

A MATLAB-based eye tracking control system using non-invasive helmet head restraint in the macaque

Paolo De Luna¹, Mohamed Faiz Bin Mohamed Mustafar¹, Gregor Rainer*

Visual Cognition Laboratory, Department of Medicine, University of Fribourg, Chemin du Musée 5, Fribourg CH-1700, Switzerland

H I G H L I G H T S

- We have developed a MATLAB based eye tracking control system with excellent timing performance.
 - The experiment was performed using non-invasive head restraint.
 - The system performance reported adequate stability in both short and long timescales.
-

Background: Tracking eye position is vital for behavioral and neurophysiological investigations in systems and cognitive neuroscience. Infrared camera systems which are now available can be used for eye tracking without the need to surgically implant magnetic search coils. These systems are generally employed using rigid head fixation in monkeys, which maintains the eye in a constant position and facilitates eye tracking.

New method: We investigate the use of non-rigid head fixation using a helmet that constrains only general head orientation and allows some freedom of movement. We present a MATLAB software solution to gather and process eye position data, present visual stimuli, interact with various devices, provide experimenter feedback and store data for offline analysis.

Comparison with existing method: Our software solution achieves excellent timing performance due to the use of data streaming, instead of the traditionally employed data storage mode for processing analog eye position data.

Results: We present behavioral data from two monkeys, demonstrating that adequate performance levels can be achieved on a simple fixation paradigm and show how performance depends on parameters such as fixation window size. Our findings suggest that non-rigid head restraint can be employed for behavioral training and testing on a variety of gaze-dependent visual paradigms, reducing the need for rigid head restraint systems for some applications.

Conclusion: While developed for macaque monkey, our system of course can work equally well for applications in human eye tracking where head constraint is undesirable.

1. Introduction

The scleral search coil (Judge et al., 1980; Robinson, 1963) has been used extensively since its introduction in the 1960s for monitoring of eye position in the context of studies of visually-based behavior in macaque monkeys. Its use requires a surgical intervention, where an insulated wire coil is attached to the sclera by glue or sutures, and the wire is linked to a connector implanted

on the animal's head. Such implants have a limited life span and often require repair or replacement, potentially involving additional interventions. In the late 1990s optical eye tracking systems became available; they track the position of the pupil illuminated by an appropriate light source using an infrared sensitive camera. Infrared eye tracking systems are non-invasive and do not require any surgical procedures, and are therefore increasingly used in behavioral and neurophysiological studies. It is important to note that while the eye tracking itself is non-invasive, it is generally used in conjunction with a head-post that is implanted on the skull of the monkey and attached to the experimental setup by a rigid connection. The head-post ensures that the position of the head and particularly the eye remains fixed during the behavioral

* Corresponding author. Tel.: +41 26 300 8689; fax: +41 26 300 9734.

E-mail address: gregor.rainer@unifr.ch (G. Rainer).

¹ These authors contributed equally to this work.

session. An alternative to this might be a non-rigid head restraining system, which maintains the head in an approximately constant position without a rigid connection to the setup, thus permitting some movements. Indeed, eye tracking in human subjects often employs non-rigid head restraint using a chin-rest and bite bar. Eye-tracking using head-mounted eye trackers, as it is also used in human subjects, are not easily tolerated by monkey subjects. Non-rigid head restraint would allow the training of monkeys on gaze dependent behavioral paradigms without the need of surgical interventions such that these could be postponed to after the training is completed. A recent publication has described a thermo-deformable plastic helmet that can be easily adapted to the shape of individual monkeys' head (Machado and Nelson, 2011). In this study, we demonstrate that eye tracking of sufficient quality for many applications is possible using such a helmet-based restraint system, and describe performance characteristics and limitations of eye tracking under these conditions.

For controlling the sequence of events during a visual task, computer software is necessary to gather eye position data from the eye tracker, present visual stimuli, and deliver rewards at appropriate times while at the same time provide online feedback to the experimenter. There are numerous proprietary as well as open-source software packages available for this purpose. For ease of programming and adaptation of code, solutions written in a widely used high-level interpreted programming language such as MATLAB are preferable, particularly since Psychophysics toolbox, a widely used package for visual stimulus display is also available in this programming environment. Using MATLAB to perform online control is inherently somewhat problematic, because this programming language was not designed for real-time control. Nevertheless, we are aware of two software solutions that employ MATLAB together with Psychophysics Toolbox for the control of gaze-dependent behavior (Asaad and Eskandar, 2008; Asaad et al., 2013; Meyer and Constantinidis, 2005). Our own software implementation is based on these two systems. We have introduced several new elements that enhance performance and reliability, making a particular effort to produce compact code that can be relatively accessible for beginners.

2. Materials and methods

2.1. Subjects

Two 11 year-old male macaque monkeys (*Macaca fascicularis*), referred to as Monkey C (6.8 ± 0.2 kg, mean \pm standard deviation) and Monkey D (8.2 ± 0.1 kg), participated in the experiments. Animals were maintained on a diet of fresh fruit, vegetables and monkey chow with water available ad libitum. During the experiments, for each successful fixation the monkeys were rewarded with a 45 mg purified dustless pellet (Banana flavor 5TUQ tab, Test-Diet, 1050 Progress Drive, Richmond IN 48384, USA), delivered via a metal tube. Monkeys performed an average of 183 ± 11 (S.E.M.) trials per experimental session, and had no head-posts or other implants. Head restraint was achieved non-invasively using thermoplastic helmets (Uni-frame Thermoplastics MTAPUI2232, Civco Medical Solutions) adjusted to each individual monkey (Machado and Nelson, 2011). The thermoplastic material was made flexible by submerging the sheet in a 80°C water bath for 2 min. During the next 2 min, before the material regained its hardness, the helmet was pre-shaped on a toy doll head, finely adjusted by hand, and two openings for eyes and mouth were cut with scissors. Final adjustment of the helmet to each monkey did not require sedation of the animals, and was performed iteratively such that a good fit that permitted chewing movements was obtained after about four iterations. Importantly, the helmet fit was adjusted to give monkeys

a certain amount of freedom in head position, both laterally as well as vertically, to allow for chewing of rewards as well as minimizing discomfort. All animal procedures were in full compliance with Swiss and European Union animal experimental regulations and approved by the Fribourg cantonal veterinary authorities.

