

EyeLink and iEye Adaptation

Qingqing Yang, qy775@nyu.edu

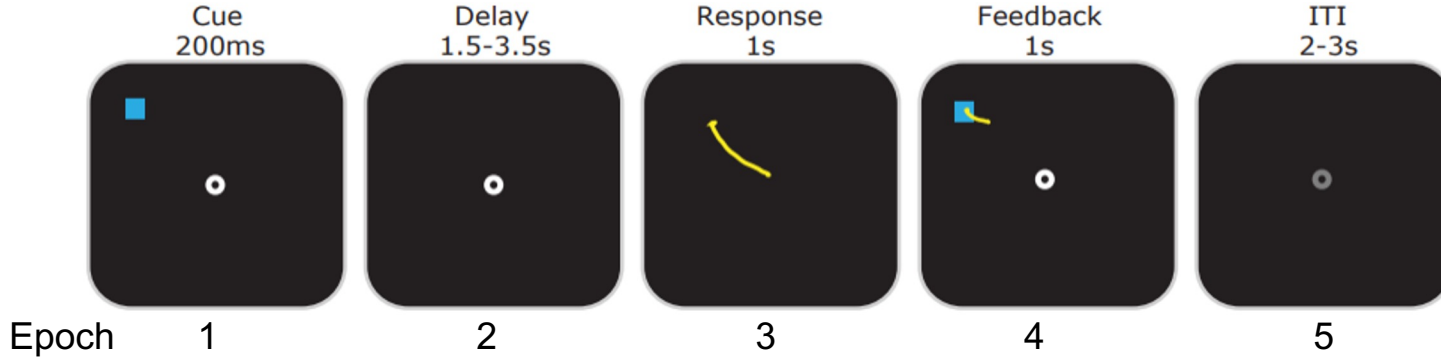
Clayspace Lab Meeting



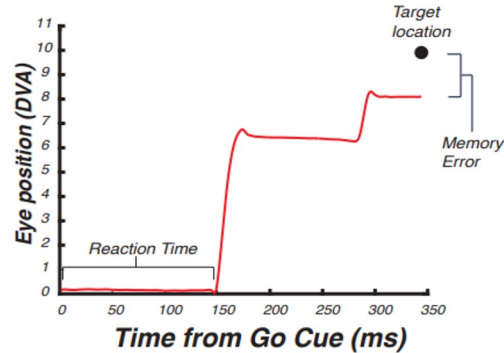
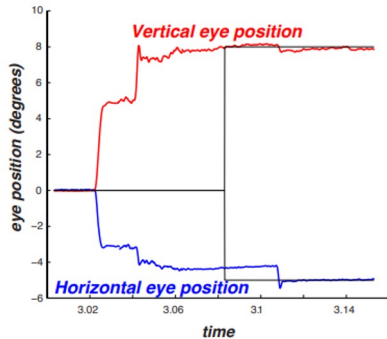
Contents

- **EyeLink Tutorial**
 - **Eye-Tracker data**
 - **Code for Setup and Message**
- **iEye Tutorial**
 - **Procedure of data processing**
 - **Adaptation**

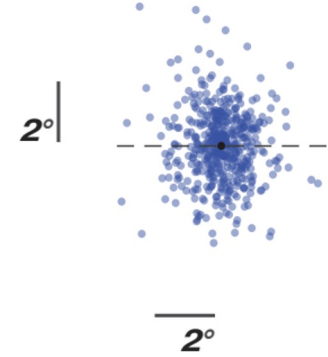
Memory Guide Saccade (MGS) Task



Saccade Trajectory

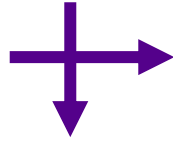


Aligned Saccadic Endpoints



Eye-Tracker Data

- X axis: Right +
- Y axis: Down +
- Video:
Right --> Left --> Up --> Down



- Explanation SR-Research EyeLink(c) gazetrackers
 - Sample: pupil size, x & y gaze position (pixel) at each time point
 - Event: event type such as start-end of blink, saccade, fixation.



