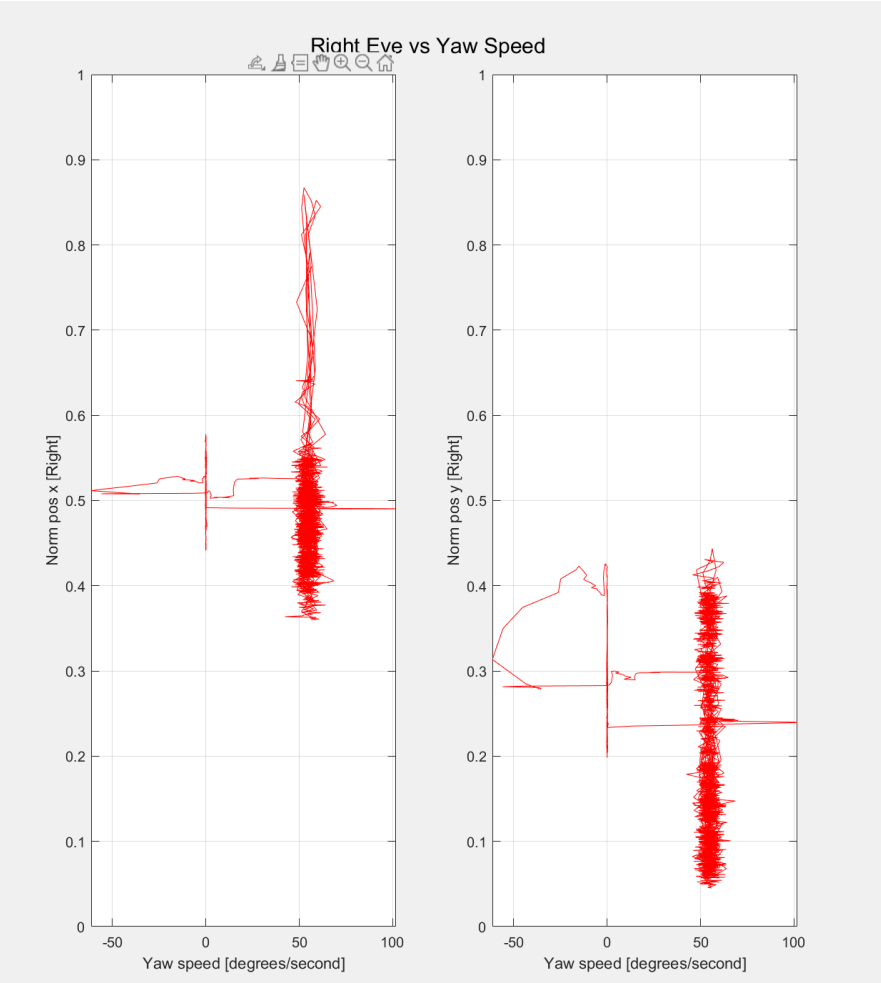
Experimental setup for nystagmus analysis with SP7 motion platform & pupil eye tracking