2.2. Setup

The monkey was seated in a primate chair (custom-made at University of Fribourg), on a wheeled trolley at the base which could be locked in a fixed position inside a wooden recording box as shown in Fig. 1A. An infra-red camera (ISCAN ETL-200 system, x2 lens attached) was mounted at an elevation angle of 27° at a distance of 65 cm from the animal. The infrared light source was attached to the left side of the camera objective lens, and tilted for optimal illumination of the left eye (Fig. 1B). The inside of the box was covered with matte black plastic panels. Visual targets were presented on a cathode ray tube (Compaq P900 19-inch CRT monitor at resolution of 1280×1024 pixels, with the refresh rate of 75 Hz) monitor positioned 65 cm directly in front of the animal. We employed black squares of 0.5° diameter (20×20 pixels), arranged along a circle of radius of 6° regularly spaced 45° around the ninth square in the center of the circle (Fig. 1C). Targets were presented on a background of intermediate intensity (50% grayscale), corresponding to a luminance of 13 cd/m^2 , measured using a Minolta TVCA-II color analyzer. Alternatively, six red 0.8° diameter light emitting diodes (LED) on a black panel mounted at a distance of 30 cm from the animal could also serve as visual targets (Fig. 1D). During the behavioral sessions, auditory white noise was delivered via a speaker inside the box to mask external distractions. The speaker was also used to deliver a 1 kHz pure tone for 1 s as auditory feedback for each correct trial.

2.3. Hardware configuration

We used an infrared eye tracking system (ISCAN ETL-200). In detail, the camera was connected via a PCI acquisition board (ISCAN RK8X6PCI-0) to a computer running the eye-track acquisition software (DQW version 1.20N). The computer was equipped with a 2.4 GHz Intel Core2Duo processor, 4 GB of RAM and a 3-channel analog output card. The Eye-Tracking System's analog output was interfaced via a single PCI-6221 acquisition board (National Instruments) connected to a BNC-2110 connector block (National Instruments) to a second PC that hosted the Experiment Control System (Fig. 2). We used two analog channels to relay horizontal and vertical eye position. The ISCAN system was configured to track eye position using the difference between the pupil and corneal reflection, as is commonly done in eye tracking systems without rigid head restraint. It is crucial to use this difference signal for non-invasive eye tracking, as it is less sensitive to head translations than the raw pupil signal.

The Experiment Control System (HP Compaq 8200 Elite, Windows 7) was equipped with a 3.3 GHz Intel i5-2500 processor and 4 GB of memory. An NVidia GeForce 7300GT video card was installed for managing two displays at a resolution up to 1280×1024 pixels. The subject display viewed by the animal was duplicated using a VGA splitter to be visible also to the experimenter. In addition, an experimenter display was used to provide online feedback about the animal's behavior to the experimenter. The "Extend these displays" option of Windows 7 was used to simultaneously manage experimenter and subject displays. The PCI-6221 board was also used to illuminate visual target LEDs and to deliver reward pellets (ENV-203-45 pellet dispenser, Med Associates, St. Albans, VT 05478, USA). An overview of the connections between the setup components is shown in Fig. 2. To optimize performance of the ISCAN and experimental control system PCs, all

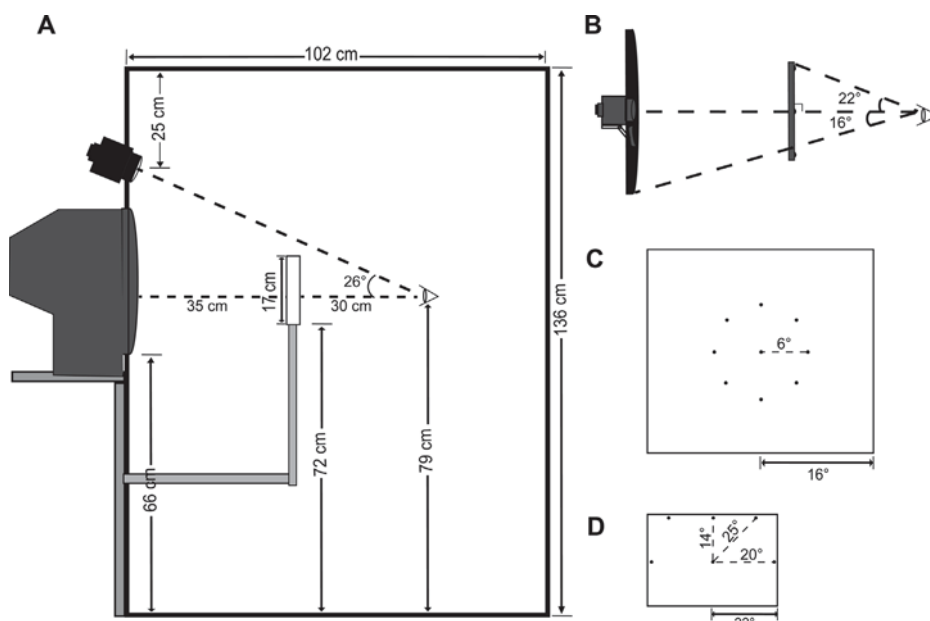


Fig. 1. Behavioral experiment setup. (A) Side view of the recording box where behavioral data was recorded. Monkey's head position is marked by an eye symbol. On the left the CRT screen and, on top of it, the infra-red camera are drawn; the white rectangle in the middle is the black LED panel. (B) Top view of recording box. LED panel and CRT screen horizontal size were 22° and 16° respectively. The CRT monitor, the LED panel and the infra-red camera were centered on monkey's head position (eye symbol). (C) Positions of visual stimuli on CRT monitor. Eight black square targets along a 6° circle surround a ninth black squared central fixation point. Each stimulus is about 0.5° (20×20 pixels). (D) Positions of visual stimuli on LED panel arranged on two rows of three elements. Each stimulus was a red LED of approximately 0.8° diameter (3 mm).

non-critical processes including interface-, security-, network- and devices-related ones were terminated and neither computer was connected to any network.

2.4. MATLAB configuration

All the data were acquired by and stored into the Experiment Control System using a custom-made script run in MATLAB (version R2010a, The Mathworks Inc.). The script used functions from the MATLAB Data Acquisition Toolbox (DAQ, version 2.16) and Psychophysics Toolbox (PTB) version 3.0.11 (Brainard, 1997; Kleiner et al., 2007; Pelli, 1997). The script presents a series of randomly selected single visual targets and provides reward to the animal if that target is fixated for a predetermined period of time. This type

of script is routinely employed for the calibration of eye position. Three fragments of the source code are provided below, illustrating important elements of DAQ configuration, data buffering and online behavioral control. The full script can be obtained from our website (<http://www.unifr.ch/inph/vclab/home/internal/eye-tracking-software>).