EDF_Data_Structure.pdf

EyeLink Setup

- EyeLinkToolbox can be used to create eye-movement experiments and control the SR-Research EyeLink(c) gaze trackers
- It's incorporated into the PsychToolbox ([Psychtoolbox-3 – EyeLinkToolbox](#))
- It provides a number of wrapper functions to simplify creating an eye-tracking program
- The main functionality could be found in demos: [EyeLinkToolbox/EyeLinkDemos](#)

EyeLink Setup

- Initialization

```
% Provide EyeLink with details about the graphics environment
```

```
el=EyeLinkInitDefaults(window);
```

```
EyeLink('Initialize','PsychEyeLinkDispatchCallback');
```

```
EyeLink('command', 'sample_rate = 1000');
```

```
% make sure that we get gaze and event data from the EyeLink
```

```
status=EyeLink('command','link_sample_data = LEFT,RIGHT,GAZE,AREA');
```

```
EyeLink('command','file_sample_data = LEFT,RIGHT,GAZE,AREA,GAZERES,STATUS');
```

```
EyeLink('command','file_event_filter =
```

```
LEFT,RIGHT, FIXATION,SACCADE,BLINK,MESSAGE,BUTTON');
```

EyeLink Setup

- Calibrate

```
% Calibration Setup
```

```
Eyelink('command', 'enable_automatic_calibration = YES');  
Eyelink('command', 'calibration_type= HV5');
```

```
% Calibrate the eye tracker
```

```
EyelinkDoTrackerSetup(e1);
```

```
% do a final check of calibration using driftcorrection
```

```
EyelinkDoDriftCorrection(e1);
```

EyeLink Save File

- Save data

```
% get gaze data from EyeTracker
tempedfFile = 'temp.edf';
edfFile='actual_file_name.edf';
Eyelink('openfile',tempedfFile);

% stop Eye Tracker Recording
if is_eyetracker
    Eyelink('stoprecording');
    Eyelink('ReceiveFile',tempedfFile,tempedfFile);
    movefile(tempedfFile,edfFile);
    WaitSecs(3); print('Data Trans Completed.');
    Eyelink('Shutdown')
end
```


EyeLink Send Message

- Send info of experiment epoch, target locations to eye tracker

- During epoch fixed at the center

```
Eyelink('Message', 'TarX1 %s', num2str(0));
```

```
Eyelink('Message', 'TarY1 %s', num2str(0));
```

- During the epoch respond to the target

```
Eyelink('Message', 'TarX1 %s', num2str(task.dotXdva{iTrial}));
```

```
Eyelink('Message', 'TarY1 %s', num2str(task.dotYdva{iTrial}));
```

- Send Target Location in the unit of **Degree of visual angle**

EyeLink GazeContingent

- Send info of experiment epoch, target locations to eye tracker

```
eye_used= EyeLink('EyeAvailable');
if EyeLink( 'NewFloatSampleAvailable') > 0
    evt = EyeLink( 'NewestFloatSample'); % Get most recent sample data (gaze loc, pupil
size)
    if eye_used ~= -1 % do we know which eye to use yet?
        % if we do, get current gaze position from sample
        x = evt.gx(eye_used+1); % +1 as we're accessing MATLAB array
        y = evt.gy(eye_used+1);
        % do we have valid data and is the pupil visible?
        if x~=e1.MISSING_DATA && y~=e1.MISSING_DATA && evt.pa(eye_used+1)>0
            mx=x;
            my=y;
        end
    end
end
```

EyeLink GazeContingent Cont.

- Send info of experiment epoch, target locations to eye tracker
 - When subject gazes at an object, it changes color. When gaze moves away, it toggles back

```
% check for events
evtype=EyeLink('getnextdatatype');
if evtype==e1.ENDSACC % if the subject finished a saccade check if it fell on an object
% check if saccade landed on an object
    choice=-1;
    noobject=0;
    i=1;
    while 1
        if 1==IsInRect(evt.genx,evt.geny, object(i).rect )
            choice=i;
            break;
        end
        i=i+1;
        if i>length(object)
            noobject=1;
            break;
        end
    end
end
```

iEye Procedure

- iEye is a set of command line functions built to translate data from 'raw' format (typically, EDF files) into scored responses on each trial.
- [clayspacelab/iEye at iEye_ts \(github.com\)](https://github.com/clayspacelab/iEye_ts)



iEye Procedure

- Preprocessing

ii_preproc.m

ii_import

ii_data

Message events based on .ifg

ii_trim, ii_rescale, ii_invert, ii_censorchans, ii_definetrial

ii_blinkcorrect

```
% blink correction
```

```
ii_params.blink_thresh = 1.5; % PERCENTILE
```

```
ii_params.blink_window = [150 50]; % before and after, ms
```

