2010

Eye Movements in Elite Athletes - An Index for Performance.

Harshad Hegde
Virginia Commonwealth University

EYE MOVEMENTS IN ELITE ATHLETES: AN INDEX FOR PERFORMANCE

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science at Virginia Commonwealth University.

by

HARSHAD B. HEGDE
Bachelor's of Science, Mumbai University, India, 2006

Director: DR. GERALD E. MILLER
PROFESSOR AND DEPARTMENT CHAIR, DEPARTMENT OF BIOMEDICAL ENGINEERING.

Virginia Commonwealth University
Richmond, Virginia
August, 2010

Acknowledgement

First and foremost I would like to thank my research advisor, Dr. Peter Pidcoe, who has supported me throughout my thesis with his patience, knowledge and occasional humor. I attribute the level of my Masters degree to his encouragement and effort and without him this thesis, too, would not have been completed or written.

I am grateful to the department chair Dr. Gerald Miller for his invaluable support and help in whatever difficulties I had throughout my time in school. I would like to thank Dr. Paul Wetzel, for his unconditional guidance and support during the Signal Processing class that helped me understand crucial concepts essential for my project and ultimately my thesis.

I would also like to thank Dr. Mary Shall for her invaluable knowledge and suggestions during my research.

I am extremely thankful to Dr. Rosalyn Hobson and Ms. Leena Joseph for their invaluable help and guidance which help me graduate.

I would like to thank my Lab-mate Ms. Cortney Bradford for helping me whenever I had queries.

I am extremely thankful to Mr. Tom Diehl, Mr. Al Jeffers and my coworkers (J.J., Ian, Jordan and Sneed) at the Cary Street Gym through whom I learnt a lot about the American culture. I really enjoyed working with them.

Where would I be without my family? My parents deserve a special mention for their inseparable support and prayers. My father, Mr. U. Balakrishna Hegde, in the first

place is the person who showed me the joy of intellectual pursuit ever since I was a kid. My mother, Mrs. Shevanti Hegde, is the one who sincerely raised me with her caring and gentle love. Ashu, thanks for being a supportive, loving and caring brother.

Last but not the least, I would also like to thank all my friends especially Avani, Bobsen, Bondal, Dipen, Dhaduk, Jayul and Shankar for all the great times we've shared together throughout my graduate study. Also, I would like to thank all my subjects without whom none of this would have been possible. Above all, I would like to thank God for helping me in every walk of life.

Table of Contents

Page

List of Tables

Page

List of Figures

Page

Abstract

EYE MOVEMENTS IN ELITE ATHLETES: AN INDEX FOR PERFORMANCE

By Harshad B. Hegde, B.E. Biomedical Engineering, India.

A Thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science at Virginia Commonwealth University.

Virginia Commonwealth University, 2010

Major Director: Dr. Gerald E. Miller,
Professor and Chair, Department of Biomedical Engineering

In gymnastics, athletes perform twisting and flipping skills at high angular velocities. These athletes rely heavily on sensory information from the visual, proprioceptive, and vestibular systems. The vestibulo-ocular reflex (VOR) is responsible for stabilizing the visual field on the retina during head movement. To accomplish this, the eyes are reflexively moved in a direction opposite the head. In a twisting gymnast, this actually reduces the ability of gymnasts to see the landing during airborne skills. Hence it becomes necessary for the gymnasts to cancel or suppress their VOR in order to

view the landing. The purpose of this research is to investigate the relationship between gymnastics skill level and their ability to suppress the VOR.

Ten female gymnasts (mean age 15±2.2) were obtained via a sample of convenience from a local club. Saccadic peak velocities and latencies were calculated for the sample population. Their performance did not differ from the normal population. VOR and VORc gains were also calculated and compared. The higher level (competitive) gymnasts were better at suppressing their VOR. In addition, left/right VOR gain asymmetries correlated highly with twist direction in seven of the competitive gymnasts.

There is a correlation between VOR performance and gymnastic level. These results do not suggest that VOR differences develop as a result of practice. These differences may simply allow some individuals to become better performers. A longitudinal study on a larger population would be required to test the causal relationship between these variables.

Introduction

Sports like gymnastics, diving, and ice skating often involve rotational elements with high angular velocities. Many of these are airborne skills and have a significant element of risk. The athletes who perform these skills have to rely heavily on sensory information from the visual and vestibular systems. It has been suggested that the vestibular system adapts to these types of inputs and the athlete no longer experiences the dizziness associated with multiple rotations [1].

With gymnastics, there are two types of rotations to consider. The first is sagittal plane rotation. This occurs around a medial / lateral axis defined as the flipping axis (Figure 1). Multiple flipping skills are typically performed by more advanced gymnasts. The second is transverse plane rotation. This occurs around a superior / inferior axis defined as the twisting axis (Figure 1). Single and multiple twisting skills occurring around both axes simultaneously are taught to intermediate and advanced gymnasts.

A primary focus in training skills of this type is head position. Not just to promote good mechanics, but to allow the gymnast a view of the landing. The focus of this research is on twisting (superior / inferior axis rotational) skills. A reflex called the vestibular ocular reflex (VOR) actually reduces the gymnasts' ability to see the landing during twisting activities. The details of this reflex will be discussed later, but simply put the horizontal component of this reflex acts to stabilize the eyes during rotational (yaw)

head movements by moving the eyes in a direction opposite head rotation. This would result in "blind" landings for the gymnast and many do seem to rely on timing to perform multiple twisting flips. Some gymnasts however, report being able to see their landings. This implies they have the ability to suppress the VOR. The purpose of this research is to investigate differences in the VOR performance of young female gymnasts, who routinely perform flipping twisting activities. To fully understand the complexity of a flipping and twisting body, a discussion of balance and the components that lead to balance control must be discussed.

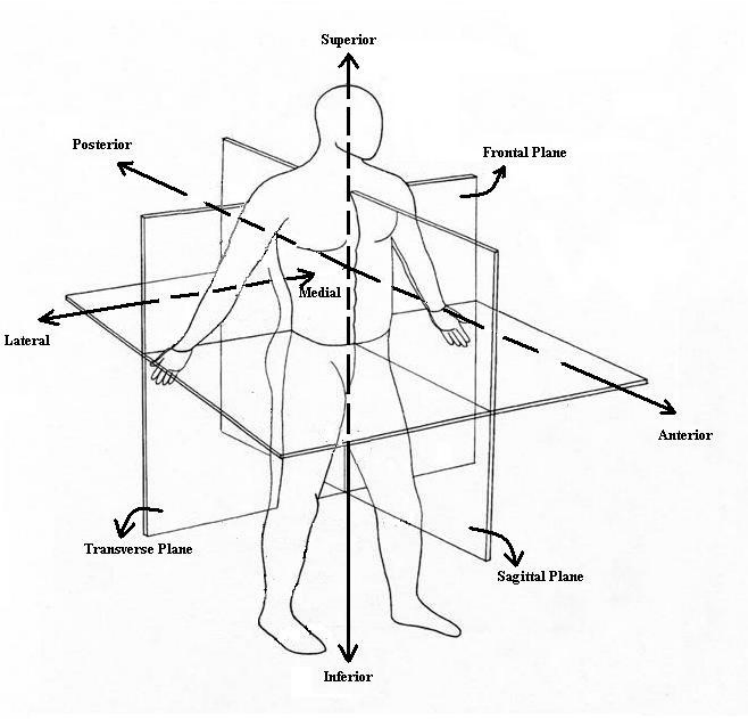

Figure 1 – The planes and axes of the body.

Balance

Proper balance or the control of balance requires the complex integration of information regarding body position and the generation of appropriate motor responses to

control that position [2]. Maintenance of balance involves the accurate co-ordination of the visual, vestibular, and proprioceptive information [2]. Integration of these systems are described in Figure 2. This diagram depicts each system's information combining to be interpreted by the central nervous system. Some responses to balance perturbation are reflexive, others require cognitive control [2]. Classic theories suggest that the loss or degradation of any one system is compensated for by the remaining systems [45]. Degradation or loss would be depicted in this diagram as a gain reduction to less than 1.