2.4.1. DAQ configuration

Ideally, one wants to stream the most recent sample acquired by the board as soon as possible for online control while at the same time storing samples on the board to be picked up at the end of each behavioral trial. However, simultaneous storing and streaming of data through the DAQ toolbox is associated with severe performance impairments (see Section 3). For this reason, our solution focuses only on streaming and does not rely on storing data on the board. The following code snippet illustrates our solution for analog data acquisition.

```
ai = analoginput('nidaq','Dev1');
addchannel(ai,[0 1]);
ai.SampleRate = 1000;
ai.SamplesPerTrigger = 2;
ai.TriggerRepeat = 0;
start(ai);

startLoop = GetSecs;
while (GetSecs - startLoop < time4fix),
    thisSample = getsample(ai);
    ...
end
```

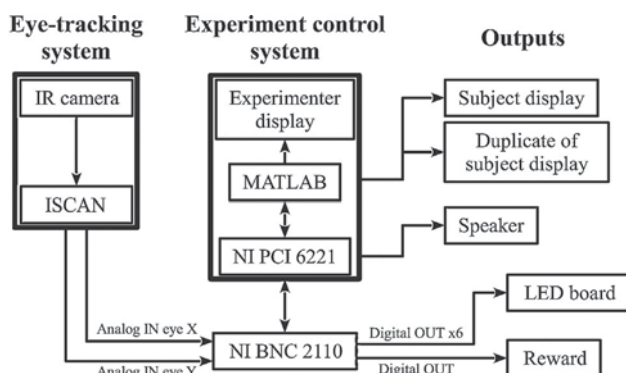


Fig. 2. Overview of connections of hardware components. Eye position signals were computed by ISCAN software and sent via a NI BNC 2110 connector block to a NI BNC 6221 acquisition board to the Experiment Control system. The gaze position was monitored with MATLAB during presentation of visual stimuli either on a CRT monitor (and to a mirror display visible to the experimenter), or on an LED panel via the NI PCI 6221 digital output channels. MATLAB also presented the last acquired gaze position to the experimenter. Finally one digital output channel of the NI PCI 6221 was used for controlling an electric valve for reward delivery.

The board is configured to store only two samples, the minimum permitted by MATLAB. After starting data logging, only the first two samples are stored on the board, whereas all the subsequent samples are not stored and thus cannot be recovered later on by a *getdata* function call. However, subsequent samples can be pre-viewed using the *getsample* function. In this case, the most recently

acquired sample is made available to MATLAB without compromising performance. Our approach thus optimizes data streaming to MATLAB. As a result, only the data that are actually streamed to MATLAB using `getsample` for online control can be retained for subsequent data analysis.

2.4.2. Circular buffer

Since infrared eye tracking data can be noisy, it is useful to include some temporal averaging to reduce the variability in the acquired data. For this purpose we use a First-In First-Out (FIFO) circular buffer, with the parameter `buffer_length` that determines over how many subsequent samples, acquired using `getsample`, the averaging process takes place.

```
buffer_length = 25;
Buffer = zeros(buffer_length,2);
function [Buffer,eyeX,eyeY,sampleInfo] = bufferizeSamples(ai,Buffer,Gains,Offsets)
    % Put together the information about the last sample
    timestamp = GetSecs;
    thisSample = getsample(ai);
    sampleInfo = [thisSample timestamp];
    % Update the buffer with the last data point and take a new average point
    Buffer = [Buffer(2:end,:); thisSample];
    avgBuff = mean(Buffer,1);
    % Convert volts to degrees
    eyeX = (avgBuff(1) - Offsets(1)) * Gains(1);
    eyeY = (avgBuff(2) - Offsets(2)) * Gains(2);
end
```

In our MATLAB script, this function is called several hundred times each second, and timestamped using the PTB `GetSecs` function. The last sample replaces the first one in the buffer and a new gaze position is computed by converting the average input signal from voltage to degrees of visual angle, using ad hoc gains and offsets for both the axes.

2.4.3. Main loop

The following code fragment illustrates the main loop of the behavioral control system, where the system checks for whether the eye position is within a radius of the target position. It uses the circular buffer and data streaming elements described above.

```
startLoop = GetSecs;
while GetSecs - startLoop < time4fix
    % 1) Get and store a new sample
    [Buffer,eyeX,eyeY,sampleInfo] = bufferizeSamples(ai,Buffer,Gains,Offsets);
    Data(trial).EyeDataX(index) = sampleInfo(1);
    Data(trial).EyeDataY(index) = sampleInfo(2);
    Data(trial).Time(index) = sampleInfo(3);
    index = index + 1;
    % 2) Update eye position on experimenter's display
    set(eyePosH,'XData',eyeX,'YData',eyeY);
    drawnow;
    % 3) Check for fixation
    if sqrt((TargetX - eyeX)^2 + (TargetY - eyeY)^2) <= FixWinRadius
        Data(trial).FixStart = GetSecs;
        break;
    end
end
```

The loop consists of three components: (1) using the circular buffer, a new gaze position is evaluated as previously described, and the last sample and corresponding timestamp are stored in a trial-wise manner in pre-allocated matrices; it has to be noted that we also stored the length of the buffer, the gains and the offsets in order to reconstruct all the gaze positions from the raw data; (2) the experimenter's display is updated with the new gaze position; (3) if the gaze position is inside the fixation window the loop is interrupted; in that case a similar loop will check if the subject is keeping the fixation. We refer to this code fragment with all of its three parts as the "Main loop".

2.5. Eye tracking

We used a simple fixation task to calibrate the eye position signal, and to test the system performance as a function of several parameters. As described above, we employed a set of nine targets arranged in a circular fashion on a CRT monitor, as well as six LED targets positioned on a panel mounted in front of the animal. Monkeys needed to fixate a single target for a brief period to obtain a reward pellet. We varied both the length of the circular buffer (values: 1, 10, 15, 20 and 25 samples) and the radius of the fixation window (values: 1.5° and 2° for the CRT screen and 2°, 2.5° and 3.5° for the LED panel). Larger values were chosen for the LED panel fixation window because targets were more eccentric in that case. During blocks of 50 trials, a random combination of fixation window radius and buffer length was chosen.

Each session started with the monkey outside the recording box without the helmet. Then the helmet was fixed on the primate chair and the monkey was moved in the box. At the beginning of each session the camera was centered on the monkey's left eye and about 20 trials were used to adjust the gains and the offsets for the horizontal and vertical directions using a 25-sample buffer. Only one set of visual stimuli (CRT screen or LED panel) was presented during the same session.

During each trial, after the appearance of the target on the panel (LED) or CRT screen (square dot), the monkey had to start the fixation within 2 s and hold it for 200 ms to obtain a reward pellet with simultaneous auditory feedback. The inter-trial interval was 7 s. At the end of each block, the monkey was moved again outside the box and the helmet was removed; the inter-block interval was about 2 min.

3. Results

We ran extensive tests on our system, and observed that the choice of MATLAB DAQ acquisition parameters had a very large impact on system performance, with non-optimal parameters introducing substantial lags between acquisitions of individual samples as well as timing inaccuracies.