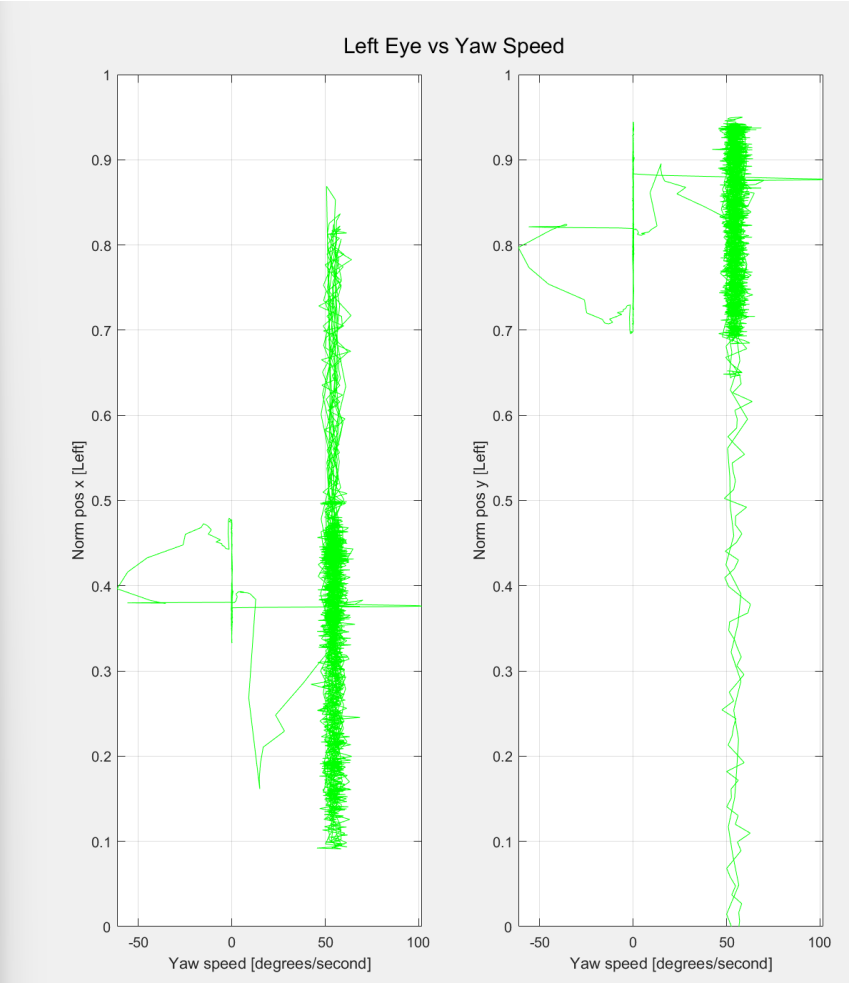
Mohamed Sadiq IKBAL, Sofia CAMPI

A use case of SP7 motion platform in rehabilitation studies