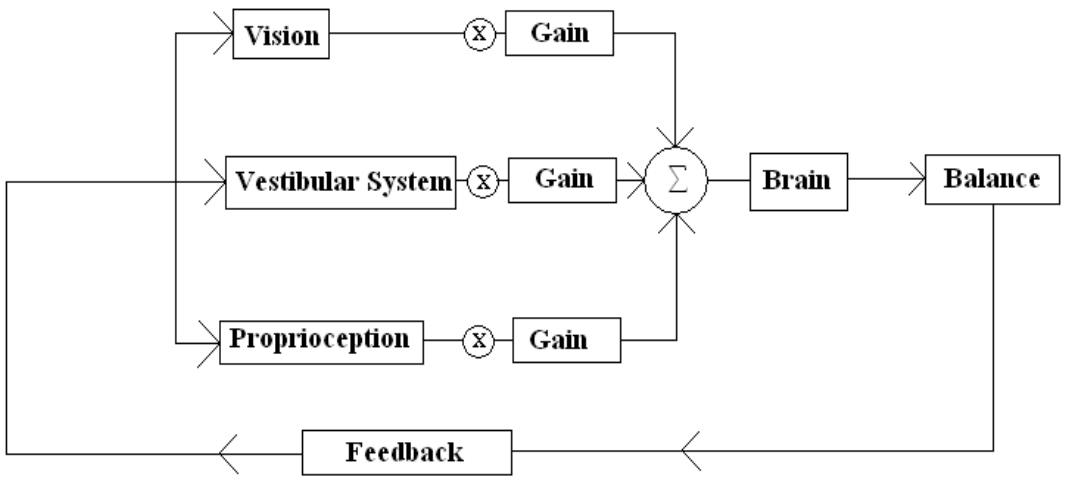

Figure 2 – A model showing balance maintenance.

To understand balance, the role of each system needs to be considered. In the context of gymnastics and twisting gymnasts, the inputs of each system change dramatically from the start to the end of the skill. This differs from the common standing balance examples described in literature [63 7]. Proprioception is typically related to sensed joint positions and somatosensory information from contact with external surfaces. Like the pressure you feel between your feet and floor when standing and how that pressure

changes and you sway or walk. When a gymnast is airborne, proprioceptive input from external surfaces is lost. The gymnast must rely on vestibular and visual information to determine position in space. If the eyes are closed, then visual information is also lost. In this case, the vestibular information would not be sufficient to orient the gymnast since visual information is needed to reset this sensory input [7]. Visual information during airborne twisting skills is extremely important [9]. A discussion of the visual system follows.

The Visual System.

The visual system in human beings helps in judgement of distances and interpretation of spatial information [2]. When any object is viewed, light that reflects off that object reaches the eye. This light is focussed by the cornea and lens onto the light-sensitive membrane in the back of the eye called the retina. The photoreceptive cells (rods and cones) on the retina, produce nerve impulses the moment light (photons) is incident on it. The optic nerve carries these nerve impulses to the brain. This helps the person see the object and its surroundings clearly (as shown in figure 3).

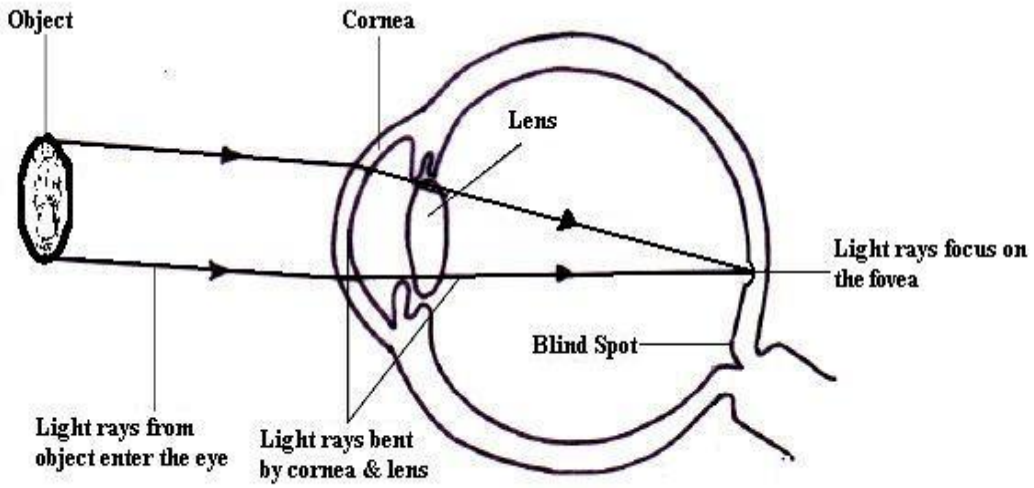

Figure 3 – The Visual System: The path followed by light rays from an object when viewed by an observer.

To see an object clearly(maximum visual acuity), the object must be focused on the fovea. The fovea is a pit-like structure at the back of the retina which has a diameter of about 1mm. It has a high concentration of photoreceptor cells (called cones). The high spatial density of the cones in the fovea is responsible for the high visual acuity in this region. The fovea is located about 4 – 8 degrees temporal to the optical axis and occupies about 1 degree of visual angle [10].

In proximity to the fovea, there is an area in the retina of each eye where the optic nerve exits the retina which forms the optic disc. The optic disc is deprived of photoreceptors and hence supplies no visual signal. This area is called the 'scotoma' or a blind spot. A graphical representation of the visual acuity of the human eye as a function of retinal position in the horizontal plane is shown in figure 4.

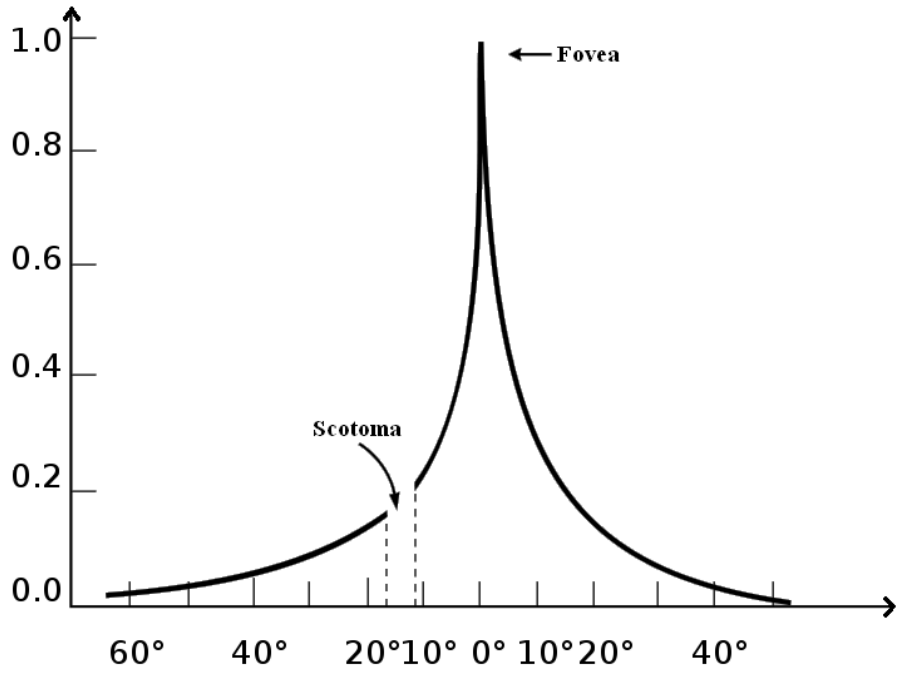

Figure 4 – Visual acuity of the left human eye in terms of degrees from the fovea in the horizontal plane. The X-axis represents the degree of acuity with 1 being the maximum. The scotoma lies nasal to the fovea.

Eye movement is made possible by six extraocular muscles.These are illustrated in Figure 5. Activation of these muscles sigularly or in combination allow fine control of the eye position. It's easiest to first describe the role of each muscle and then to consider the effect of combinations.