3.1. Timing considerations

We first compared the performance of DAQ storing mode (parameter setting `ai.SamplesPerTrigger = Inf;`) that has been previously proposed in several reports to DAQ streaming mode without storing (parameter `ai.SamplesPerTrigger = 2;` see Section 2). In storing mode the board is collecting data and making them available to MATLAB at the same time, whereas in streaming mode all the samples are discarded by the board and can only be previewed with `getsample` function calls. For this test, we used the Main loop described above, and switched from data storing to data streaming at time $t = 0$ ms. A synthesized sine wave at 50 Hz produced by a signal generator (GW Instek SFG 2004) was used as an input in this case instead of an eye position signal. Importantly, the `getsample` function was used in both modes to obtain a data sample for online eye movement control. The results are shown in Fig. 3A. Clearly, calling `getsample` in data storing mode is associated with unpredictable lags between the input signal and the acquired sample. These lags are furthermore variable, and we found them to be in range between 10 and 40 ms. In addition, previewing data using `getsample` in data storing mode produced widespread sample repetitions. Individual samples could be repeated up to one thousand times or more, because at the time of the `getsample` call the board is busy storing data to memory and thus returns the same previously acquired sample. Using data streaming mode these problems disappear, and samples are now lying very close to the input

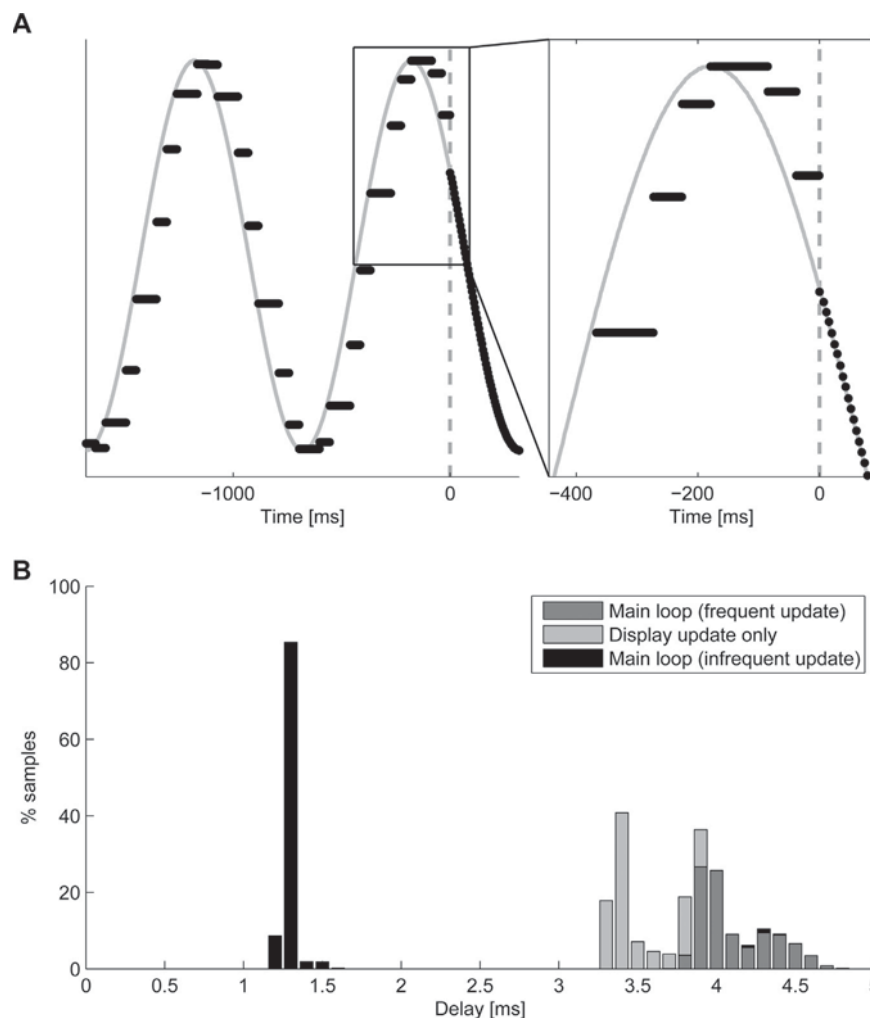


Fig. 3. (A) Two cycles of a synthesized 1 Hz sinusoidal wave are shown in the left panel. At time $t=0$ the acquisition board stopped simultaneous data storing and streaming to MATLAB workspace, and discarded all new collected samples from its memory. All presented data points were collected in MATLAB calling the *getsample* function. In the right panel, a zoomed in view of time $t=0$ is shown. During combined storing/streaming mode, samples can be repeated up to 1000 times and they are acquired after a variable delay of about 50 ms. Instead, during streaming only mode, all samples lie closely to the input signal with no redundancy or time lag. (B) Stacked bars show time lags between two consecutive samples. Note that the signal sampling rate was 1 kHz and the board's memory size was set to 2 samples. Running the Main loop with display update at each cycle (dark gray bars) generated a lag of about 4 ms of which about 3.5 ms were needed to the display update itself (light gray bars). Updating the screen every 100 ms (black bars) lowered the average lag to about 1.3 ms, of which 1 ms was a user-defined pause to be sure to acquire a new sample. Note that when the experimenter display update occurred, an average lag of about 4.3 ms was observed.

signal, certainly without unpredictable lags or repetitions. For these reasons, we abandoned data storage mode and relied only on data streaming. It follows that at the end of each trial, *getdata* cannot be used to recover all of the samples, and only the samples gathered using *getsample* are actually available to be stored for offline analysis.

To acquire and display one eye position sample, our system required a delay of about 4 ms, as shown in Fig. 3B for the histogram corresponding to “Main loop (frequent update)”. In this case, the experimenter display is updated for every acquired sample. The slowest component of the loop is accounted for by the experimenter's display update, which requires an average delay of about 3.5 ms for each call to MATLAB's function *drawnow*. Running the Main loop with infrequent screen update (for example once every 100 ms), it is possible to achieve a performance of under 1.5 ms for most samples. In this case, only the few samples that are actually drawn on the screen exhibit a delay of about 4.5 ms.

Using frequent screen update mode, which displays all acquired samples on the experimenter screen, our software solution exhibits a delay of under 5 ms, which we consider suitable for monitoring eye position signals acquired with an infrared camera. We

therefore used this configuration in the subsequent experiments detailed below.