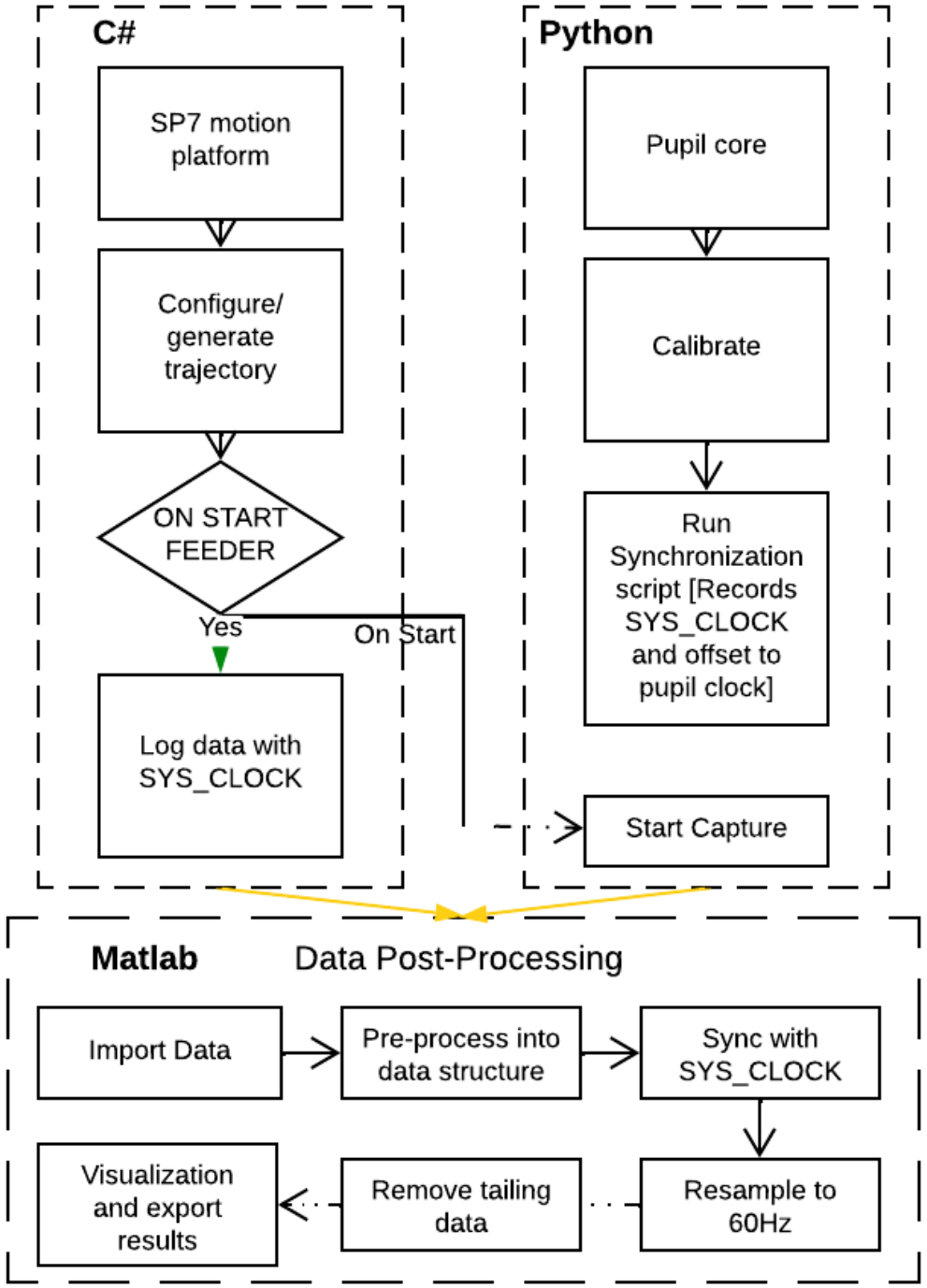
Objective

- Compare eye tracking data with motion data
  - Synchronized point pair matching with accuracy