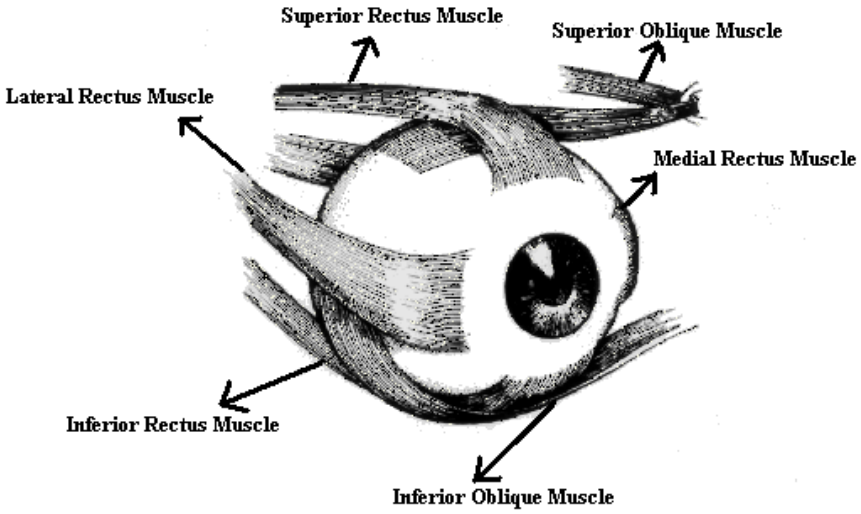

Figure 5 – Extraocular muscles for the right eye.

The Superior Rectus muscle has the primary function of elevation of the eye but also plays a role during intorsion and adduction of the eye. The Inferior Rectus muscle has the primary function of depression of the eye but also plays a role during extorsion and adduction of the eye. The Lateral Rectus muscle is responsible for the movement of the eye away from the midline of the body (abduction) whereas Medial Rectus muscle contributes to the movement of the eye towards the midline of the body (adduction). The Superior Oblique muscle is responsible for the movement of the top of the eye towards the nose (intorsion) and also plays a role during depression and abduction. The Inferior Oblique muscle is responsible for the movement of the top of the eye away from the nose (extorsion) and also plays a role during elevation and abduction. The combinations of muscles required to move the eye are listed in Table 1.

Axis of Rotation	**Motion of the eye**	**Contracting Muscles**	**Relaxing Muscles**
Vertical	Abduction Adduction	Lateral Rectus Medial Rectus	Medial Rectus Lateral Rectus
Horizontal	Supraduction (Elevation) Infraduction (Depression)	Superior Rectus Inferior Oblique Inferior Rectus Superior Oblique	Inferior Rectus Superior Oblique Superior Rectus Inferior Oblique
Antero-posterior	Incycloduction (Intorsion) Excycloduction (Extortion)	Superior Rectus Superior Oblique Inferior Rectus Inferior Oblique	Inferior Rectus Inferior Oblique Superior Rectus Superior Oblique

Table 1 – Single eye movements and the muscles involved.

Although visual inputs can help detect motion, the use of visual information alone can lead to misinterpretation of movement cues. For example, consider someone sitting in a car at a stop sign. They look out the window and see the truck next to them moving forward slowly. Relying on vision alone can lead to one of two conclusions: (1) the truck is moving forward or (2) their car is moving backwards.

The addition of vestibular and prorioceptive information would help lead to a singular conclusion. Under normal circumstances, it is a combination of sensory inputs that lead to the correct interpretation of motion. Loss or degradation of these inputs can result in a skewed perception of the surrounding environment [8].

Smooth Pursuit and Saccades

When viewing an object of interest, the eyes and head move in a coordinated fashion promoting a stable retinal image. The ability to track a moving object termed as *smooth pursuit* and can occur when the object velocity less than 80 to 90°/s [11]. When object velocities are greater than this, *saccades* are produced to reduce tracking errors and re-foveate the target [12]. These are discrete eye movements with velocities up to 700°/s [13].Typical visual tracking includes some combination of smooth pursuit and saccades [1412].

Smooth pursuit and saccadic eye movements help improve spatial resolution by reducing retinal smearing [15]. *Spatial resolution* is defined as the smallest detectable difference between two lines in an image of parallel lines. *Retinal smearing* is a property wherein the retinal images become blurry and are without definite shape or size. This occurs when the target object is not stable on the retina. Spatial resolution is improved and retinal smearing is minimized when images are focused on the fovea [15]. In fact, object recognition is reduced significantly during smooth pursuit and saccade if the target image is not stable with respect to the retina [15].

Vestibulo-ocular Reflex (VOR)

Vestibulo-ocular reflex is one mechanism to improve retinal stability. This reflex promotes a stable retinal image during active head movement. It works as follows: if the head is rotated to the left, the eyes automatically move to the right (Figure 6). This reflex is driven by inputs from the vestibular system and can be enhanced with visual cues [16].

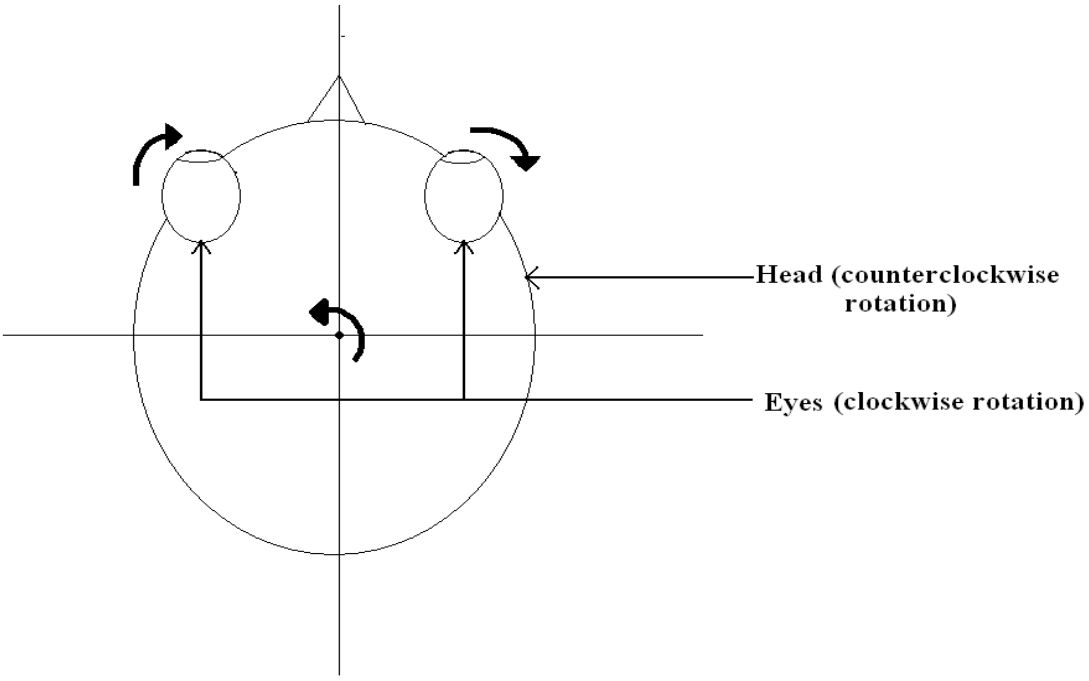

Figure 6 – The Vestibulo-ocular Reflex.

The gain of VOR can be calculated by dividing the eye angular velocity by the head angular velocity. For a normal individual, the VOR gain while focusing on a target at infinity (termed a far-target) is -1. The negative sign indicates the velocity vectors of the eye and head are in opposite directions and mathematically confirm the effort maintain a stable retinal image [16].

A number of experiments have been carried out to study changes in VOR gain during head rotations [18, 19]. Crane & Demer tested the VOR response in normal subjects. In their study, they asked subjects to sit in a chair which was rotated in the transverse plane (yaw) controlled by a 500N-m stepper motor. Magnetic search coils were used to measure the angular head and eye positions. The head position was measured using two

search coils – one taped to the forehead of the subject and the fixed to the bite-bar which was made to fit the subject's upper teeth. The eyes were topically anesthetized before using the ocular search coils embedded in an annular suction contact lens. The rotation velocity was controlled and was applied using four different acceleration rates. Subjects were also subjected to rotation around four different axes: 10cm anterior to the eyes, centered between the eyes, centered between the otoliths, and 20cm posterior to the eyes when asked to view targets at distance of 15cm, 30cm, and 500 cm.