3.2. Circular buffering

It is useful to employ data smoothing to reduce spatial variability of gaze positions and attenuate technical noise coming from the eye tracker. This can be achieved by averaging over the last n acquired samples stored in a (FIFO) circular buffer. A consequence of signal averaging is that analog signals are also temporally delayed, and this drawback becomes relevant for studying transient and rapid phenomena, like for example saccadic eye movements. We used a synthesized 0.1 Hz square wave (GW Instek SFG 2004) to approximate a saccadic eye movement, and examined the temporal delays introduced by circular buffering. For a buffer length of 10 samples, an abrupt change in the analog signal input at time 0 generates a linear ramp lasting about 40 ms in our system, and thus introduces a lag in the detection of the signal transition of at least $\Delta t = 20$ ms (Fig. 4A) Depending on the fixation window size, Δt can be larger but cannot exceed 40 ms for a buffer length of 10. Examining the dependence of the minimum lag Δt on buffer length, we observed

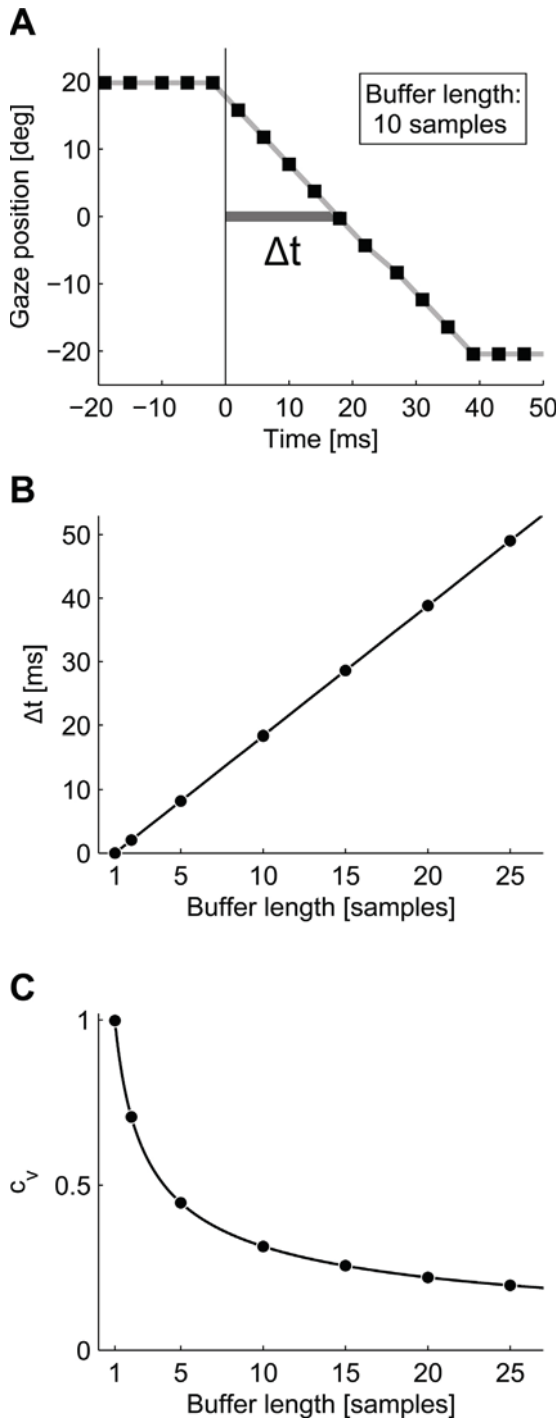


Fig. 4. (A) The effect of using a circular buffer in online applications is shown. The input signal was a synthesized 0.1 Hz square wave. Each square symbol represents the mean value of the last 10 samples stored in a circular buffer (see Section 2). The timeline is centered between the last sample of a stationary period and the first sample of the transition period. We refer to minimum time delay for detecting a saccade as Δt . (B) Increasing buffer length linearly increases Δt . Note that averaging across a buffer with 1 sample is like using raw data. (C) Power decay relationship ($R^2 = 0.99$) between coefficient of variation (c_v) of Gaussian noise wave and buffer length.

a linear relationship, as shown in Fig. 4B. This illustrates that temporal lag grows systematically with buffer length. At the same time, the variability of analog signals is reduced with increasing buffer length. We illustrate this by computing the coefficient of variation (σ/μ) of data (100,000 data-points) sampled from a normal distribution ($\mu = 1, \sigma = 1$) that are smoothed with various buffer lengths

(Fig. 4C). It can be seen that variability quickly decays for buffer lengths up to about 15, and then continues to decline more gradually. This suggests that a circular buffer of at least 15 samples is advisable when there is substantial noise present in the input data.

Taken together, averaging across large circular buffers is optimal for reducing technical variability in infrared eye tracking analog inputs. However, it introduces temporal lags in detectability of transient signal changes, such as those that occur during saccadic eye movements. The parameter choice for circular buffer length depends on a trade-off between these two effects.

3.3. Behavioral data

The gaze positions of two monkeys, C and D, were monitored during a fixation task, while the subjects fixated a target either on a LED panel or on a CRT screen for 200 ms (monkey C: 50 blocks LED panel, 26 blocks CRT screen; monkey D: 51 blocks LED panel, 28 blocks CRT screen). Each block of trials consisted of 50 trials with randomized fixation location and a fixed combination of buffer length and fixation window radius. Data presented here are based on about 4 blocks of trials for each parameter combination. The monkeys were subjected to non-invasive head restraint using an individualized thermoplastic helmet (see Section 2), that allowed up to about 1 cm of lateral and vertical head movement. These small translational movements were compensated using the signal difference between pupil center and corneal reflection and did not affect the performance of the system since the position of the eyes remained within a window suitable for eye tracking. Note that there is a trade-off between the magnitude of permissible translational movements and the spatial accuracy of the eye tracker, because the eye tracker field of view needs to be expanded to allow larger translational movements, reducing spatial resolution. Our chosen parameters represent a compromise between residual permissible movements and eye tracking resolution. Monkeys were not pre-trained on fixation or other tasks, so that they can be considered naïve. The results of the behavioral experiments are shown in Fig. 5. Clearly, system performance is poor without circular buffering (buffer length = 1 sample), where many broken fixation errors occur. As expected, larger fixation windows, as well as longer buffer lengths, led to higher behavioral performance for both the sets of visual stimuli. A two-way analysis of variance (ANOVA) revealed a significant effect of fixation window size (FWS) and buffer length (BL) on behavioral performance ($p < 10^{-9}$) with no interaction between these factors ($p > 0.1$). To obtain high spatial resolution, temporal accuracy has to be sacrificed and vice versa. For example, monkey D performed similarly on the LED panel task for parameter combinations of $FWS = 3.5^\circ/BL = 10$ samples and $FWS = 2^\circ/BL = 25$ samples. Parameter combinations for FWS and BL will likely depend on particular experimental requirements and can be optimized accordingly.

