency. Their apparatus consists of a manually propelled one degree of freedom swing arm on which the tracker's receiver is mounted. The swing arm terminates data transmission to the host computer (or dumb terminal) instantaneously by interrupting a light source–phototransistor pair located at a fixed reference position on the apparatus. Based on an assumption of constant swing arm velocity while passing through the known fixed reference position, tracker latency is extrapolated from the last *uninterrupted* position report received by the computer prior to termination of the serial data stream. The

temporal resolution of the test method described by Krieg (1993)—which is otherwise limited to the tracker's sampling period—can be enhanced by timing the additional data bits that follow after transmission of the last uninterrupted report until termination of the serial stream (Jones, 1993).

In the studies summarized above, all measurements, with the exception of those by Polhemus Inc., included execution times for various elements of the application software. These software elements are responsible for handling the sensor interface, drawing to the display monitor, interprocessor communication in multiple computer systems, and, in the case of Mine (1993), D-to-A converter operations to produce "analog" tracker output for the oscilloscope. Consequently, even when intended to, these studies did not report the performance of the tracker alone. Furthermore, none of the authors described a complete timing characterization or calibration of their various system components—including those software or hardware components that served as the "latency measurement instruments." Calibration, even for a highly nondeterministic UNIX environment, can separate systematic instrument biases from stochastic measurement uncertainty, thereby allowing more accurate estimation of tracker performance. Stochastic uncertainty can be reduced by averaging many repeated latency measurements. Biases due to imprecise determination of the actual zero crossing position can be removed by combining latency estimates from bidirectional (i.e., positive- and negative-going) motion. With the exception of Bryson and Fisher (1990), none of the cited studies mention explicitly whether multiple data samples were combined or averaged.

When human motion provides the displacement input to the VE sensor, the site where the independent instrument (the device with known characteristics against whose readings the VE sensor's performance is compared) makes its measurements becomes a concern. Complications arise if the independent instrument measures a nearby point on the human's body rather than a location that is fixed precisely relative to the VE sensor's housing. Unknown dynamics can be introduced into the VE sensor's orientation and position relative to the body point because of imperfect sensor mounting and the biomechanical compliance of underlying human tissues. *Involuntary* human movement can introduce random motion input with frequency content up to 10 Hz and higher, and therefore can add tens of milliseconds or more of uncertainty to lag measurements. One way to deal with this potential problem when studying sensor dynamic response is to follow Liang et al. (1991) and Mine (1993) and use an easily characterized mechanical linkage rather than a human as the motion input source.

Liang et al. (1991) and Mine (1993) did not mention that they had systematically varied the pendulum length during their tests, leading to the inference that only a single input frequency was used by each. The pendulum frequencies in both cases were not disclosed. Bryson and Fisher (1990), Hirose et al. (1990), and Krieg (1993) allowed for only single direction, nonreversing, tracker motion during each test and therefore do not have an oscillatory frequency. Given the data they reported, it is not possible to distinguish whether the slowed tracker response they measured was solely the result of transport delay or whether additional filter lag was present. If measured time delay could be shown to be constant at all input motion frequencies, transport phenomena such as pipeline buffering would have been implicated. Time delay that varied according to input frequency would have pointed to the presence of filtering. If the oscillatory displacements of both the pendulum input and the sensor output were measured, gain magnitude (output amplitude divided by input amplitude) versus frequency could have been calculated. Constant gain over the range of frequencies would have indicated pure transport delay. Varying gain (i.e., attenuation) would have implied that sensor data are being filtered, either intentionally as part of signal processing or due to the characteristic physics of the sensor.

3 Apparatus

This section describes the hardware and software for our tracker testbed—a device akin to the "shaker table" apparatus commonly used in characterizing the dynamic response of structures, machinery, as well as many other kinds of transducers. The current version of

the testbed incorporates several improvements over an initial implementation reported by Adelstein et al. (1992).

3.1 Design Criteria

Based partially on our review of previous work, we compiled the following list of design objectives for our dynamic response testbed. The testbed should

- preserve the sensor's intrinsic transduction capabilities by not introducing additional spatial distortions or noise;
- be easily adaptable to accommodate a variety of different sensors while still adhering to a single uniform test method;
- apply *controlled* displacement inputs to the VE sensor;
- span the frequency range of human limb and head motion—typically up to 3–5 Hz for all but the smallest amplitude volitional hand movements and 8–10 Hz for involuntary tremor;
- eliminate the need for direct human contact with the sensor during data collection to avoid the uncertainties of human-driven motion;
- provide for rigid attachment of the sensor to its motion source to prevent any drift in its position or orientation during testing;
- provide accurate independent measurement of the input displacement to the sensor;
- provide accurate time stamps relative to the same time base for both the independently measured input displacement and the VE sensor output;
- allow measurement of internal tracker processing time to be separated from any other data communication and computation activity in the VE system.

3.2 Hardware

The testbed hardware components, along with a typical VE spatial tracker, are shown schematically in Figure 1. The testbed components include an IBM AT (8 MHz; 80286 CPU) personal computer with two serial ports, external programmable motor controller and servo amplifier, a manually adjustable function generator voltage source, and a motorized swing arm.

The swing arm apparatus, depicted to scale in Figure 2, is driven by a permanent magnet DC servo motor (Model ME-5370, EG&G Torque Systems, Watertown, MA) bolted to a rigid acrylic plate and wood frame base. The swing arm itself is a 1.0-in. (25.4-mm)-diameter solid glass-epoxy rod, firmly clamped to the top end of the motor shaft. A polycarbonate mounting block that accepts a variety of adapter plates for any of several common spatial trackers (e.g., Ascension, Polhemus, and Logitech devices) is located at the distal end of the swing arm. Nylon screws are used both to secure the sensors to the mounting block and to clamp the halves of the mounting block onto the swing arm. The distance from the spatial sensor to the motor shaft center of rotation, r, is adjustable, either by changing the length of glass-epoxy rod, or by sliding the mounting block along the rod. Padded limit stops fixed to the frame restrict swing arm angle, θ, to a maximum range of $\pm 40°$. The total weight of the swing arm apparatus is approximately 75 lb (35 kg).