The results from the study suggested that the absolute value of the VOR gain was always greater with far targets as compared to near targets regardless of the eccentricity of the rotation. Similar results were found by Viirre and Demer. Their average VOR gain was around -0.65 for the near target (0.1m) which increases to -0.85 for the far target (3m). Far targets were used in this study to maximize VOR gain. They were also more consistent with the real-world application of the results since most visual targets in gymnastics are at some distance to the gymnast.

Just as the VOR can be modified by providing the subject with a fixed visual target, VOR suppression or cancellation (VORc) can be promoted by providing a visual target that is linked to head motions [19]. VORc is the ability of the central nervous system to suppress vestibular inputs and result in eye movements that do not act to stabilize gaze on a stationary target. Theoretically, when the reflex is fully suppressed, the eye-in-head angular velocity would be zero, resulting in a VOR gain of zero. The ability to suppress the VOR may be important to twisting gymnasts. Although it does result in decrease

visual acuity and retinal smearing, it also more importantly allows the gymnast to view the landing zone prior to impact.

The Vestibular System.

The vestibular system not only provides input to the VOR, it is also responsible for the maintenance of equilibrium and balance [2]. The vestibular organ is located in the temporal bone of the skull adjacent to the auditory sense organ (cochlea). It forms the labyrinth of the inner ear and is subdivided into two parts: (1) the macula (utricle and saccule), and the semicircular canals (Figure 7).

Macula

There are two maculae, one on either side of the head. The macula is sensitive to linear acceleration and is composed of the utricle and saccule. Tilting the head forward or laterally increases neural activity in the ipsilateral utricle whereas tilting it backwards decreases this activity.

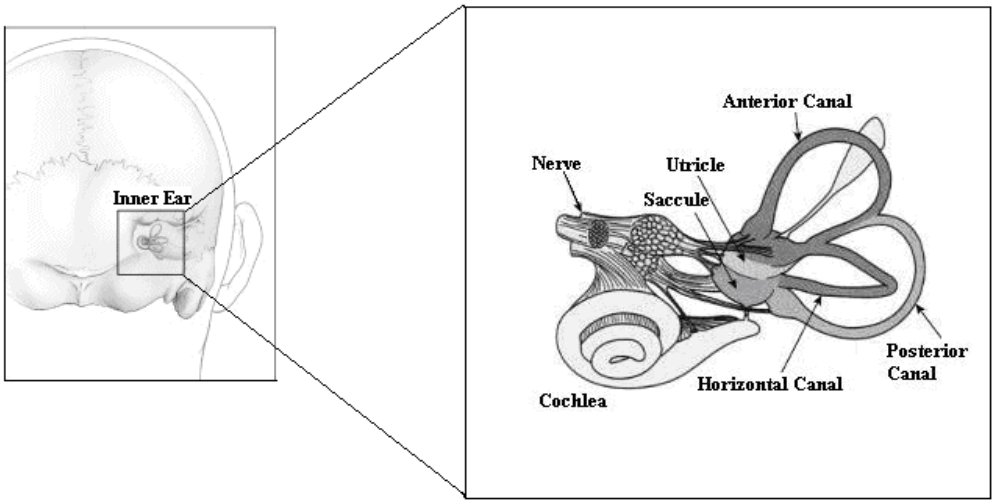

Figure 7 – The labyrinth of the inner ear.

The saccule responds to head movements in muliple directions i.e. pitch, roll and even upwards and downwards displacements of the head. The bilateral arrangement makes it possible for this sense organ to respond to movements in a variety of head orientations.

Semicircular Canals

There are three semicircular canals on each side of the head. These canals are positioned perpendicular to each other (orthogonal) and respond to rotational motions of the head. They are labeled *horizontal*, *anterior*, and *posterior*. The horizontal canal is oriented in a horizontal plane. The anterior and posterior canals are oriented in a vertical plane. The horizontal semicircular canals are responsible for detecting rotation along a vertical axis. These are the ones thought to be most impacted by gymnastic twisting elements since twisting occurs along a superior inferior axis. The anterior and posterior semicircular canals are responsible for detecting body movements in the saggital and

frontal planes respectively (See Figure 7). The anterior canal would be most sensitive to gymnastics flipping elements.

Each canal has a parallel counterpart on the other side of the head. This allows the vestibular system to operate in a fashion known as the push-pull mechanism [20]. For example if an individual turns his/her head to the right, the activity in the right horizontal canal increases while that of the left horizontal canal decreases. Using this difference to "sense" movement increases the sensitivity of the system by doubling the gain. It is this difference that is interpreted as movement. Similarly stimulations that excite the anterior canal of one side, inhibits the posterior canal of the opposite side and vice versa. Thus these opposite pairs produce signals whose difference indicates the net movement [21]. Figure 8 illustrates the push-pull mechanism in the form of a differential amplifier model.

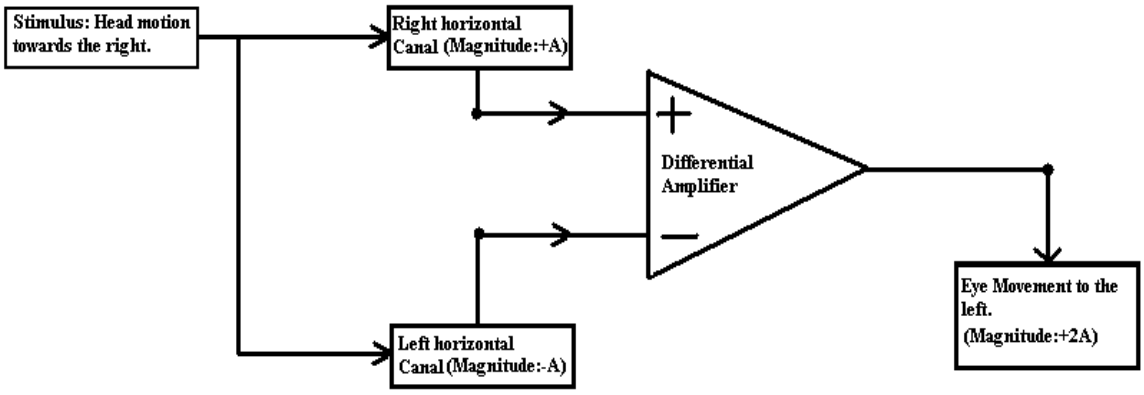

Figure 8 – Differential amplifier model illustrating the push-pull mechanism of the vestibular system. In this example, moving the head to the right results in stimulation of the right horizontal canal (+A). The left horizontal canal would be inhibited (-A). The summation of these inputs results in an output with a magnitude of 2A. The sign of the output would be reversed for head movements in the opposite direction.

The vestibular system can detect motion independent of from the visual system, but the two inputs combined provide information to the gymnast during the aerial phase of a skill. The final system to discuss is the proprioceptive system.

The Proprioceptive System

Proprioception, loosely defined, is a system that provides information about relative joint positions. It is a part of a closed-loop system composed of force generators (muscles) and active sensorimotor feedback controls (receptors – e.g. golgi tendon organs and muscle spindles), and is responsible for the maintenance of body posture [22]. In conjunction with the vestibular system, it plays an important role in maintaining balance during dynamic perturbation. In the twisting gymnast, most proprioceptive information is lost when they leave the floor and enter the airborne phase of the skill. At this point, the vestibular and visual inputs become more valuable in determining postion in space.

Systems Integration

A compromise in any of these systems can affect the balance of an individual [2, 22]. As an example, consider vertigo. Vertigo is a false feeling of self-motion when one is stationary. It usually occurs when there is an information mismatch between the visual, proprioceptive, and the vestibular systems due to a vestibular malfunction. Consider a normal individual who is at rest. Proprioceptive and visual cues provide information consistent with this state. The vestibular system should also support it due to its push-pull mechanism. The net difference between stationary left and right semicircular canals should be zero. If there was a malfunction in either side of the vestibular system, then the net output would not be zero. The individual would begin to experience a sense of motion. This mismatch between the visual (steady position) and vestibular cues (motion) results in the experience of vertigo or dizziness and can lead to spatial disorientation [23].