To illustrate the performance of our system, we show horizontal and vertical eye position traces around time of fixation acquisition (Fig. 6A and B) for sixteen correct trials ($FWS = 2^\circ, BL = 15$ samples) taken from the LED panel task for the central target position. The traces show saccadic eye movements made to the target position at around time $t = 0$ ms from a variable initial location mostly in the upper right visual field quadrant. After acquisition of fixation, the eye position remains inside the fixation window of $\pm 2^\circ$. To assess sample-to-sample variability of eye position estimates during fixation, we computed the distance of data samples from the mean eye position during the fixation time period between 0 ms and 200 ms for each trial. This sample-to-sample variability quantifies eye position signal stability at short time scales. We estimate a median value of $0.09 \pm 0.001^\circ$ for both horizontal and vertical dimensions. To assess the stability of our system at a longer timescale corresponding to the course of an entire behavioral session, we examined the

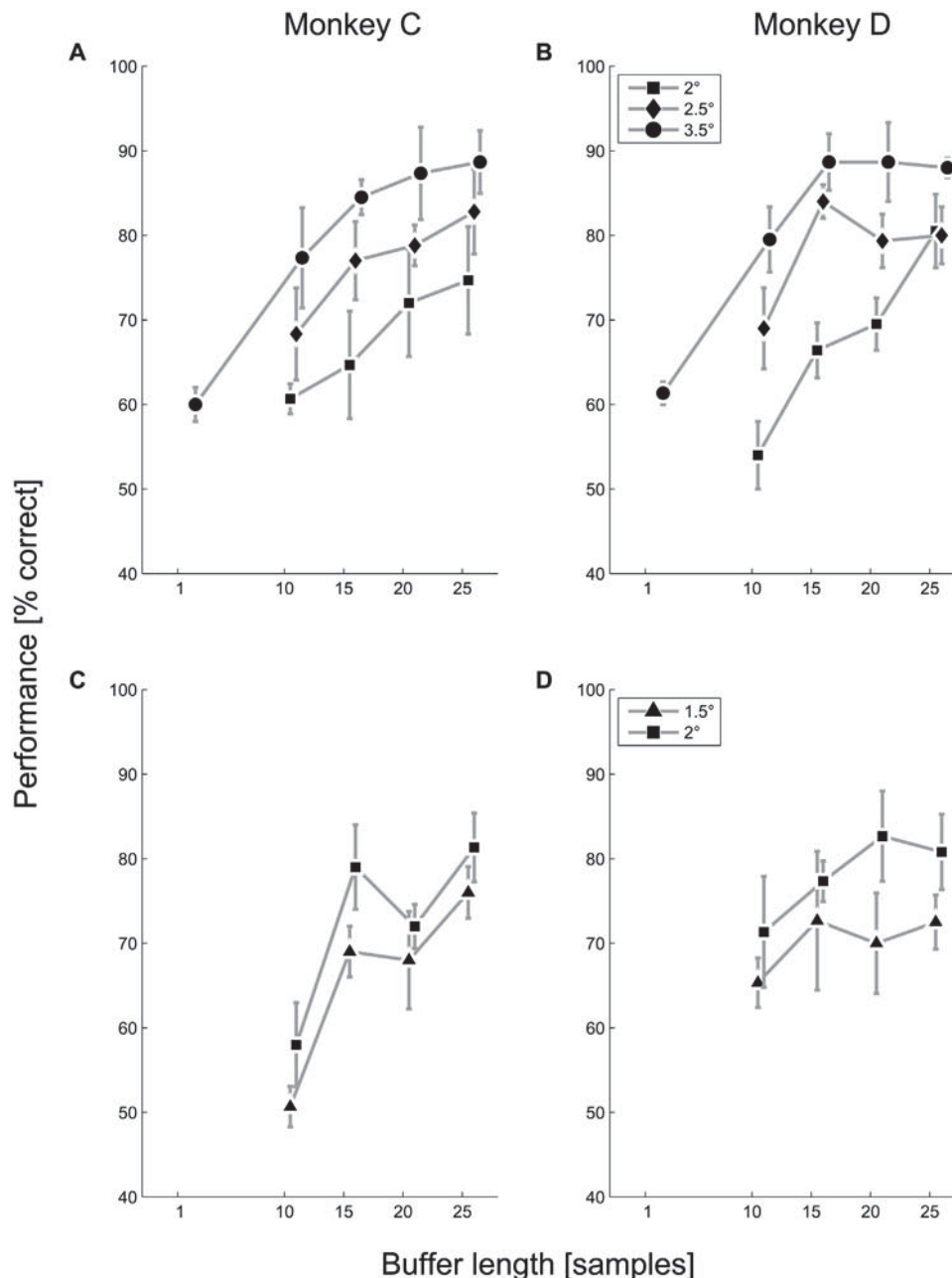


Fig. 5. Performance of target fixation for monkeys C (A and C) and D (B and D) during blocks of 50 trials. Different lines represent different fixation window radii as shown in figures legend. Targets were presented on an LED panel (A and B) or a CRT monitor (C and D) for both animals.

mean gaze position during the fixation periods of subsequent trials for the central target positions only in CRT and LED panel tasks. These data are shown for an example behavioral session lasting about one hour in Fig. 6C and D. A linear regression revealed that both horizontal and vertical eye position drifted only little during the course of this session, as shown by the slope parameter estimates ($m_{\text{horizontal}} = 0.08^\circ/\text{h}$ and $m_{\text{vertical}} = 0.20^\circ/\text{h}$). An analysis of eye position drift across all behavioral sessions with three or more blocks is shown in Fig. 6E. For 73% of the sessions, the drift was less than $1^\circ/\text{h}$ both horizontally and vertically. The remaining sessions exhibited larger drift, which might necessitate manual readjustment of gain and/or offset calibration between blocks of trials. Of interest is also the generally larger variability in gaze position in the vertical compared to the horizontal dimension, which is quite noticeable in Fig. 6C and D. To quantify this effect, we estimated the distance of the within-trial mean fixation locations from the

within-session mean. The trial-to-trial variability was computed as the median value of these distances. Considering the population behavioral data (Fig. 6F), the average horizontal trial-to-trial variability ($0.32 \pm 0.05^\circ$) was indeed smaller than the vertical variability ($0.47 \pm 0.08^\circ$) as assessed with a paired t -test ($p < 0.001$). We attribute this difference to licking movements in anticipation of rewards, that cause preferentially vertical rather than horizontal head displacements and therefore introduce noise mostly in the vertical dimension.