During tests with electromagnetic trackers, the transmitter was fastened to the top of a sturdy nonmetallic base—an upside down 35-gallon (132.5-liter) plastic garbage can. The tracker's receiver was attached to the top surface of the mounting block with the swing region facing the transmitter. Referring to Figure 2, z_0, the height difference between transmitter and receiver centers, ranged from 13.3 to 14.2 in. (33.8 to 36.1 cm), depending on receiver and transmitter size. In all of our experiments with electromagnetic trackers, the separation between the center of the receiver and motor axis was set to $r \approx 14.5$ in. (36.8 cm). The horizontal distance from the center of the transmitter to the center of rotation of the swing arm was fixed at $x_0 = 26$ in. (66.0 cm), ensuring that the tracker's receiver always remained within an acceptable 17.5 to 22.5 in. (44.5 to 57.2 cm) operating range from its transmitter. With this relative placement of the receiver and transmitter, the tracker did not suffer from any interference that could otherwise have been caused by the motor and the underlying metal floor tile in our lab.

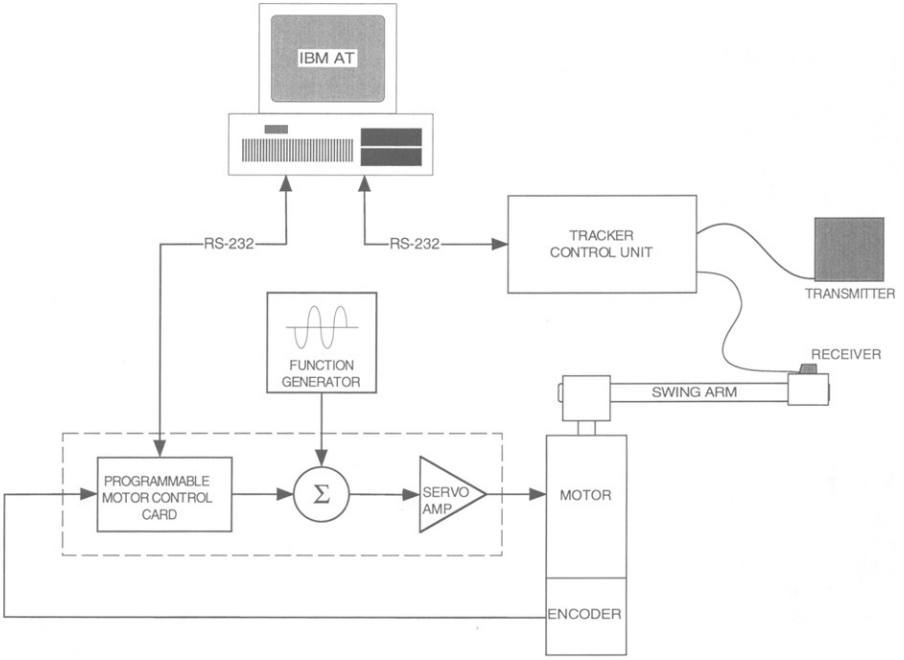

Figure 1. *Testbed hardware components.*

The effect of the motor and the rest of the lab environment on each of the trackers to be tested was checked informally during quasistatic displacements of the swing arm. With the tracker output set to continuous or stream mode, x and y tracker data were drawn directly on the host computer monitor and not erased.[1] The result of moving the swing arm very slowly (at much slower velocities than the lowest frequency dynamic tests) was a thin circular arc plotted on the screen, representing the trajectory of the tracker's receiver. No noise or deviations from the constant radius expected for the fixed swing arm length were witnessed with the motor either powered or unpowered.

Angular displacement of the swing arm is measured directly by a 1024 cycle/revolution incremental optical encoder coupled directly to the motor shaft. The encoder output is quadrature decoded to an angular resolution of 0.09°—0.02 in. (0.5 mm) at the 14.5-in. swing arm radius—by the motor control card (Model SMCC, Delta Tau Data Systems, Canoga Park, CA). Decoded swing arm angle values are always available within a specified maximum latency of 960 μsec (one motor controller sample and hold period) for return to the host computer via an RS-232 serial line. The other RS-232 port on the computer serves as the interface to the

1. "Continuous" is Polhemus, Inc.'s term and "stream" is Ascension Technology Corp.'s for the same mode of tracker data transmission.

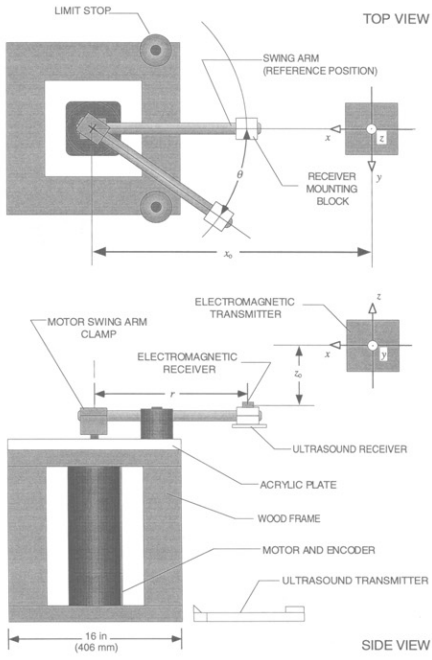

Figure 2. *Testbed swing arm apparatus. The motorized swing arm is shown with the electromagnetic tracker coordinate frame. The coordinate origin is adjusted when the ultrasound tracker is used.*

tracker. During all of our experiments, both serial ports were operated at 19.2 KBaud.

As shown in Figure 1, angle measurements from the encoder are fed back through the combined gains of the motor control card and servo amplifier (Model 215H, Copley Controls, Westwood, MA) to close a position servo loop around the swing arm motor. Sinusoidal position commands to the servo loop are provided by a function generator (Model 3311A, Hewlett Packard). The swing arm oscillatory amplitude and frequency are selected by manually adjusting the controls of the function generator. When the controller is set to maximum stable feedback gains, the function generator can drive full amplitude (±20°) steady-state sinusoidal swing arm motions at up to 4 Hz. Reduced amplitude (±5°) sinusoids can be obtained up to at least 9 Hz. We chose this method to produce swing arm oscillations rather than rely on the control card to generate time varying reference commands, because of the particular motor controller's inability to produce smooth sinusoids at the frequencies of interest to us. Advantages of our testbed over a simple pendulum are (1) the frequency of oscillation can be changed without altering the length of the swing arm, and (2) with constant amplitude sinusoids from the function generator, swing arm oscillation magnitude is undamped.