ii_smooth, ii_velocity

ii_findsaccades, ii_extractssaccades

```
sacc_dur >= dur_thresh (0.0075) & sacc_amp >= amp_thresh (0.25)
```

```
& velocity >= 30 (dva/s)
```

ii_findfixations

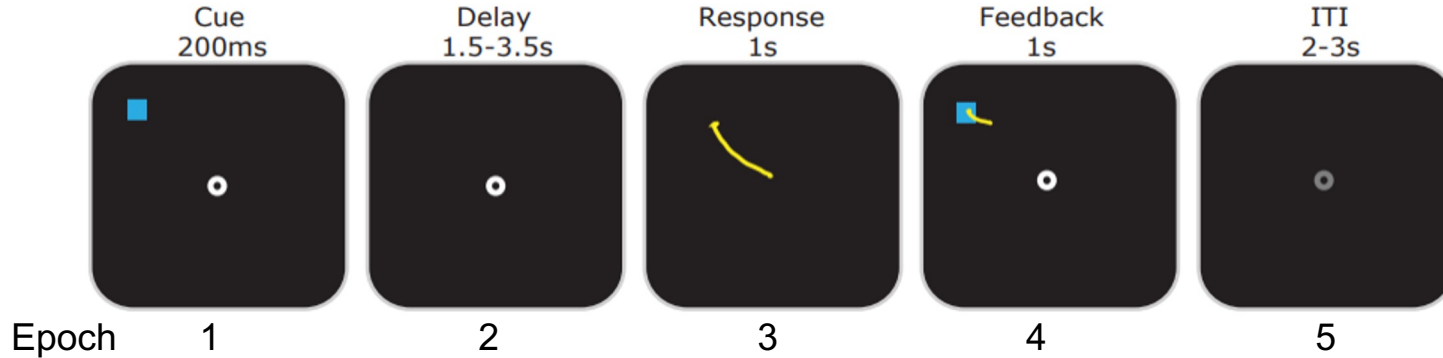
X_fix
Y_fix

499977x1 double
499977x1 double

X 500143x1 double
Y 500143x1 double
Pupil 500143x1 double
XDAT 500143x1 double
TarX1 500143x1 double
TarX2 500143x1 double

amplitude
peakvelocity
duration
idx
epoch_start
epoch_end
trial_start
trial_end
X_start
X_end
X_trace
Y_start
Y_end
Y_trace
t

iEye Procedure



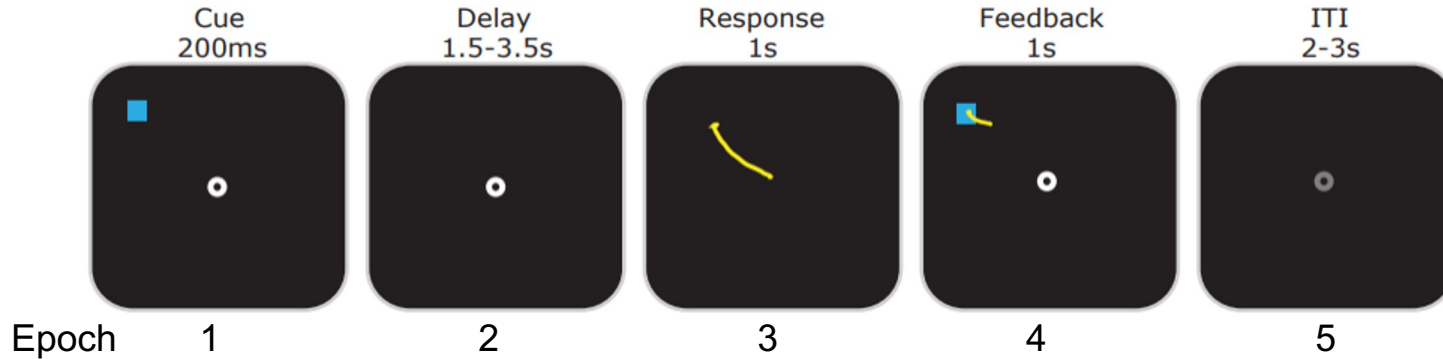
- Preprocessing cont.