4. Discussion

The current standard method for eye tracking in non-human primates research requires invasive procedures like the implantations of a fixed head-post and a metal wire in the sclera. As

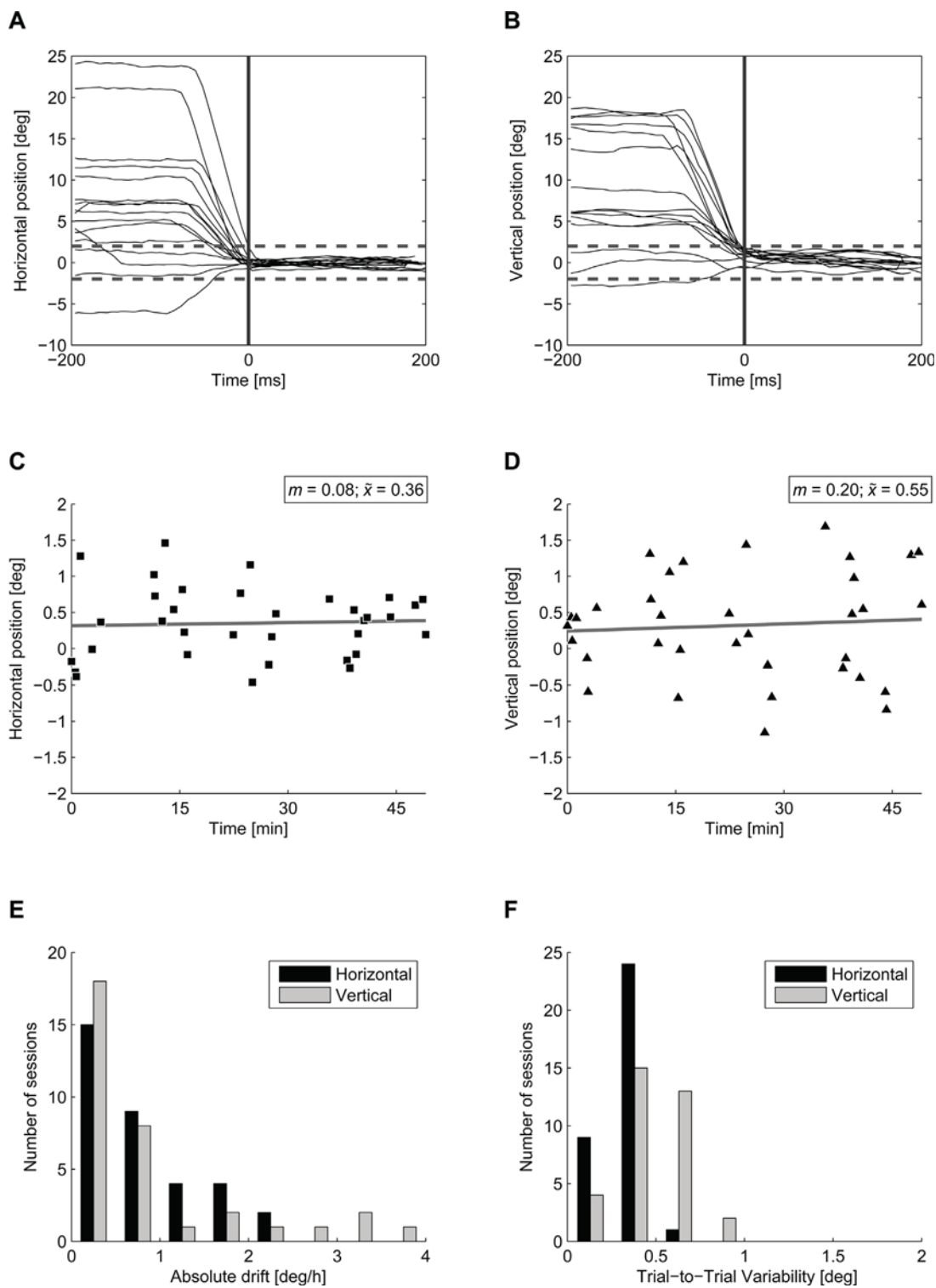


Fig. 6. Sixteen trials are shown (A and B). At time $t=0$ target fixation was acquired. Horizontal dashed lines are the boundaries of the fixation window for central target on the LED panel. (C and D) Each square symbol represents the average gaze position during the fixation time for the central position on the LED panel recorded during an example session, respectively for the horizontal and vertical dimension. Timeline starts at the beginning of the first block; each cluster of points is part of a 50-trials block. For each session with more than three blocks, drift and variability of average gaze positions during fixation periods were systematically measured calculating, respectively, the slope of the linear regression line (gray line) and standard deviation of the mean positions. (E) Distribution of the absolute values of regression lines slopes for both directions. The drift was less than one degree per hour for more than 73% of the sessions in both directions. (F) Distribution of standard deviations of mean gaze positions during fixation periods is shown for both directions. Values are less than 0.5° more often in the horizontal direction (52% of the sessions) than the vertical one (21% of the sessions). The distributions are significantly different ($p < 0.001$).

recently shown, non-invasive head restraint can be achieved by employing a plastic helmet, adjusted to each individual animal that allows limited head movements while constraining the overall orientation of the head (Machado and Nelson, 2011), and an infrared camera-based system for tracking eye position. This approach is however also useful for neurophysiological work, since surgical procedures can be carried out after appropriate behavioral training, reducing the duration that devices need to remain implanted on the animal to achieve the experimental objectives. For the behavioral data reported here, monkeys were rewarded with a 45 mg pellet for correct trials, and made chewing movements during the consumption of these rewards, as well as frequently at times during the task and prior to reward delivery. Nevertheless, this system can function despite these chewing movements, mainly because the infrared eye-tracking device employs the difference signal between the pupil center and the corneal reflection, which is insensitive to chewing as well as translational head movements. We noticed however, that spatial accuracy of our system is greater in the horizontal than in the vertical dimension; an effect that we partially attribute to the more frequent occurrence of vertical compared to horizontal head movements both in the context of chewing as well as general behavior. In addition however, the placement of the camera is non-optimal for tracking vertical eye movements. We chose this position because the space below the monitor, which is generally recommended for positioning the camera, is occupied by experimental devices in our setup. One might consider an above-mounted camera in conjunction with a 45° slanted mirror as an alternative configuration (Kimmel et al., 2012) to improve vertical gaze tracking.

Our behavioral control system is implemented in MATLAB (Mathworks, Natick MA, USA) and uses the MATLAB data acquisition toolbox as well as the free Psychophysics toolbox. For the behavioral data reported here, we used a single m-file script of about 400 lines of code, which collects analog eye position data from the eye tracker for online control and storage, displays visual targets and monitors their fixation, delivers rewards, provides online feedback to the experimenter about current eye position and trial specific data. Our implementation is similar to the “WaVE” system (Meyer and Constantinidis, 2005), in that the code is basically contained in a single script, which has advantages for ease of modifications and adaptations for users who are skilled in MATLAB. Other MATLAB solutions such as “MonkeyLogic” (Asaad and Eskandar, 2008; Asaad et al., 2013) provide a general purpose environment that utilizes a multitude of scripts, and provide a graphical user interface for modifying experimental parameters. An important difference between these two systems and our implementation concerns the way that data are collected from the data acquisition board. Whereas both “WaVE” and “MonkeyLogic” use the combined storage and streaming mode of the board, our implementation relies uniquely on data streaming mode. Exclusive use of data streaming mode avoids timing inaccuracies and extensive data repetition that can occur during combined storage/streaming mode (Fig. 3A). It is a major advantage of our system that eye position data can be sampled with high temporal accuracy and without repetition. Providing experimenter feedback whenever a new eye position sample is acquired, our system achieves a timing performance of about 4 ms per update (Fig. 3B). Over 3 ms of this update time is needed to refresh the experimenter display, and if this display refresh is carried out infrequently, instead of after each newly acquired data point, the timing performance can be enhanced substantially. In our case, we can achieve about 1.3 ms per update on most trials, and about 4 ms for updates that include display refresh. Note that the 1.3 ms update time includes a waiting period of 1 ms, that we introduced because the DAQ sampling rate was set to 1 kHz and it therefore makes no sense to collect data at time intervals below 1 ms. Without the waiting period, our system also reaches