The ability to maintain balance deteriorates with age [2]. Some of the reasons for the loss of this ability in older individuals include failure to realize or inappropriate reaction to the center of mass displacement [24]. This example also illustrates the need to appropriately integrate sensory information and what happens if these data are compromised.

At the other end of the age spectrum are children. Infants who are learning to stand and move around rely heavily on visual cues as compared to mechanical proprioception [25]. Mechanical proprioception develops as a function of experience as the infant grows. Lee and Aronson came to this conclusion by running tests on seven healthy infants between the ages of 13 to 16 months. The experiment involved placing the subject in a three sided room (with ceiling) that could be moved in the forward and backward direction. With the opening to their back, their theory was that if the room swayed in the forward direction (i.e. away from the subject), the subject would perceive this as a backward sway of the body and would compensate with a postural muscular reaction to sway in the forward direction. But since the floor was stationary, the forward sway would result in loss of balance. The experiment was carried with both forward and backward wall movement with the predicted results. The authors felt that if the mechanical proprioception were developed, then the compensatory sway secondary to the visual cues would have been muted.

The point of these examples is that the sensory systems used to balance and navigate through daily tasks are dynamic in nature. Their performance can change as a result of the aging process or practice. Athletes involved in sports like gymnastics, diving, skating

etc. rely heavily on sensory cues to retain their balance during the performance of aerial skills. Training in their respective sport makes their balance maintenance system (visual, vestibular and proprioceptive systems) superior to their non-athlete counterparts [26]. It is thought that through this specialized training they acquire a superior understanding of the gain variation and reliability of these senses. In other words, they pay more attention to the sense that provides the most reliable data.

The proposed study focuses on female gymnasts whose routine training involves body rotations around one or more axis. It is feasible that this results in a difference between the leftwards and rightwards VOR responses due to training and that this difference is related to the direction they twist. Also, their ability to suppress the vestibulo-ocular response is expected to be correlated to their level of gymnastics performance.

Methods

Subjects

The study was approved by the Internal Review Board of Virginia Commonwealth University. Ten female subjects volunteered to participate. They ranged in age from 13 to 20 with a mean age of 15.5±2.2 years. They had participated in gymnastics for up to 14 years. Their ability level ranged from novice to elite with 7 of the 10 still active in gymnastics competition. They were consented on site and told that they could exit the study at any time without recourse. Table 3 outlines the study population details. Twist direction and hand dominance were acquire via self report. Twist direction was defined by the shoulder that moved backwards faster during the performance of a twisting skill. Eye-dominance was determined by asking each subject to point to a distance object with both eyes open. They were then asked to close each eye and report the sighting eye that had the least change in pointing finger movement. This eye was noted to be the dominant eye.

Subject	Age	Twist Direction	Hand Dominance	Eye Dominance	Level
AC	16	R	R	L	10
CR	20	R	R	L	10
KW	14	R	R	R	8
KGB	16	R	R		8
ML	16	R	R	R	10
RP	18	L	R	L	REC
EC	14	L	R	L	8
LW	13	L	R	L	7
AD	13	L	R	L	7
KB	15	-	R	-	REC

Table 2 – Demographic table of all ten subjects displaying their age, twist direction, eye-dominance, hand dominance, and gymnastics level-of-competition. Note that a higher "level" number indicates a more advance gymnast. (REC stands for recreational gymnast).

Experimental Setup

All data were collected using a kinematic system to monitor head and trunk position with 6 degrees-of-freedom (dof) and a binocular eye tracking system to monitor eye position with 2 dof. Data collection with these two systems were integrated and provided a synchronous data set. The subjects were seated on a wooden stool at the end of a 0.762m high workbench. They were asked to wear a helmet fitted with the camera-based infrared eye-tracker (EyeLink II™, SR Research Ltd. Mississauga, Ontario Canada.). The eye-tracker had a tracking range of ±30° in the horizontal direction and ±20° in the vertical direction with accuracy of 0.5°. The data was collected at 250 Hz and provided eye-in-head position. Head and trunk position were collected from two electromagnetic (EM) motion sensors (Motion Monitor™, Innovative Sports Training,

Chicago, Illinois, USA). The first was on the helmet that also held the eye tracking hardware. This provided head-in-space position. The second was mounted on a strap worn around the subject's thoracic area (T12 joint). This provided a reference to link the head-in-space data to the trunk so that orthopedic angles at the neck could be determined. Data were collected from the kinematic system at 100 Hz. This system has a linear resolution of 0.5mm and an angular resolution of 0.1°.

A transmitter was located behind the subject and provided orthogonally oriented electromagnetic fields. The theory of operation for this type of sensor is that it orients itself based on transmitter field strength. The field strength decreases as a square of the distance from the transmitter. The transmitter used in this application had a functional radius of 10 feet. The subjects (and sensors) were located well within this range. Care was taken to avoid the presence of metal within the transmitter field to minimize eddy current distortion. To further reduce any effect of metal, a mapping procedure was performed prior to data collection. This metal mapping procedure uses known sensor location data to construct a distortion map of the collection space. These data are then used to linearize any measurement error across the mapped space. All collected data were stored in a coded file using subject initials, the date, and trial number/descriptor.

Subject Setup

After securing the helmet and sensors, the subject was asked to sit steady and face an LED display board placed on the workbench at a distance of 1 meter. The subject was approximately eye-level with the center target on this board. A world-based right-hand coordinate system was defined on the workbench top. The origin was located to the

subject's right. The subject faced the positive Y direction. Positive X-axis was defined towards the left side of the origin and positive Z-axis going down to the floor from origin. Subject landmarks were defined in a prescribed manner using a stylus fixed to a free floating sensor. These standardized locations allowed the reconstruction of a rigid body model of the subject for reconstruction and interpretation of position data.

The digitized locations included the back of the head (occiput), the C7 spinous process, and the T12 spinous process. These points provided reference marks for the reconstruction of the head and trunk in the collection space. Left-right symmetry was assumed. The locations of the subject's eyes were also digitized. This was done by gently placing the stylus tip at the center of the subject's closed eyelids and the bridge of the nose. These landmarks located the eyes in the head and provided an estimate of the interpupillary distance. This distance estimate was later improved during the calibration/validation procedure by having the subject focus on a moving stylus while performing a vergence activity. This is discussed in the next section.

Calibration and Validation

With the subjects head, trunk, and eyes located, the next step was to calibrate the eye tracking system. This was done by asking subjects to visually identify 9 fixed targets on the display board in front of them. The targets were red LEDs configured symmetrically on a grid 0.493 m wide by 0.477 m high (Figure 9). At the 1 meter distance, this provided a view field of 26° horizontally and 27° vertically. Each LED occupied a visual angle of 0.03° at the 1 meter distance. Calibration was semi-automated. Each LED was lit in a predefined sequence. The LED remained lit until stable gaze was

detected for at least 1000m. Then the LED was extinguished and the next one lit. Left and right eye gaze were calculated independently using a combination of head-in-space data and eye-in-head data. Subjects were instructed to minimize head movement in order to maximize the visually calibrated area, but head movement did not negatively impact the calibration in any other way.

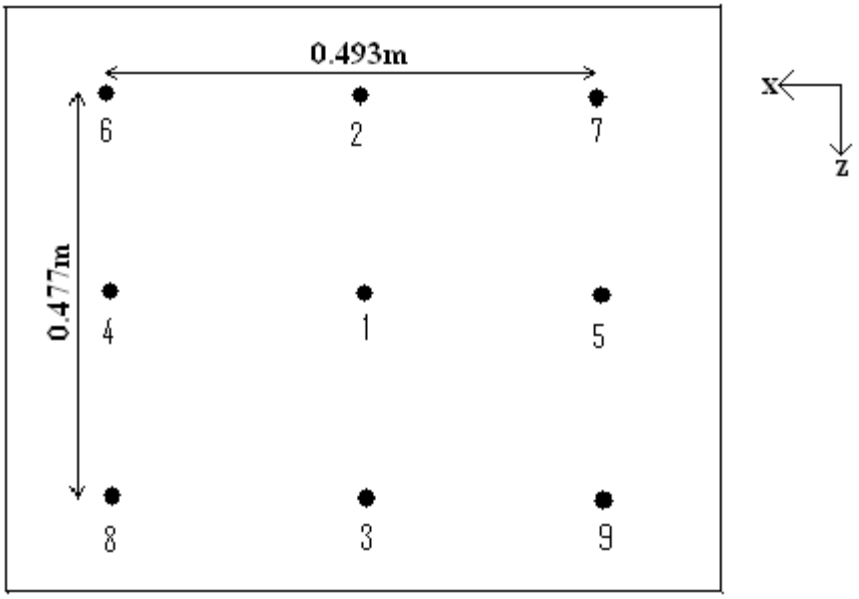

Figure 9 – Calibration screen with numbers representing the order of target presentation.