3.3 Software

The testbed control software was written in C (Borland Turbo C++ v 1.01) and used under MS-DOS (v 3.30). The software was responsible for downloading initialization parameters specific to the motor control board (e.g., servo gains) and to the particular tracker under examination (e.g., operating mode, data format, and report type), collecting data from the tracker, requesting and collecting encoder shaft angle data from the motor controller, time stamping the data, and writing data to the computer's hard disk at the end of each test for later analysis.

The device drivers for the various VE trackers tested were written in-house. The time stamping function called by our software is part of a general purpose precision timing library (PCHRT, Ryle Design, Mt. Pleasant, MI). The functions in this library access the computer's internal hardware timer *directly,* in advance of its input to the 16-bit counter responsible for the standard 18.2-Hz BIOS clock tick. PCHRT library functions have a specified resolution of 1 μsec. Because of software overhead in the particular PCHRT time stamp function call, as explained in Section 3.4, an individual time stamp has an accuracy to within 615 μsec for our computer.

For the purposes of these tests, the trackers were operated in continuous or stream mode. Thus, the computer receives only tracker data—it performed no polling or

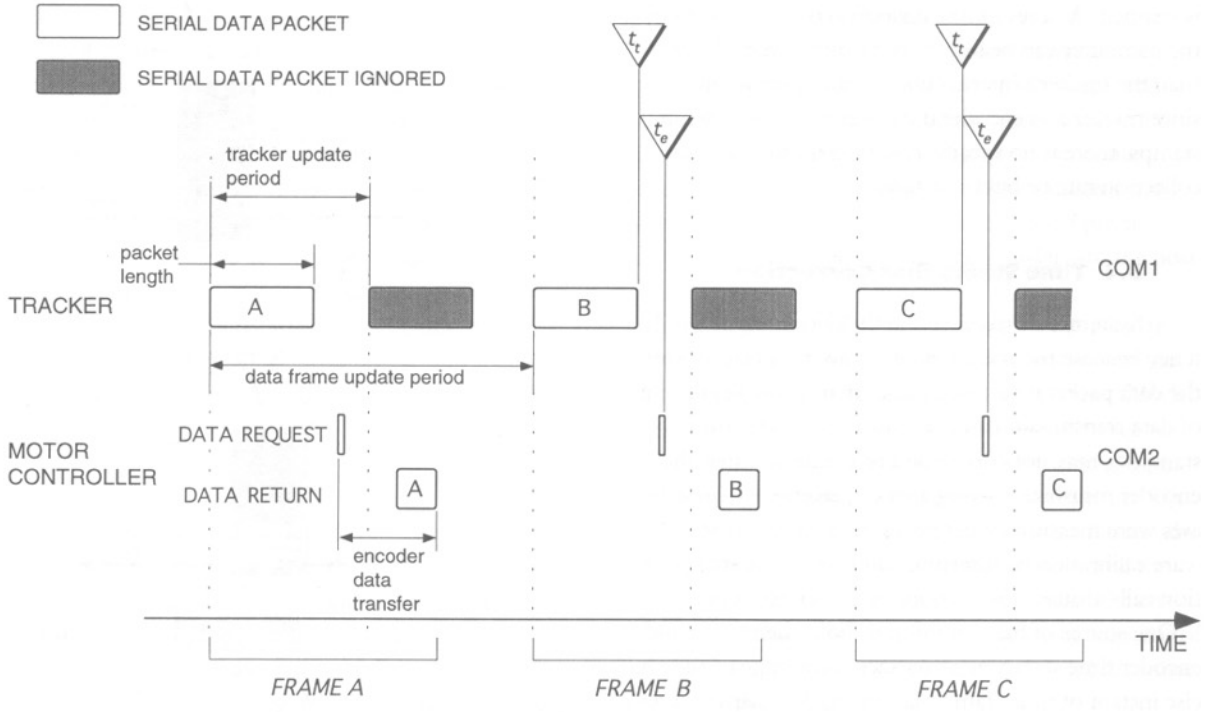

Figure 3. *Data collection and time stamping procedure. The relative timing for three consecutive updates—data frames A, B, and C—is shown. Each frame includes one tracker and one encoder data packet.*

sensor synchronization function. This ensures that the *tracker's* internal clock governed the pace of the data collection cycle, including interface application code execution on the computer as well as encoder data transmission from the motor control card.[2]

The operating principle for collecting and time stamping the data from a spatial tracker and the motor shaft encoder is illustrated by the timing diagram in Figure 3. Data packets containing successive samples of position and orientation information become available for transfer from the tracker to computer serial port COM1 at the tracker's internal sample update rate. Upon detection of the start of a new packet by the interface application, arriving tracker data are collected into a buffer in memory. Immediately after the packet has been read completely, a time stamp (t_t) is affixed to this tracker data and a byte is sent to the motor control card via serial port COM2 requesting the current encoder angle. A second time stamp (t_e) is placed immediately after this data request byte is sent. Each tracker packet plus its time stamp, and the corresponding encoder angle value plus its time stamp constitute a data frame (e.g., one of frames A, B, C, etc. in Fig. 3) that is appended as a row to a larger data array buffer in computer heap memory. To maximize data collection rates by the computer, the data buffer is written away to disk only after available heap space has been completely filled. After storing the data on disk, the buffer is cleared for refilling and data collection resumes. The process of collecting tracker and encoder data and time stamping is repeated until halted from the computer keyboard.

Note that if the next tracker update becomes available before the transfer of encoder data from the motor controller has been completed, the incoming tracker packet

2. With an oscilloscope on the serial line, we observed that the response time of a single receiver Fastrak to a polled "single data record" request varies between 7 and 15 msec, depending on when the request arrives during the tracker's update cycle.

is skipped. As a result, the collection of tracker data by the computer can be one, two, or more times slower than the tracker's internal update rate. Furthermore, since tracker and encoder data each have their own time stamps, there is no specific requirement that the data collection rate be held constant.

3.4 Time Stamp Bias Correction

Systematic biases arise in the determination of latency because the tracker time stamp, t_t, is placed after the data packet is received rather than at the beginning of data transmission and because the encoder time stamp, t_e, may not correspond to the instant that the encoder measures a swing arm displacement. These biases were measured where possible during testbed software calibration by inserting additional time stamp function calls around key portions of the source code.