update intervals under 0.5 ms as reported in other studies (Asaad and Eskandar, 2008; Meyer and Constantinidis, 2005), but most of these updates are picking up redundant data from the DAQ board. We note that timing performance of the behavioral control system generally does not need to exceed the timing performance of the infrared tracking device. In our case, we used a camera with 120 Hz refresh rate, corresponding to about 8 ms between frames, such that a configuration with frequent experimenter display update was suitable for this hardware. For cameras with higher refresh rate, one may consider infrequent experimenter display update to improve system temporal precision.

A notable feature of our system is the use of a circular buffer, which reduces noise in the eye position estimates and enhances spatial accuracy. Because the buffering must be causal in online applications, it necessarily introduces a temporal delay that is proportional to the buffer length over which eye positions samples are averaged. The choice of buffer length will depend on particular experimental considerations, but we have found that values around 15 samples, corresponding to about 60 ms, tend to work well for the simple eye position calibration task investigated here. For a given buffer length, the detection delay for fixation onset depends on the distance between initial and target eye positions, and will have a minimum value of Δt as shown in Fig. 4B. It is important to remember that for many applications, the temporal delay can be essentially equated to a prolongation of fixation time. We therefore do not view the temporal lag introduced by circular buffering as problematic for most applications. An exception to this are tasks involving smooth pursuit eye movements, which are thus outside the scope of the procedures described here. Note that since the actual acquired eye position samples and not the circular buffered values are saved for later analysis, a simple (non-causal) moving average filter can be applied offline to deliver a veridical estimate of eye position without any temporal delays. In applications where faster response times are necessary, the infrequent update option described above can be used to reduce the temporal lag by a factor of up to about 3, or the camera can be placed in a more optimal position such that circular buffering may not be necessary.

The necessity of circular buffering is underscored by the behavioral data. Without circular buffering, monkeys achieved performance levels of only about 60% correct with a fixation window size (FWS) of 3.5° on a simple fixation task. With circular buffering, the FWS could be reduced to values as low as 2°, and still permit behavioral performance of about 80% correct in most cases. Note that monkeys were not extensively pre-trained on fixation tasks for these tests, so that their performance might improve with additional training. The reported values are certainly suitable to serve as a basis of training monkeys on a wide variety of oculomotor paradigms.

We quantified the stability of our system both at short and long timescales, and were able to demonstrate generally adequate stability that permits gathering data over periods of about one hour without the need for recalibration of any eye-tracker-related parameters. For eye position signal stability at short timescales, we obtained a value of $0.09 \pm 0.001^\circ$ for sample-to-sample variability, which is similar to previous reports for infrared eye tracking on rigidly head-posted monkeys (Kimmel et al., 2012). For eye position stability at longer timescales, we used a linear regression to determine the trial-to-trial drift over the course of behavioral session lasting approximately one hour. We found that drift tended to be generally small, with about three quarters of the session showing drift of less than $1^\circ/\text{h}$ and many sessions showing drift under $0.5^\circ/\text{h}$ (Fig. 6E). For head-posted monkeys, drift values of up to $0.3^\circ/\text{h}$ have been reported (Kimmel et al., 2012), which suggests that long timescale stability of eye tracking helmet-restrained monkeys can reach values observed in the rigidly head-posted condition. There are however occasional sessions with larger observed drift, that

will require manual readjustment of eye calibration parameters during the session to achieve optimal performance. The trial-to-trial variability of eye position data corresponding to the same fixation location over the time-course of about one hour was about a factor of two larger than values reported for head-posted monkeys (Kimmel et al., 2012), suggesting that our system performs only moderately inferior for this parameter.

Taken together, we have described a compact MATLAB implementation for behavioral control and eye tracking that is optimized for non-rigid head restrained macaque monkey applications. It is a low cost solution using a 120 Hz frame rate tracking system and the freely available Psychophysics toolbox. It relies exclusively on the streaming mode of the data acquisition board, which confers substantial benefits, particularly in terms of temporal fidelity, compared to the widely used combined streaming and storage mode. Although our system does not reach performance characteristics of invasive eye coil based eye trackers or high frame rate infrared tracking systems with rigid head-posting, it nevertheless achieves a performance that is suitable for many applications. We note that this system can of course also be used for human psychophysics studies, particularly in situations where head immobilization using chin-rest or bite-bar is not possible or undesirable.

References

- Asaad WF, Eskandar EN. [Achieving behavioral control with millisecond resolution in a high-level programming environment.](#) *J Neurosci Methods* 2008;173:235–40.
- Asaad WF, Santhanam N, McClellan S, Freedman DJ. [High-performance execution of psychophysical tasks with complex visual stimuli in MATLAB.](#) *J Neurophysiol* 2013;109:249–60.
- Brainard DH. [The psychophysics toolbox.](#) *Spat Vis* 1997;10:433–6.
- Judge SJ, Richmond BJ, Chun FC. [Implantation of magnetic search coil for measurement of eye position: an improved method.](#) *Vis Res* 1980;20:535–8.
- Kimmel DL, Mammo D, Newsome WT. [Tracking the eye non-invasively: simultaneous comparison of the scleral search coil and optical tracking techniques in the macaque monkey.](#) *Front Behav Neurosci* 2012;6:49.
- Kleiner M, Brainard D, Pelli D. [What's new in Psychtoolbox-3?](#) *Perception* 2007;36.
- Machado CJ, Nelson EE. [Eye-tracking with nonhuman primates is now more accessible than ever before.](#) *Am J Primatol* 2011;73:562–9.
- Meyer T, Constantinidis C. [A software solution for the control of visual behavioral experimentation.](#) *J Neurosci Methods* 2005;142:27–34.
- Pelli DG. [The VideoToolbox software for visual psychophysics: transforming numbers into movies.](#) *Spat Vis* 1997;10:437–42.
- Robinson DA. [A method of measuring eye movement using scleral search coil in a magnetic field.](#) *IEEE Trans Biomed Eng* 1963;10:137–45.