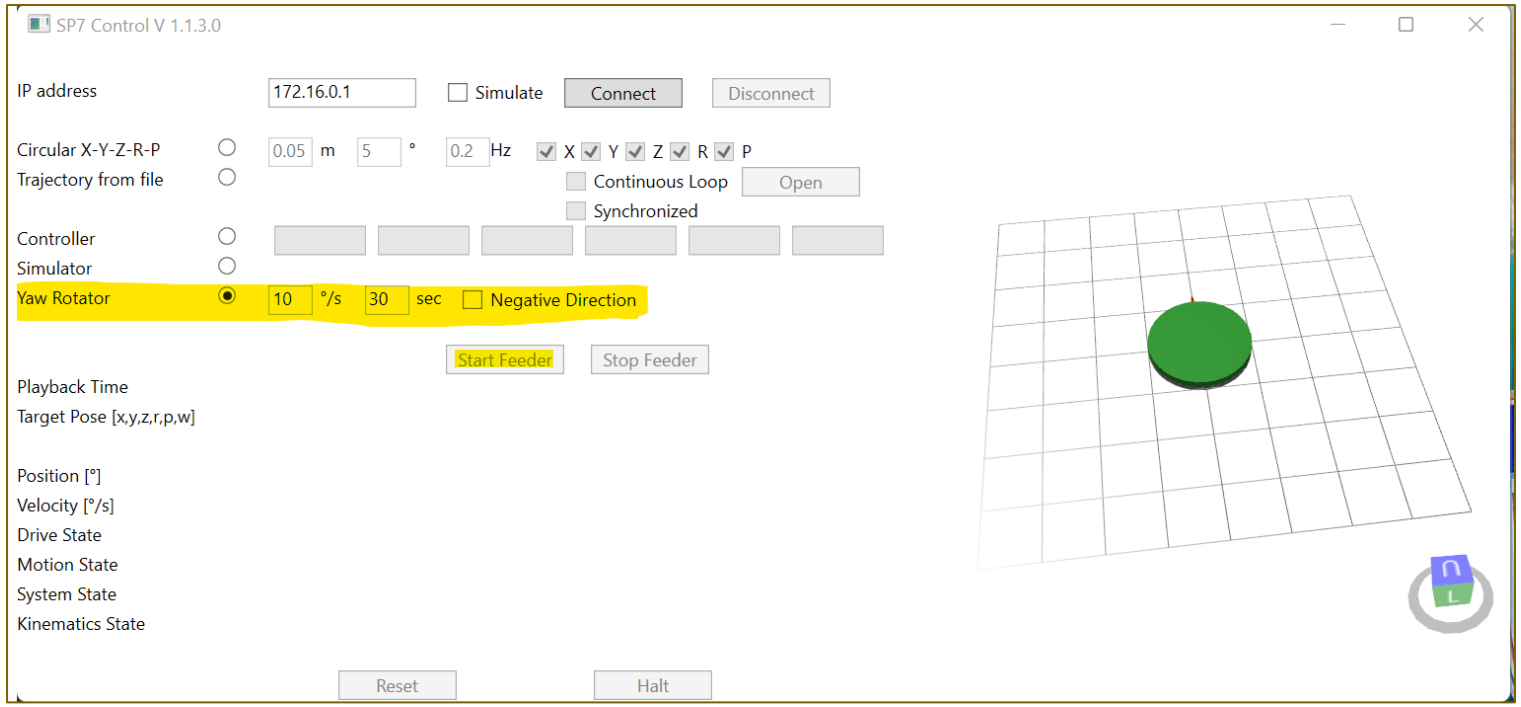




System Architecture



SP7 Controller



- SP7 control app customized for pure yaw rotation
- Logs data with SYS_CLOCK on click of “Start Feeder”
- Parameters: speed and duration [Negative direction can be set, if needed]
- Connect →Set parameter → Start Feeder
- Log file located in the location of the executable file [ex: SP7MotionData-20220601182715.log]

SP7 Controller Customization

```
<TextBlock Grid.Row="8" Grid.Column="0" Text="Yaw Rotator" />
<RadioButton Name="YawRotationFeederCB" Grid.Row="8" Grid.Column="0" HorizontalAlignment="Right" GroupName="FeederType"/>
<TextBox Name="NumberOfRevolutionBox" Grid.Row="8" Grid.Column="2" Text="10" HorizontalAlignment="Left" Width="30" />
<TextBlock Grid.Row="8" Grid.Column="2" Text="°/s" Margin="35,0,0,0" />
<TextBox Name="DurationBox" Grid.Row="8" Grid.Column="2" Text="30" HorizontalAlignment="Left" Width="30" Margin="65,0,0,0" />
<TextBlock Grid.Row="8" Grid.Column="2" Text="sec" Margin="100,0,0,0" />
<CheckBox Name="NegativeDirCK" Grid.Row="8" Grid.Column="2" Content="Negative Direction" Margin="130,0,0,0" VerticalAlignment="Center"
 IsChecked="false" />
```

```
while (feeding && _sp7.IsConnected)
{
    ms = stopwatch.ElapsedMilliseconds;
    long dtms = ms - lastms;
    lastms = ms;
    double t = ((double)ms) / 1000.0;

    Dispatcher.Invoke(() =>
    {
        PlaybackTimeLbl.Text = $"{t:0000.000}";
    });

    if (t <= duration)
    {
        jdot.vax7 = speed;
        j.ax7 = jdot.vax7 * t;
        _sp7.SetJointTargets(j, jdot);
    }
    else
    {
        stopwatch.Stop();
        feeding = false;
        Thread.Sleep(500);
        jdot.vax1 = jdot.vax2 = jdot.vax3 = jdot.vax4 = jdot.vax5 = jdot.vax6 = jdot.vax7 = 0;
        _sp7.SetJointTargets(j, jdot);
        Thread.Sleep(500);
        UpdateControls();
        //_sp7.Reset();//uncomment this if you want the platform to return to zero
    }
    Thread.Sleep(1000 / 60);
}
```