One source of bias common to both the tracker and encoder time stamps is an inexact knowledge of the precise instant of time stamp placement. The elapsed interval from the initiation of the call to t_check_timer, the PCHRT function that actually queries the computer clock for the time stamp, until the return to the subsequent line of source code in the calling routine was measured on our computer to be 615 μsec (standard error ±1 μsec for 100 samples). While the time from initiation of the function call to the actual instant when the computer's internal clock is queried (α in Fig. 4a and b) cannot be ascertained exactly, it is assumed constant because the 615 μsec execution time associated with t_check_timer does not vary with location of the time stamp call in the testbed software.

A second source of bias is finite serial data transmission time—from the tracker to the host computer and from the computer to the motor control card. Average transmission times of 521 to 526 μsec per byte (standard errors less than ±4 μsec for each average of 200 to 400 consecutive data packets) at 19.2 KBaud were measured for both trackers under the variety of position and orientation data packet lengths considered in this study. Thus, assuming the theoretical transmission time of 0.521 msec/byte (10 bits per data byte, including stop and start bits, at 19.2 KBaud), the first byte of a data

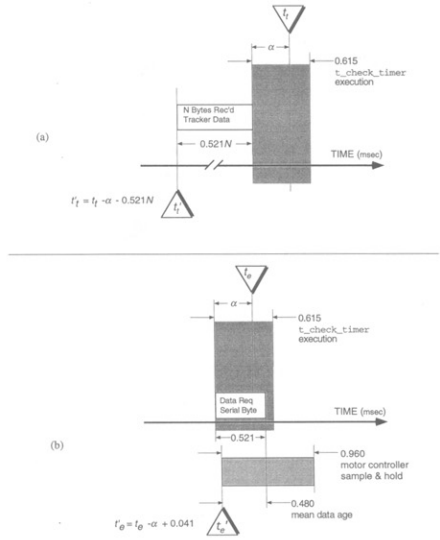

Figure 4. *Time stamp bias corrections. (a) Tracker time stamp: raw (t_t) and corrected (t'_t). (b) Encoder time stamp: raw (t_e) and corrected (t'_e).*

packet is ready to be released by the tracker at the corrected time

$$t'_t = t_t - \alpha - 0.521N \text{ msec} \quad (1)$$

where t_t was the actual time stamp placed *after* the receipt of all N bytes of tracker data as depicted in Figures 3 and 4a.

Immediately after receipt of the N bytes of tracker data and placement of the tracker time stamp, t_t, a data request byte is transmitted via serial line COM2 to the motor control card to capture the current encoder measurement of swing arm angle. The encoder time stamp (t_e) function call to t_check_timer follows immediately the execution of the software code that sends the encoder data request byte to computer port COM2. Thus we assume that the call to t_check_timer begins concurrently with the 0.521 msec long transfer of the data request byte from COM2 to the motor controller as

shown in Figure 4b. Receipt of the data request byte freezes the current encoder angle sample in a buffer on the motor control card for serial transmission back to the host computer. However, since the motor controller is not synchronized with respect to the computer, encoder data, which were actually measured at the start of the controller's sample-and-hold cycle, have a uniformly distributed age between 0 and 960 μsec with an expected value of 480 μsec at the time the data request is received. Summing the bias contributions for the time stamp function call (α), serial data request byte, and motor controller asynchrony, the corrected time stamp in Figure 4b is given by

$$t'_e = t_e - \alpha + 0.521 - 0.480 \text{ msec} \quad (2)$$

Because the relative timing differences between tracker and encoder events rather than the individual corrections given by Eqs. (1) and (2) for each of t_t and t_e are of concern, a single overall correction may be applied to the tracker and encoder time stamps. Subtracting Eq. (2) from Eq. (1), the bias corrected times can be written as

$$t'_t - t'_e = (t_t - t_e) - \Delta \quad (3)$$

where

$$\Delta = (0.041 + 0.521N) \text{ msec} \quad (4)$$

An advantage of this representation is the elimination of α, the unmeasurable constant associated with both the t_e and t_t time stamps.

4 Experimental Method

Tracker settings for the dynamic response tests were chosen to minimize the time for internal sensing and processing of electromagnetic field activity into raw position and orientation information and to ensure that the devices under examination (Ascension Technology Corp.'s Flock of Birds and Polhemus Inc.'s Fastrak) operated from a common baseline. First, *all* user-selectable internal filtering was disabled. This eliminates processing by unspecified filter structures whose performance may not be directly comparable between the two devices. Second, as noted in Section 3.3, only continuous or stream mode data transfer was used to ensure that the trackers ran freely and were never slowed by the computer requests for data. Third, both devices reported their measurements in fixed integer binary format. Finally, testing was restricted to single receiver configurations, removing from consideration the various ways multiple receiver devices could be connected physically—and their output queued for transfer—to the host computer.

At the beginning of the test session for a particular device, the receiver was fastened to the attachment block at the end of the testbed swing arm. Next, using fiducial marks on the lab floor and a plumb bob, the swing arm was carefully aligned so that its starting position/orientation lay in the tracker transmitter's x–z plane (see Fig. 2). Applying power to the motor control card in the absence of function generator output voltage locked the swing arm in this starting position and zeroed the encoder count. The x–y–z position of the swing arm axis of rotation (i.e., the motor shaft) was then noted for subsequent data processing.

Swing arm servo gain and the function generator's sinusoidal frequency and amplitude were set prior to the start of each trial in a series. Servo gains were selected such that possible loop resonances did not coincide with the function generator output frequency. A test series consisted of a sequence of single trials at frequencies of 0.25, 0.5, 1.0, 2.0, 4.0, 6.0, and 9.0 Hz.

For a given test series, only one of either "position only" (*P*), "position plus Euler angle orientation" (*PE*), or "position plus quaternion orientation" (*PQ*) data report types was requested from the tracker. Because of heap memory size limits and different amounts of data per frame for each report type, *P* test records comprised two sets of 655 frames of data uninterrupted by disk writes, *PE* tests three sets of 379 uninterrupted frames, and *PQ* tests three sets of 331 uninterrupted frames.

5 Data Analysis

5.1 Coordinate Conversions

Our dynamic response analysis is based on the time domain comparison of the swing arm angle as measured

by the encoder, θ, against the swing arm angle reconstructed from tracker position or orientation reports. In quantifying tracker response to input oscillations, the output of the encoder is treated as the reference or "true" measurement of swing arm position because of its documented angular accuracy, resolution, and latency.