Immediately following calibration, a validation target sequence was presented to the subject. The same LED targets were again illuminated in a predefined sequence different from the calibration sequence. Left and right eye gaze data were compared to the left and right eye calibration data and singular targets recollected up to three times if the validation minus calibration gaze error was greater than 0.5°. Once again, targets were illuminated until stable gaze was achieved for about 1000ms. At the completion of

the validation process, an error map for each target was presented to the experimenter. The calibration/validation process was repeated if the average error for all targets was greater than 0.5°.

The next step in the calibration/validation process was to improve on the digitized interpupillary distance estimate. The subject was asked to follow the tip of the stylus as it was moved along the Y axis (toward and away from the subject at eye level). With the assumption of the left and right eye equally fixed on the stylus target, the locations of the eyes were calculated using the vergence angle data. RMS errors betweent the known target location and the point-of-gaze were typically reduced following this process as the interpupillary distance was refined.

This combination of eye tracker and head tracker has been found both reliable and valid. A previous study using this system found an RMS error of 0.45° ± 0.12° across subjects. Subjects repeatedly located targets with a standard error of the mean (SEM) of 1cm in the horizontal direction. At a target distance of 1m, this presents a visual angle error of 0.57° [27].

Once the system was calibrated and validated, four experiments were performed. These experiments included: (1) a nine point validation, (2) a characterization of the horizontal saccadic system, (3) an assessment of the VOR system, and (4) an assessment of volitional VOR suppression. The nine point validation (NPV) of the eye tracker was performed repeatable in an effort to ensure the continued integrity of the data since portions of data collection required vigorous head rotations. Although the repeated validations of the eye tracker were really a data quality control measure, they are labeled

as a separate experiment for consistency. The experimental sequenced is illustrated in Figure 10. Each experimental protocol is detailed below.

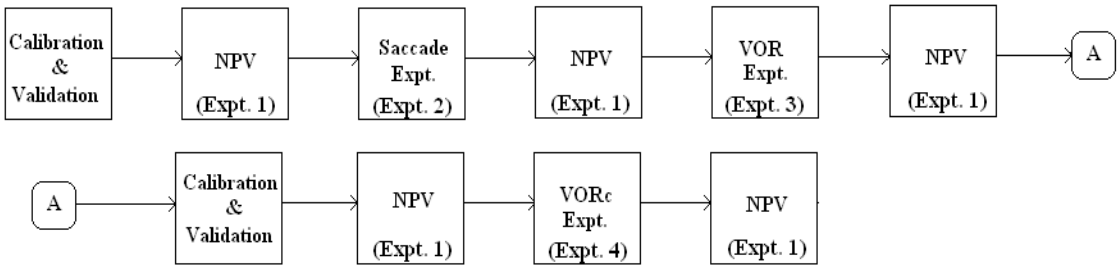

Figure 10 – This is the presentation sequence of each experiment. A validation protocol was performed (Experiment 1) multiple times to improve the quality of the data. Subjects were allowed to take a rest at any time during the presentation of experiments. If the eye tracking headset was removed, then a recalibration/validation was performed prior to resuming data collection.

Required Calculations

Before describing each experimental protocol, it is easiest to first define and describe common calculation methods. Comparing the visual responses of the subjects during each experiment required the calculation of gaze vectors and target errors.

Calculation of PoG

PoG is the point of intersection of the two vectors originating from both eyes of the subject (Gaze Vectors). It does not necessarily lie on the plane of the targets. It could lie ahead or beyond of the plane of targets. The plane of targets was located at a distance of 1m from the origin (along Y-axis, parallel to the X-axis).The data provided by both the eye-tracker system and the electromagnetic sensor system were the locations, gaze vectors, horizontal and vertical eye angles for both eyes. Using these variables and some formulae, the PoG can be calculated. The following equations were used to do the same:

$PoG(Y) = (b_r - b_l)/(m_l - m_r)$……. (Equation 1)

Where ‘r’ is a subscript for right eye and ‘l’ is a subscript for left eye. ‘m’ is the ratio between x co-ordinate and y co-ordinate of the gaze vectors of an eye. ‘b’ is given by the following:

b = |y co-ordinate of eye position|*m + (x co-ordinate position of the same eye).

PoG(X) = m*(PoG(Y)) + b)……. (Equation 2)

For calculating the PoG(Z), many intermediate variables had to be calculated.

a) Vertical Gaze angle (Θ):

   $\Theta = \tan^{-1}$(z co-ordinate of gaze vector/ y co-ordinate of the same)

b) Vertical Offset (VO): The Vertical Offset is the vertical distance (above or below) of the observed target from the eye level. It is basically calculations done assuming the eye is in the X-Y plane.

   VO= (Difference between the y co-ordinates of the PoG and the position of an eye)*tanΘ

c) Vertical Intercept (VI): Here we translate the X-Y plane from the eye level to the original X-Y plane of the world axes by considering the Z co-ordinates of the eye position.

   VI = VO + z co-ordinate of the eye position.

Finally after solving the above mentioned equations, PoG(Z) was calculated:

$PoG(Z) = (VI_l + VI_r)/2)$……. (Equation 3)

Hence, [PoG(X), PoG(Y), PoG(Z)] denotes the 3-D co-ordinates of the point of gaze for the subject.

Cyclopian Eye Calculation

Cyclopian eye (CE) value is the average of the left and right eye positions. In other words the location of the cyclopian eye is exactly in between the two eyes of the subject.

CE Co-ordinates = (Left Eye Co-ordinates + Right Eye Co-ordinates)/2→ (Equation 4)

Point of Interception (PoI) Calculation:

The gaze vector intercepts the plane of targets (y=1) at a particular point. This point ideally should be that target's location. Intercept error is a 2-dimensional quantity, hence we do not consider depth here (y co-ordinate).

The line-plane intersection equation is given by:

$(x, y, z) = (x_1, y_1, z_1) + r * ((x_2, y_2, z_2) - (x_1, y_1, z_1))$ ……. (Equation 5)

If the value of y_3 is known, then the value of r could be determined by solving the equation:

$y_3 = y_1 + r*(y_2 - y_1)$ ……. (Equation 6)

In our case, y_3 is the y co-ordinate of the target board. Point '(x_1, y_1, z_1)' is the cyclopean eye and point '(x_2, y_2, z_2)' is the PoG. Thus the value of 'r' was calculated and also the point of intersection of the gaze vector and the target plane (point (x, y, z)).

Interception Error Calculation:

Intercept error is the difference between the point (x, y, z) and the co-ordinate position of the respective target. Only x and z co-ordinates are being considered here. Thus this error is termed as the 2-Dimensional Error.

Intercept Error = Point of Interception – Target Position……. (Equation 7)

Experiment 1: Nine Point Validation (NPV).

The aim of NPV was to re-validate the system calibration in such a way that could be easily repeated between the other three experiments. This allowed the integrity and the reliability of the system to be monitored throughout the data collection process. Nine targets were used in this experiment that were positioned centrally and at visual angles of ± 11°. Each target was a yellow circle which subtended a visual angle of 1.1°. They were located within the LED calibration field and were numerically labeled (Fig. 11). The subject was asked to visually gaze at each of these targets at a self defined pace in a left-to-right and top-to-bottom. The subject was also asked to call out the numbers while looking at each target, starting and ending on the center target. The POI during fixations was calculated and compared to the known target locations. The error between these paired points was used to assess system integrity. Although quantitative assessment of real-time data was not possible, subjective evaluation of NPV data immediately post-collection did provide a mechanism to evaluate system performance. If the NPV data appeared distorted in any way, a system re-calibration and validation was performed.