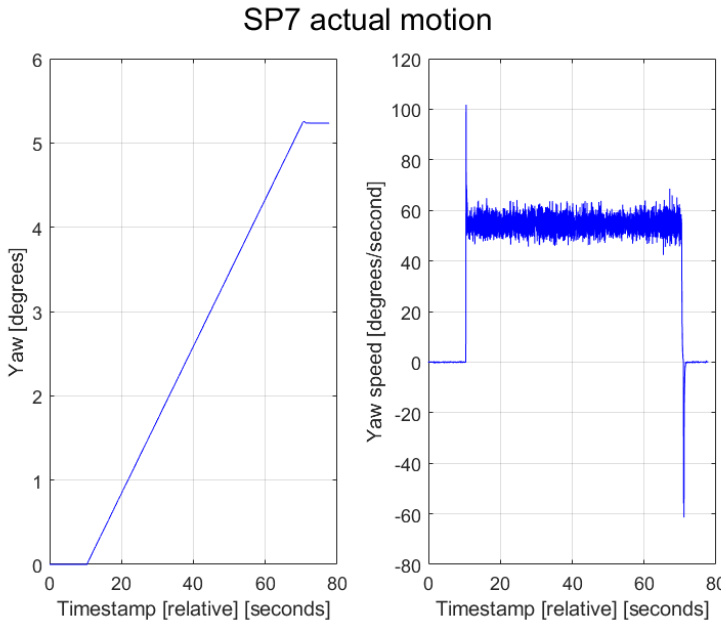
Pupil Eye Data Synchronization

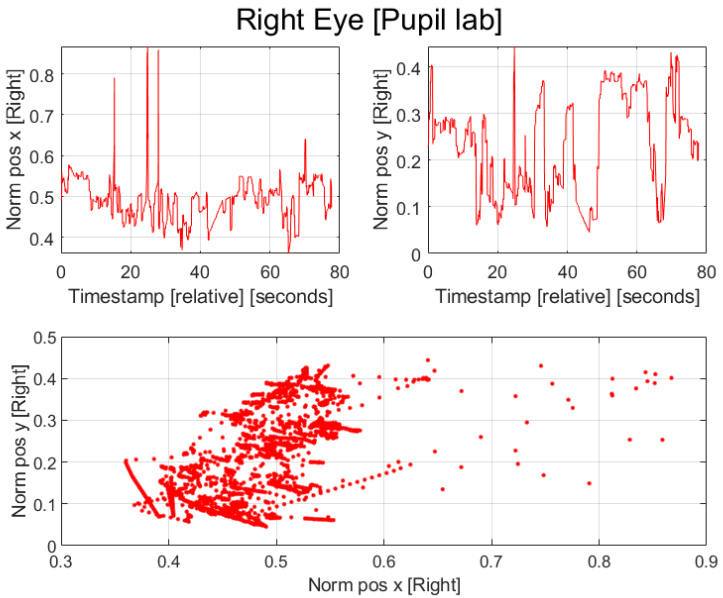
- Record the system clock and the measure the offset to pupil_timestamp [run simple_realtime_time_sync.py]
- Opens a file “sync_info_<Timestamp>.info” in the same directory.