Data processing requires initial conversion of both encoder and tracker readings to a common physical dimension: swing arm angle in degrees. The encoder output, already in angular form, only has to be rescaled. Under the assumption that swing arm motion lies parallel to the *x–y* plane of the transmitter shown in Figure 2, three different tracker based estimates of swing arm angle can be made. In the case of Cartesian coordinate position output, the estimate of swing arm angle is

$$\phi_c = \tan^{-1}\left(\frac{X_t}{Y_t}\right) \tag{5}$$

where X_t and Y_t are the rescaled and offset adjusted Cartesian tracker displacements. For quaternion output,

$$\phi_q = 2\sin^{-1}(q_z) \tag{6}$$

where q_z is the rescaled z component of the quaternion.[3] For the Euler report type, angle ϕ_e is simply the rescaled azimuth value.

Typical segments of θ, and computed ϕ_c, ϕ_q, and ϕ_e time histories for 1-Hz sinusoidal swing arm inputs to the Polhemus Fastrak and Ascension Flock of Birds, are illustrated in Figure 5. These plots exhibit several attributes common to all the tracker data collected. First, we note that tracker position (ϕ_c) and orientation based curves (either ϕ_q or ϕ_e) are not coincident either with the encoder measurement (θ) or with each other. Second, while Fastrak outputs are smooth they still have a slight, though noticeable, distortion from the near-ideal sinusoid measured by the encoder. Third, the Flock of Birds has a distorted output that is more noiselike in appearance—in fact the distortion makes it look at times as if the ϕ_e orientation output leads encoder measurements of swing arm position.

3. Since motion of the swing arm theoretically lies in the *x–y* plane, the *x* and *y* components of the quaternion are equal to zero. The *z* component of the quaternion, q_z, is chosen because it has a larger dynamic range in terms of the tracker's fixed precision binary output than $q_0 = \cos(\phi/2)$, the scalar quaternion component.

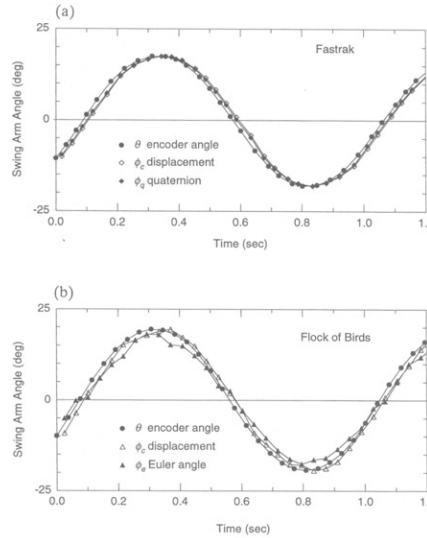

Figure 5. *Tracker and encoder response to a 1-Hz sinusoidal swing arm input. (a) Fastrak, (b) Flock of Birds. Markers indicate the individual sample points.*

Also evident in Figure 5 is the nonconstant sampling rate typical for data collection in these tests. Depending on the report type (i.e., *P, PQ, PE*), average data transfer rates to the computer for either tracker ranged between 25 and 40 Hz. Since the Flock of Birds and Fastrak in stream or continuous mode measure single receiver data at default rates of 100 and 120 Hz, respectively, on average a minimum of two of every three measurements transferred by the tracker is skipped by the computer (see Fig. 3). In the example data of Figure 5, occasionally as many as four consecutive tracker samples are skipped.

5.2 Latency, Gain, and Noise Estimation

Approaches such as interpolation and parametric model fitting were considered necessary for determining tracker-to-encoder latency because achievable tracker

sampling rates are too slow to have sufficient resolution for cross-correlation and cross-spectral analyses or for direct observation of zero crossings. Furthermore, conventional digital correlation and Fourier transform methods cannot be used because of nonconstant data collection rates—even with continuous mode output from the trackers.

In our earlier work (Adelstein et al., 1992), latency estimates were based on time differences between successive encoder and tracker zero position crossing times. The time of each positive- or negative-going zero crossing was linearly interpolated from the two sampled data points immediately adjacent to (i.e., the one preceding and the one following) the zero crossing.[4] Thus, the interpolation scheme used only four data points per oscillatory cycle—two per negative- and two per positive-going zero crossings. By comparison, the sinusoidal model fitting scheme described below employs all points in the sampled record. Consequently, our earlier interpolation scheme is less reliable, especially with noisier data, when low-frequency swing arm motion reduces the number of zero crossings per record.

Amplitude fidelity was calculated in our earlier study (Adelstein et al., 1992) by comparing RMS tracker and encoder magnitudes computed directly from raw sensor measurements. No effort was made in that study to separate measurement noise or random motion variation from the intended swing arm path.

Latency and gain estimates for the present study are derived from a least-squares fitting of each encoder and tracker test record to an ideal sinusoidal model. We chose this method for our current study specifically because it allows a linear signal estimate, separate from noise and other nonlinearities, to enter into both the phase and magnitude descriptions of sensor dynamic response.

The least-squares optimization minimizes the sum

$$R_u = \frac{1}{N}\sum_{i=1}^{N} [A \sin(2\pi f t_i) + B \cos(2\pi f t_i) + C - u_i]^2 \quad (7)$$

4. As evident in Figure 5, it is unlikely that a given zero crossing will occur at the exact instant the tracker is sampled.

by solving the set of equations

$$\frac{\partial R_u}{\partial A} = \frac{\partial R_u}{\partial B} = \frac{\partial R_u}{\partial C} = 0 \quad (8)$$

for the sinusoidal model coefficients A, B, and C. R_u is recognizable as the mean squared difference or residual error between u_i, the actual data sampled at time t_i, and $A \sin(2\pi f t_i) + B \cos(2\pi f t_i) + C$, the sinusoidal model fitted for $i = 1, \ldots, N$. Coefficient C accounts for any constant offset that may be introduced in the encoder or tracker output during set up of the test. u_i in Eq. (7) represents either the tracker (ϕ_c, ϕ_q, or ϕ_e—depending on the report type of interest) or encoder (θ) angle from an uninterrupted swing arm data record. Swing arm frequency, f, necessary for the calculation of A and B, is determined beforehand from the average time interval between consecutive positive- and negative-going encoder zero crossings.