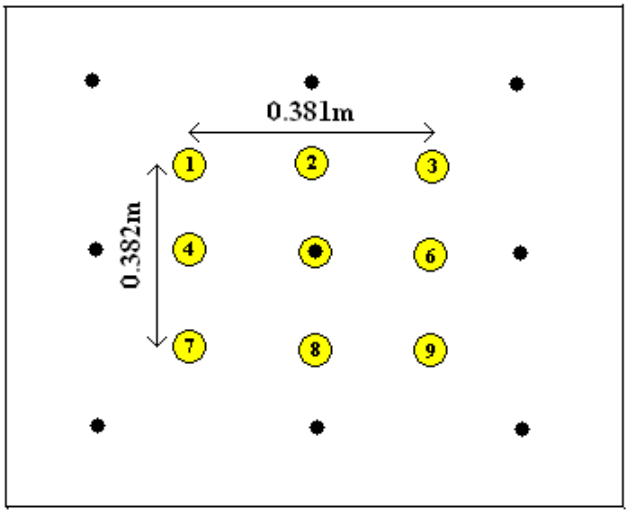

Figure 11 – The arrangement of the nine fixation targets for the NPV (marked yellow).

Data Analysis

Horizontal Eye Angle

Before calculating the gaze POI relative to the nine point target locations, the temporal locations of the fixation periods within the data stream had to be determined. These fixation "windows" were obtained from the horizontal eye angle data from the left eye. Figure 12 illustrates typical data. The nine identified fixations are highlighted in yellow with obvious saccades on either side as the subject moved from one target to the next. Fixations were defined as the data between 200ms and 300ms following a targeting saccade. The initial 200ms delay was to allow the eye position to stabilize. Each fixation was intended to represent a period of time where gaze was relatively stable. The start point and the end point frame numbers of the respective fixations were saved for use in later calculations.

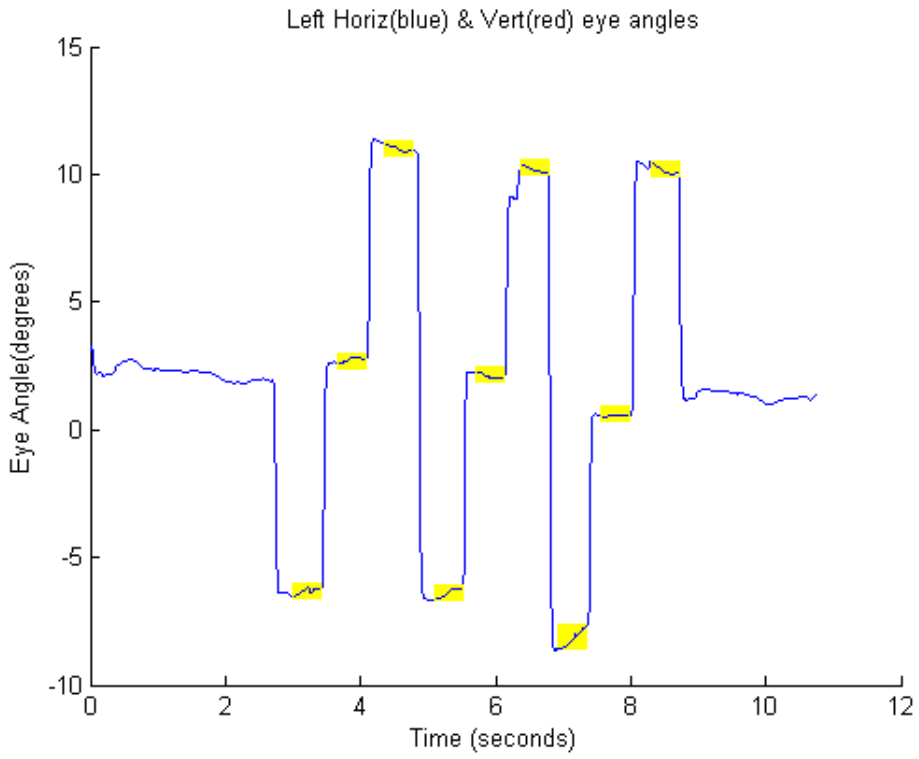

Figure 12 – Temporal plot of the left horizontal eye angle as the subject was asked to visually fixate on nine targets on the calibration board. The yellow bars represent the fixation data.

From the head-in-space and eye-in-head data, the POI of the left and right gaze vectors with the calibration plane were calculated for each fixation period. These two intersections were spatially summed and compared to known target locations. A typical result is illustrated in Figure 13.

LED location Vs PoI

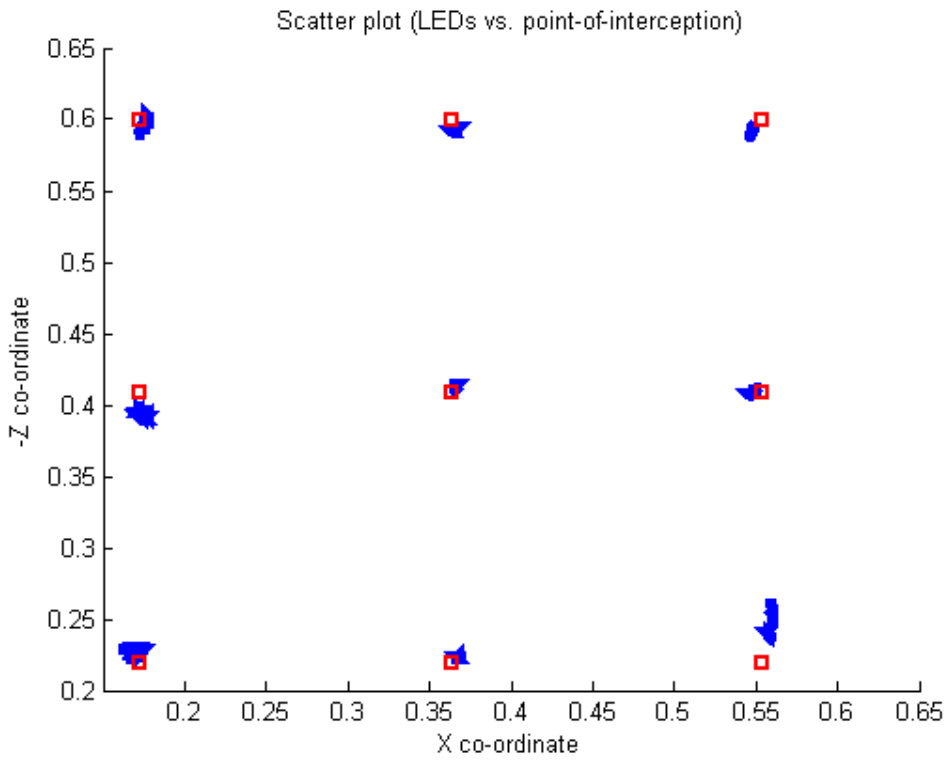

Figure 13 – The graph shows the actual location of the target (red squares) and the calculated POI (blue dots).

Recall that the main objective of the NPV was to verify the consistency of the data. During post-hoc analysis NPV data from before and after experiments 2, 3, and 4 were compared. Since this protocol is serial in nature, the post-NPV for one experiment is also the pre-NPV for next experiment (Figure 10). If the absolute summed difference between the pre- and post-NPV errors exceeded ±0.82°**,** then data from that experiment was considered compromised and was removed from analysis. This range is a function of

the resolution of each measurement device and the size of the target being viewed. It is detailed later in the Target Foveation section. No data had to be excluded based on these criteria.

Experiment 2: Volitional Saccades

The aim of this experiment was to study the temporal variations of volitional horizontal saccades. A volitional saccade can be defined as an eye movement that quickly jumps from one fixation point (or target) to another under the control of the subject [28]. The targets in this case were two LEDs located on the calibration frame 0.381m apart (at points 4 and 6 illustrated in Fig. 11). The visual angle between these targets was 22° and each LED subtended a visual angle of 0.03°. At the start of data collection, the LEDs were lit in an alternating pattern with transition delays ranging from 550 to 1500ms. This made the illumination pattern appear unpredictable to the subject and they were instructed to follow the target as accurately as possible. There were 20 lighting changes spanning a 50s data collection window. There was never a time when both targets were extinguished. The starting LED (left or right) was randomly assigned and the subject was asked to repeat the series four times. A typical response to this stimulus can be seen in Figure 14 (a and b).