`ii_preproc.m`

drift correction – epoch 1 & 2

```
ii_selectfixationsbytrial  
ii_driftcorrect
```

iEye Procedure



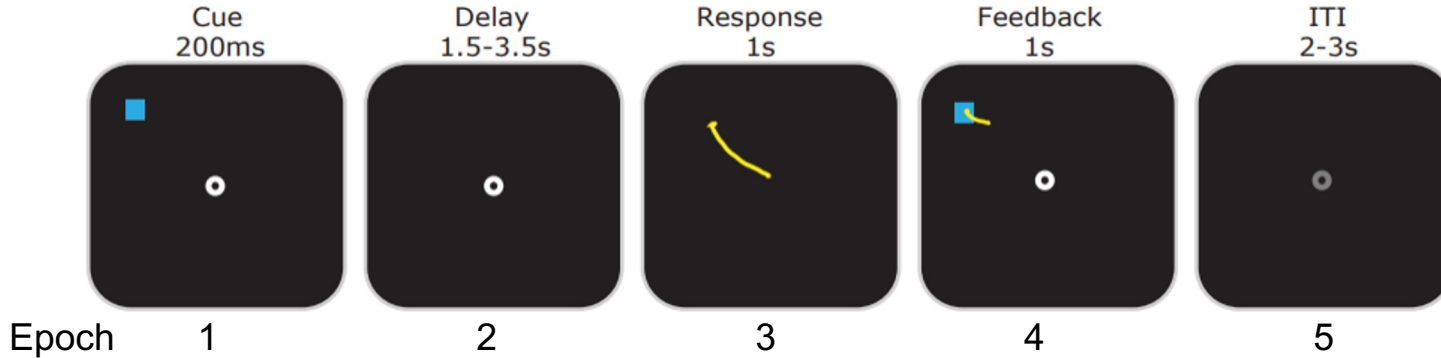
- Preprocessing cont.

ii_preproc.m

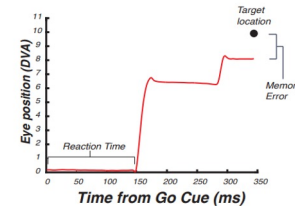
calibration – epoch 4

```
ii_selectfixationsbytrial  
ii_calibratebytrial
```

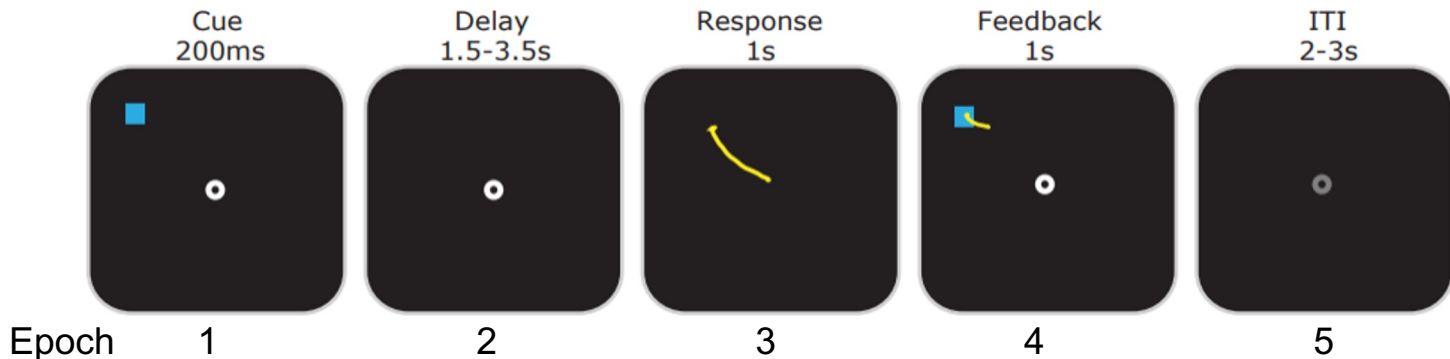
iEye Procedure



- Score the saccade error
 - Extract initial and final saccade eye position during certain **response epoch 4**
 - Target coordinate in TarX, TarY
 - Calculate the Euclidian distance and response time
 - Mark bad trials in ii_sess.excl_trial



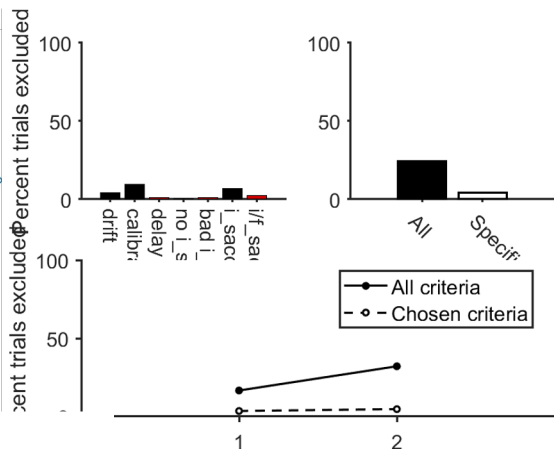
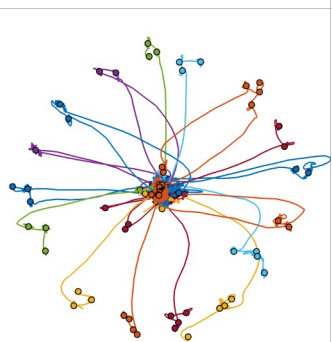
iEye Adaptation