The magnitude and phase of the resulting sinusoidal model are given, respectively, by

$$M_u = \sqrt{A^2 + B^2} \quad (9)$$

and

$$\Phi_u = \tan^{-1}\left(\frac{A}{B}\right) \quad (10)$$

Thus, the ratio of tracker model over encoder model magnitudes,

$$G_{\phi\theta} = \frac{M_\phi}{M_\theta} \quad (11)$$

provides the linear gain of tracker output (ϕ) with respect to the "true" swing arm motion as measured by the encoder (θ). The difference between the tracker and encoder model phases adjusted for frequency,

$$\delta_{\phi\theta} = \frac{1}{2\pi f}(\Phi_\phi - \Phi_\theta) \quad (12)$$

gives the tracker's internal processing delay (i.e., latency) in reporting the "true" swing arm trajectory. Incorporated into $\delta_{\phi\theta}$ is the correction for bias in time stamp placement, Δ, from Eq. (4).

By accounting for the portion of the sampled sensor record (u_i, $i = 1, \ldots, N$) not predicted by the sinusoidal

model, R_u, the mean square of differences from Eq. (7), provides a measure of sensor noise and nonlinearity, as well as nonsinusoidal swing arm motion components. Normalizing RMS residual error, $\sqrt{R_u}$, by the RMS magnitude of the raw sensor data record yields the noise-to-signal ratio (NSR):

$$n_u = \sqrt{\frac{R_u}{\frac{1}{N}\sum_{i=1}^{N} u_i^2}} \tag{13}$$

Since the same nonsinusoidal swing arm motion components underlie both the tracker and encoder records, the ratio of tracker to encoder NSRs,

$$v_{\phi\theta} = \frac{n_\phi}{n_\theta} \tag{14}$$

indicates how much noise is added (or attenuated) by a particular tracker.

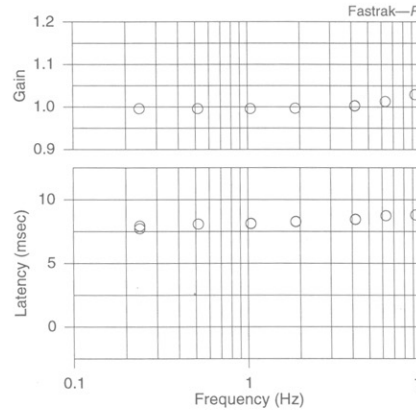

Figure 6. *Fastrak ϕ_c (○) gain and latency for report type P.*

6 Results and Discussion

The testbed and methods described above were employed to examine the dynamic response of Ascension Technology Corp.'s Flock of Birds (software version 3.39) and Polhemus, Inc.'s Fastrak (software version 101.06) in single receiver configurations. One unit of each manufacturer's device was tested.

Tracker gain, $G_{\phi\theta}$, and latency, $\delta_{\phi\theta}$, are plotted for the Fastrak in Figures 6 through 8 and for the Flock of Birds in Figures 9 through 11. Hollow symbols are used for Cartesian-based (ϕ_c) gain and latency calculations; filled symbols are used for quaternion (ϕ_q) and Euler angle (ϕ_e) calculations. When displacement and orientation were reported concurrently by the tracker (*PQ* and *PE* report types), two data points, one hollow and one filled, are plotted per uninterrupted data record. Otherwise (*P* data report type), one point is plotted per fitted test segment.

Report type—either *P*, *PQ*, or *PE*—is seen to have little effect on ϕ_c gain. For the Flock of Birds, however, a difference in gain between orientation (either ϕ_e or ϕ_q) and concurrent Cartesian position output (ϕ_c) is evident in Figures 10 and 11. No difference is seen between Fastrak position and orientation gains. While both devices have uniform gain in the 0.25 to 2 Hz interval, all Fastrak output and Flock of Birds' ϕ_c output show a slight increase in $G_{\phi\theta}$ magnitude at 4 Hz and above. The Flock

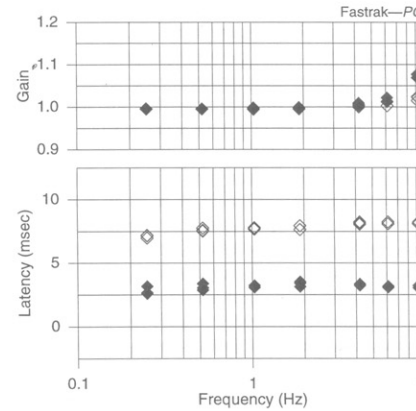

Figure 7. *Fastrak ϕ_c (◇) and ϕ_q (◆) gain and latency for report type PQ.*

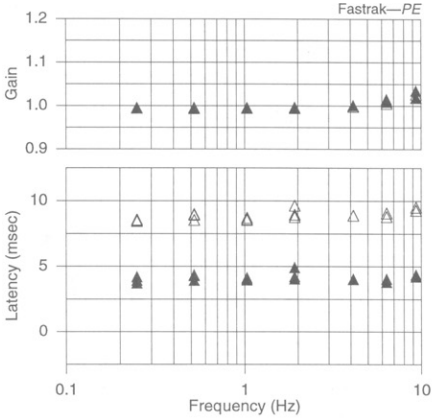

Figure 8. *Fastrak ϕ_c (△) and ϕ_e (▲) gain and latency for report type PE.*

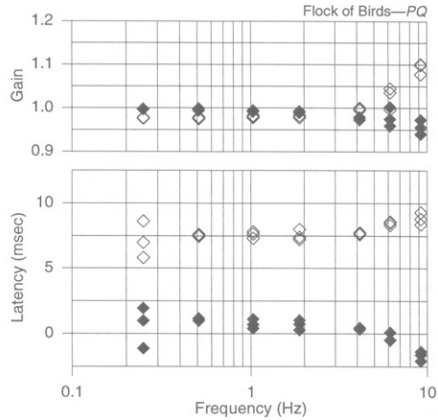

Figure 10. *Flock of Birds ϕ_c (◇) and ϕ_q (◆) gain and latency for report type PQ.*