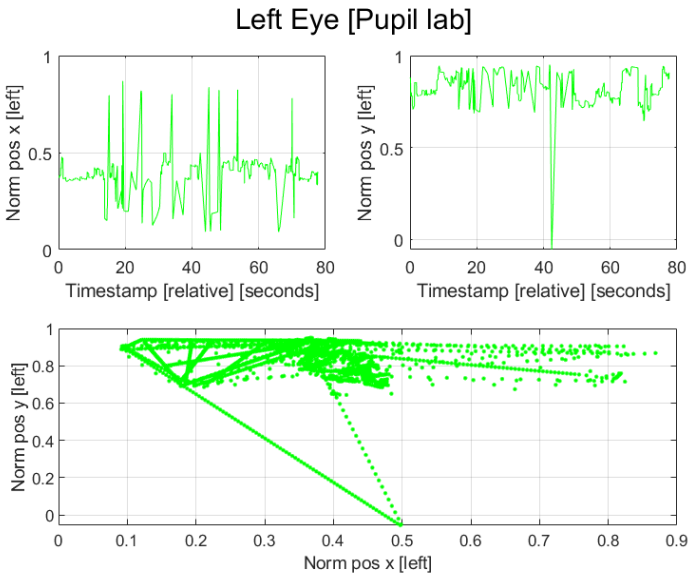
```
fileName = f"sync_info_{(int(time.time() * 1000))}.info"
try:
    f = open(fileName, 'w')
    # 3. Measure clock offset once
    offset = measure_clock_offset(socket, clock_function=local_clock)
    f.write(f"Clock offset (1 measurement): {offset} seconds\n")
    # 4. Measure clock offset more reliably to account for network
    # latency variance (multiple measurements)
    number_of_measurements = 10
    stable_offset_mean = measure_clock_offset_stable(
        socket, clock_function=local_clock, nsamples=number_of_measurements
    )
    f.write(f"Mean clock offset ({number_of_measurements} measurements): "
            f"{stable_offset_mean} seconds\n")

    # 5. Infer pupil clock time from "local" clock measurement
    local_time = local_clock()
    pupil_time_calculated_locally = local_time + stable_offset_mean
    f.write(f"Local time: {local_time}\n")
    f.write(
        f"Pupil time (calculated locally): {pupil_time_calculated_locally}\n")
    # 6. Log information to a file
finally:
    f.close()
```

```
1  Clock offset (1 measurement): -1653857113.47857 seconds
2  Mean clock offset (10 measurements): -1653857113.4873507 seconds
3  Local time: 1654100956.977858
4  Pupil time (calculated locally): 243843.49050736427
```