- Epochs in the main ieye script
- lfg channel num and names
- Multi-item: score each target response separately, calibrate for every feedback ([Qingqing-Yang-177/iEye_qy-WMLoads](https://github.com/Qingqing-Yang-177/iEye_qy-WMLoads) at `iEye_ts`)

```
ii_params.valid_epochs = [1 2 3 4 5 6];  
ii_params.trial_end_value = 6; % XDAT value for trial end  
ii_params.drift_epoch = [1 2 3]; % XDAT values for drift correction  
ii_params.calibrate_epoch = 5;  
ii_params.response_epoch = 4;  
ii_params.plot_epoch = [3 4 5];
```

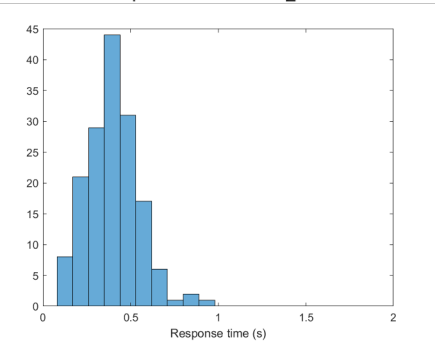
iEye Results



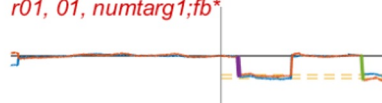
i_sacc



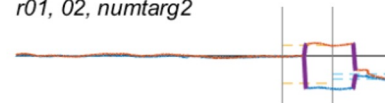
f_sacc



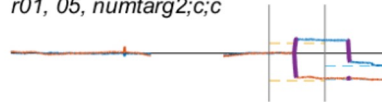
*r01, 01, numtarg1;fb**



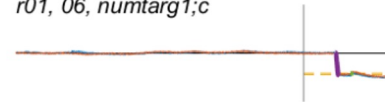
r01, 02, numtarg2



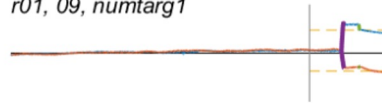
r01, 05, numtarg2;c;c



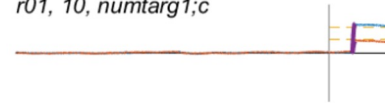
r01, 06, numtarg1;c



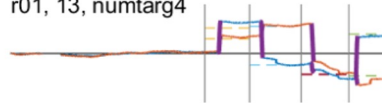
r01, 09, numtarg1



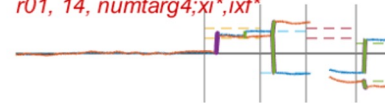
r01, 10, numtarg1;c



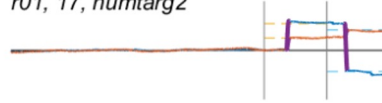
r01, 13, numtarg4



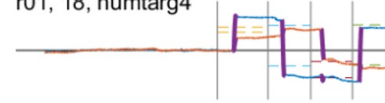
r01, 14, numtarg4;xi,ixf**



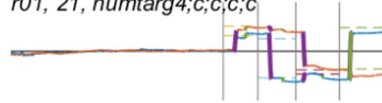
r01, 17, numtarg2



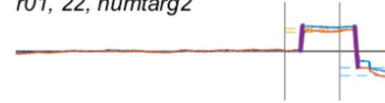
r01, 18, numtarg4



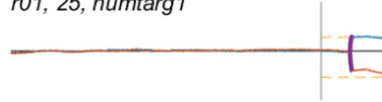
r01, 21, numtarg4;c;c;c;c



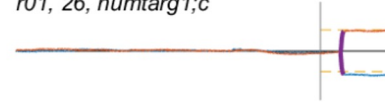
r01, 22, numtarg2



r01, 25, numtarg1



r01, 26, numtarg1;c



Thank you!