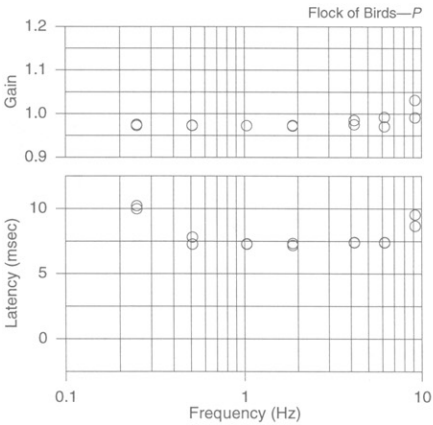

Figure 9. *Flock of Birds ϕ_c (○) gain and latency for report type P.*

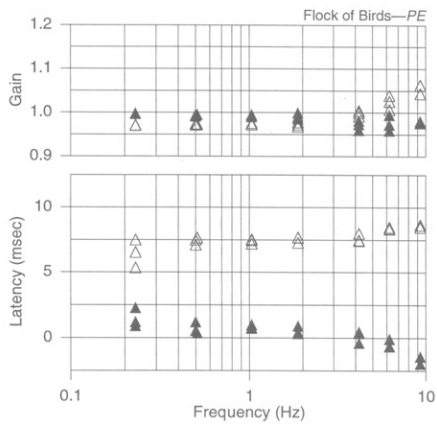

Figure 11. *Flock of Birds ϕ_c (△) and ϕ_e (▲) gain and latency for report type PE.*

of Birds orientation (ϕ_e and ϕ_q) data on the other hand show decreasing gain at the higher frequencies.

Report type does significantly affect tracker latency. Fastrak ϕ_c latencies remain fairly uniform between 8 to 9 msec over the entire 0.25 to 9 Hz range. Orientation components—quaternions or Euler angles leading to ϕ_q or ϕ_e—which arrive in the same data packets as the ϕ_c displacement components, have noticeably shorter laten-

cies of 3 to 4 msec. The Flock of Bird exhibits a similar difference between position and orientation output: 7 to 8 msec and 0 to 2 msec, respectively, over the 0.5 to 4 Hz band, with the tracker orientation output beginning to *lead* the encoder report (i.e., actually having negative latencies) at 6 and 9 Hz.

The heightened variability in latency estimates from adjacent data segments in Figures 9, 10, and 11, especially at the lowest and highest frequencies tested, lends support to the observation from Figure 5 that position and orientation measurements from the Flock of Birds are noisier than those from the Fastrak.[5] Since input swing arm NSRs were similar between tests for the two trackers, the plots of $v_{\phi\theta}$ in Figures 12 and 13 confirm that Flock of Birds noise levels were roughly two times higher in the 0.25 to 2.0 Hz band and three to six times higher from 4 to 9 Hz.[6] Furthermore, it should be noted from Figures 12 and 13 that, in the configurations tested, noise for both trackers increase significantly for swing arm inputs above 2 Hz.

The uniformity of Fastrak latency across frequency indicates that magnetic field transduction and signal processing internal to the tracker result in pure transport delays. Flock of Birds latencies, however, change with frequency, implying that other tracker dynamics may come into play—e.g., lead compensation might account both for decreased, or even negative, latencies at higher frequencies and the greater noise.

The difference observed here between displacement and orientation latencies may account for discrepancies in some of the values previously reported by others. For example, the experimental method described by Krieg (1993), and used by Polhemus, Inc. to confirm the Fastrak's advertised 4 msec latency specification, relied on azimuth measurements from *Euler angle* output. Mine (1993), on the other hand, employed *Cartesian displacement* output with a "minimum acceptable level" of low-

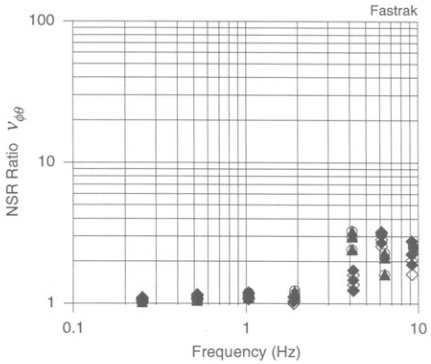

Figure 12. *Fastrak NSR ratio for P, PQ, and PE report types. Symbols are the same as in Figures 6 through 11.*

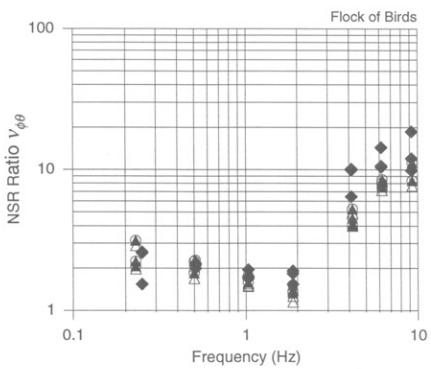

Figure 13. *Flock of Birds NSR ratio for P, PQ, and PE report types. Symbols are the same as in Figures 6 through 11.*

5. The heightened variability of the Flock of Birds made latency estimates based on zero crossings unreliable, which led us to abandon our original data analysis method in favor of the current model fitting technique.

6. For a rough comparison, the respective owner's manuals specify RMS position and orientation accuracies of 0.1 in. and 0.5° averaged over a 36-in. operating range for the Flock of Birds, and a worst case 0.03 in. and 0.15° within a 30-in. range for the Fastrak.

pass filter lag to arrive at a 10.65 msec estimate of internal Fastrak delay. We conjecture a possible cause for this difference is that reported orientation and displacement components are solved algebraically from readings of a *series* of transmitter magnetic fields that are issued and sampled *sequentially* during each tracker measurement

cycle (Raab, Blood, Steiner, & Jones, 1979). Thus solutions for components most heavily weighted by readings from earlier transmitter fields would have the longest latencies and more recent fields the shortest ones.

7 Conclusions

This paper has described the design and implementation of an experimental testbed and method for quantifying the dynamic response of VE spatial trackers. The dynamic response refers to the amplitude fidelity, latency, and noise or nonlinear response *internal* to the tracker. By *internal* we mean the characteristics attributable only to the construction and operation of the tracker—independent of data transmission from the tracker to the host computer and any application software operating on the host computer. While an important consideration in overall system latency is tracker to host communication, high-speed interfaces available on the Fastrak (IEEE-488 at 100 KBytes/Sec) and the Flock of Birds (RS-485 at 312.5 KBaud) can reduce this to a negligible component.