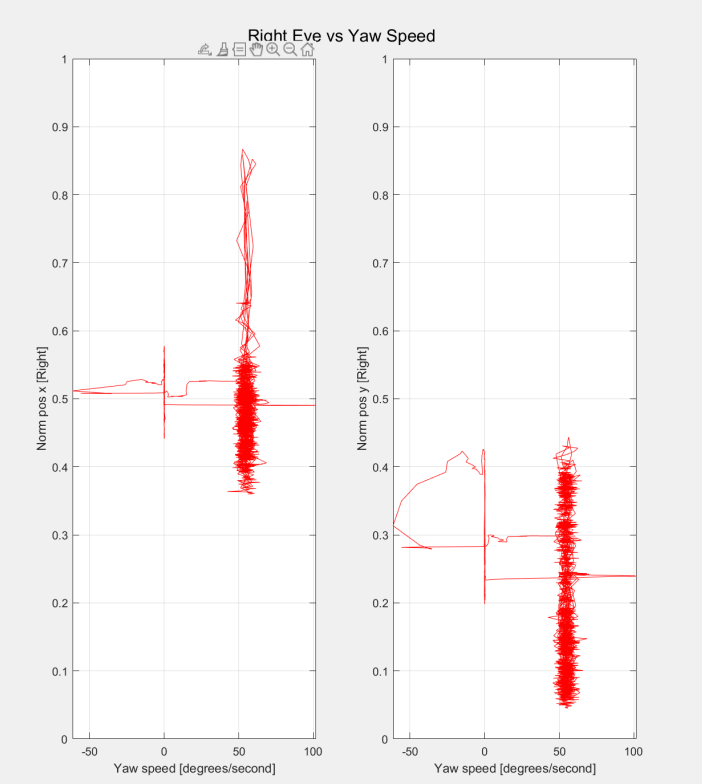
Post-Processing Data

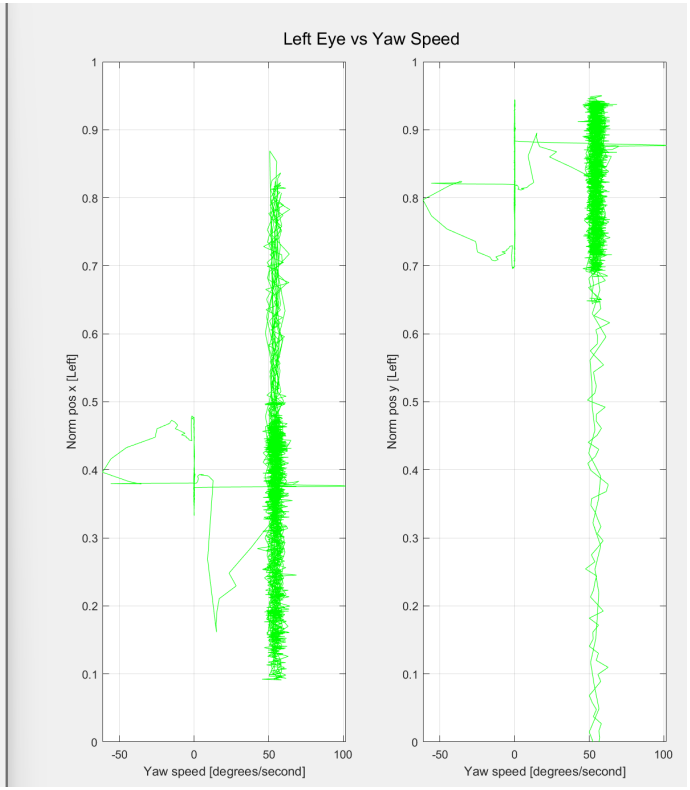


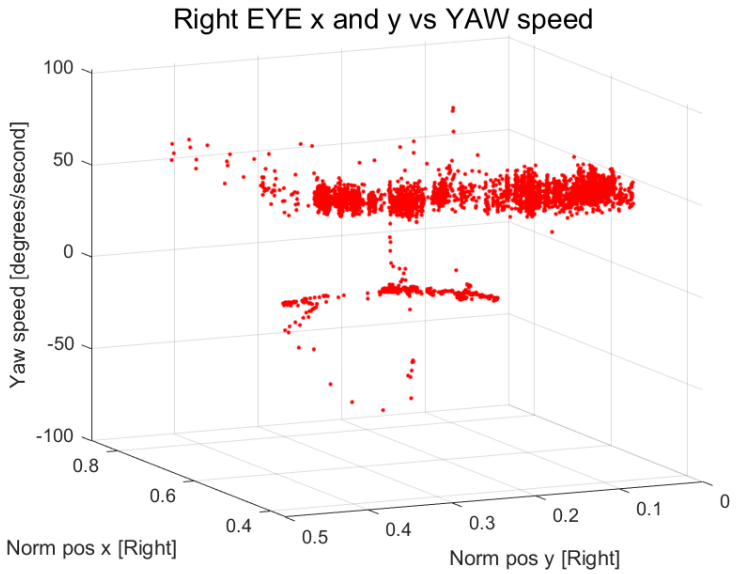




Post-Processing Data







Useful Links

- Github [post-processing code]
  https://github.com/MSIhub/NystagmusExperimentalSetup.git

- Pupil lab synchronization info
  https://docs.pupil-labs.com/core/best-practices/