Our tests of Ascension Technology Corp.'s Flock of Birds and Polhemus Inc.'s Fastrak in single receiver configurations with all optional filtering disabled showed latencies of approximately 7.5 and 8.5 msec, respectively, for Cartesian displacement output. Both devices exhibited markedly shorter latencies for concurrently reported orientation components. As summarized in Table 1, these latencies offer considerable improvement over earlier electromagnetic tracker models by the same manufacturers.

The Fastrak and the Flock of Birds both displayed uniform magnitude fidelity up to at least 2 Hz. The Flock of Birds was demonstrably noisier throughout the 0.25 to 9 Hz frequency band tested.

All of our tests were conducted with the tracker's receiver moving through the same circular arc trajectory centered about the x–z plane less than 24 in. (60 cm) away from its transmitter. We did not formally evaluate any aspect of either tracker's performance at any other location.

TABLE 1. *Summary of Internal Latencies for Electromagnetic Trackers*[a]

Device	Settings	Lag (msec)
Ascension Flock of Birds	Filtering disabled, single receiver	7.5
Polhemus Fastrak	Filtering disabled, single receiver	8.5
Polhemus Tracker	Filtering disabled, single receiver	16
Polhemus Isotrak	Normal mode	23–45
Ascension Bird	Default filter	39–47

[a]All lag values are for Cartesian position when the tracker is set to the "position plus Euler angle" report type. Performance of the Ascension Flock of Birds, Polhemus Fastrak, and Polhemus Tracker is for 1 to 2 Hz input frequencies. For the Polhemus Isotrak and Ascension Bird, in which filtering could not be disabled, the entire range of measured frequency-dependent time lags is given. Lags for the three older trackers (Tracker, Isotrak, Bird) are taken from Adelstein et al. (1992).

Acknowledgments

This work was performed in the Advanced Display and Spatial Perception Laboratory at NASA Ames Research Center. Funding was provided in part by the Virtual Windtunnel project at NASA Ames. We thank Drs. R. Holloway of the University of North Carolina and D. Begault at NASA Ames for comments on the manuscript.

References

Adelstein, B. D., Johnston, E. R., & Ellis, S. R. (1992). A testbed for characterizing dynamic response of virtual environment spatial sensors. In *Proceedings of the fifth annual symposium on user interface software and technology* (pp. 15–22). New York: Association for Computing Machinery.

Bryson, S. (1992). Measurement and calibration of static distortion of position data from 3D trackers. In J. O. Merritt & S. S. Fisher (Eds.), *Stereoscopic displays and applications III, Proceedings SPIE 1669* (pp. 244–255). Bellingham, WA: SPIE.

Bryson, S., & Fisher, S. S. (1990). Defining, modeling and measuring system lag in virtual environments. In J. O. Merritt & S. S. Fisher (Eds.), *Stereoscopic displays and applications III, Proceedings SPIE 1256* (pp. 98–109). Bellingham, WA: SPIE.

Burdea, G. C., Dunn, S. M., Immendorf, C. H., & Mallik, M. (1991). Real-time sensing of tooth position for dental digital subtraction radiography. *IEEE Transactions on Biomedical Engineering, BME-38*(4), 366–378.

Delta Tau Data Systems. (1989, September). *Smart Motion Control Card (SMCC): Instructions/operations manual.* Canoga Park, CA.

Durlach, N. I., & Mavor, A. S. (Eds.). (1995). *Virtual reality: scientific and technological challenges* (p. 193). Washington, DC: National Academy Press.

Harwin, W. (1991). *Computer recognition of the unconstrained and intentional head gestures of physically disabled people.* Unpublished doctoral dissertation, Department of Engineering, Trinity College, University of Cambridge, Cambridge, England.

Held, R., & Durlach, N. (1991). Telepresence, time delay and adaptation. In S. R. Ellis, M. K. Kaiser, & A. J. Grunwald (Eds.), *Pictorial communication in real and virtual environments* (pp. 232–246). London: Taylor and Francis.

Held, R., Efsathiou, A., & Greene, M. (1966). Adaptation to displaced and delayed visual feedback from the hand. *Journal of Experimental Psychology, 72*(6), 887–891.

Hirose, M., Kijima, R., Sato, Y., & Ishii, T. (1990). A study of modification of actual environment by see-through HMD (in Japanese). In *Proceedings of the Sixth Symposium on Human Interface,* Tokyo.

Holloway, R. (1995). Registration error analysis for augmented reality. Technical Report TR95-001. Chapel Hill, NC: University of North Carolina, Department of Computer Science.

Jones, H. R. (1993). Polhemus, Inc., Colchester, VT. Personal communication, August 5, 1993.

Krieg, J. C. (1993, May). *Testing for tracker latency and update rate.* Paper presented at Meckler Virtual Reality '93, Fourth Annual Virtual Reality Conference and Exhibition, San Jose, CA.

Liang, J., Shaw, C., & Green, M. (1991). On temporal-spatial realism in the virtual reality environment. In *Proceedings of the fourth annual symposium on user interface software and technology* (pp. 19–25). New York: Association for Computing Machinery.

Meyer, K., Applewhite, H. L., & Biocca, F. A. (1992). A survey of position trackers. *Presence: Teleoperators and Virtual Environments, 1*(2), 173–200.

Mine, M. R. (1993). Characterization of end-to-end delays in head-mounted display systems. Technical Report TR93-001. Chapel Hill, NC: University of North Carolina, Department of Computer Science.

Murry, H. F. (1993, March). *Putting motion tracking to work.* Paper presented at the VR Systems Conference, New York.

Polhemus Navigation Sciences. (1992). *Technical note: Latency—3SPACE® Magnetrak®, Tracker, Digitizer and Isotrak®.* Colchester, VT.

Raab, F. H., Blood, E. O., Steiner, T. O., & Jones, H. R. (1979). Magnetic position and orientation tracking system. *IEEE Transactions on Aerospace Electronics Systems, AES-15*(5), 709–718.

Sheridan, T. B., & Ferrell, W. R. (1963). Remote manipulative control with transmission delay. *IEEE Transactions on Human Factors in Electronics, HFE-4*(1), 25–29.

Sheridan, T. B., & Furness, T. A., III (Eds.). (1992). Spotlight on: Simulator sickness, *Presence: Teleoperators and Virtual Environments, 1*(3), 295–362.

Wloka, M. (1995). Lag in multiprocessor virtual reality. *Presence: Teleoperators and Virtual Environments, 4*(1), 50–63.