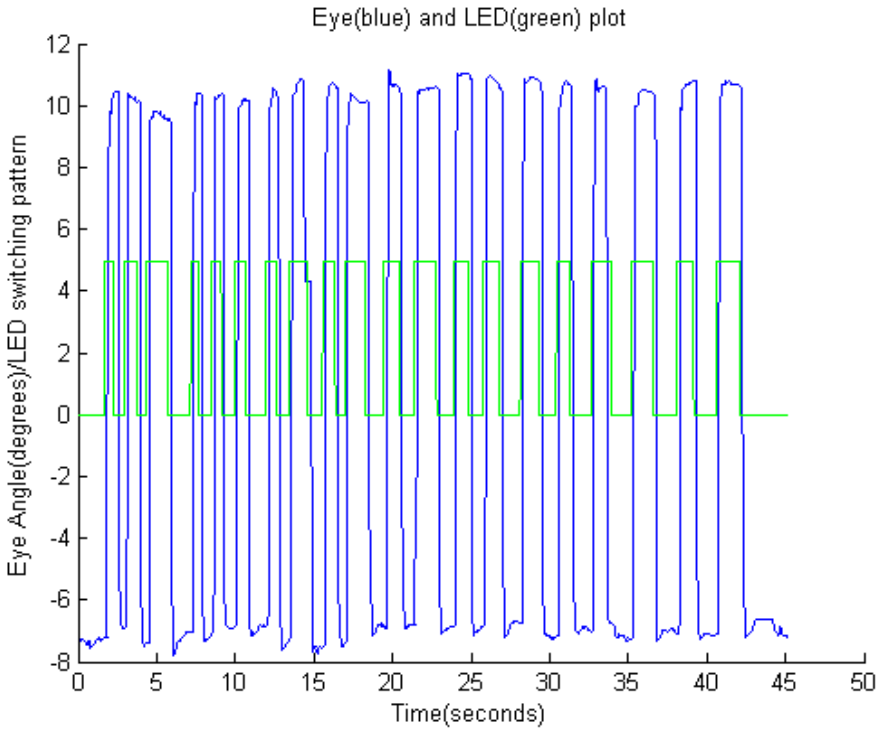

Figure 14a – Temporal plot of the eye angle with the alternately switching LEDs: Here positive indicates right LED and negative indicates left LED.

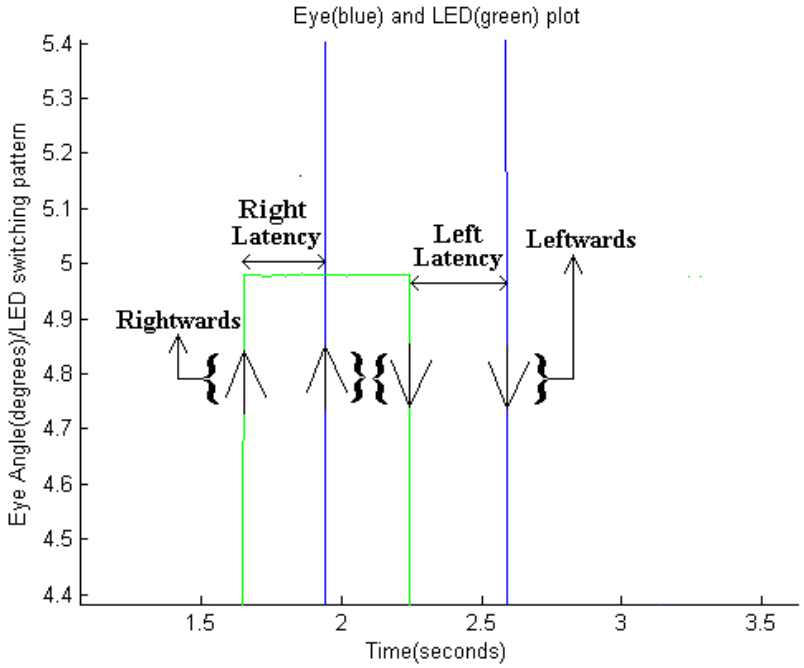

Figure 14b – Exploded view of saccade response towards both directions.

Data Analysis

Saccadic responses were separated into left and right directions. Left eye data were used to calculate horizontal peak velocity, temporal latency, and settling time. Bilateral eye data were used to determine the error between POI and the LED targets. Each variable and how it was computed is described below.

Peak Velocity

The peak velocity of saccadic eye movement has been found to range between 300 to 600degrees/second [29, 30]. It can be computed by applying central difference formula to eye angular position data during the performance of a saccade.

Eye velocity = (Eye angle(i+1) - Eye angle(i-1))/2*(sample rate) ……. (Equation 6)

Left and right velocities were separated based on the direction of eye movement following a target change. For each of these velocity segments, the peak velocity, the average velocity, and standard deviation were computed and stored.

Latency

Latency is defined as the time between the lighting of a target and the subject's initial response to fixate on that target. Again, an example of typical data can be seen in Figure 14b. Saccadic data sometimes contained blinks. These were identified and removed manually with a linear data fit through the region where the blink occurred. The temporal difference between the illumination of the LED (stimulus) and the eye movement (response) is considered the latency between the two events. Means and standard deviations for the leftward and rightward saccadic latencies were calculated.

Settling Time

When a subject performs a saccadic eye movement to view a specific target, the eye make the rapid eye movement followed by a series of adjustments to foveate the target. The time from the beginning of a saccade to the end of this adjustment is defined as the *settling time*. In these experiments, it was defined as the time taken by the subject to reduce visual error (LED location – POI) to an absolute value of less than 1.64° (i.e. ± 0.82° as discussed in Target Foveation later). A typical example of this is illustrated in Figure 14c.

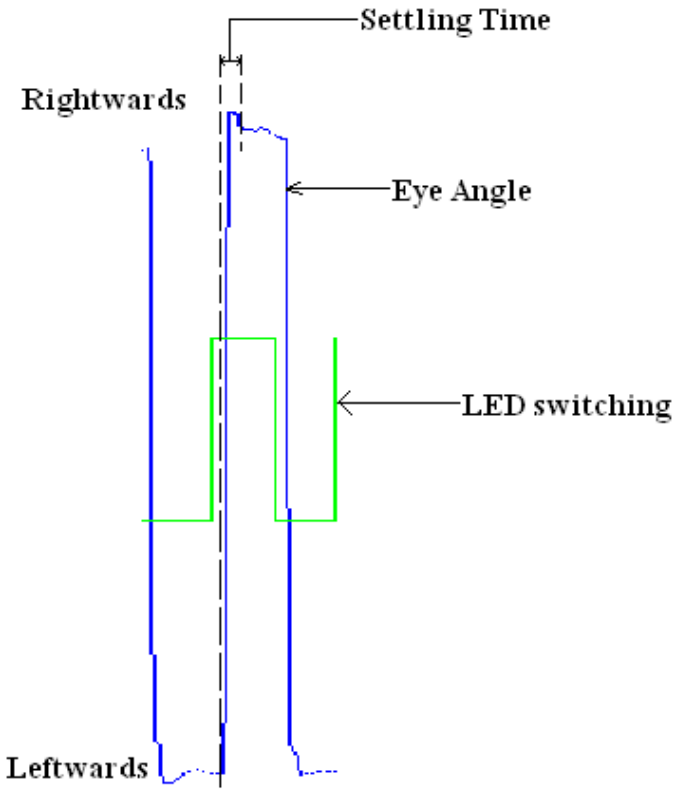

Figure 14c – Exploded view of saccade response showing settling time duration.

Error

Error is defined as the difference between the stimulus (LED location) and the response (subject POI).

Experiment 3: Vestibulo-ocular Reflex (VOR)

In this experiment, the subject was asked to visually fixate on a single lighted LED target. The target was located centrally at point 5 on the calibration screen (Figure 11) and subtended a visual angle of 0.03°. The subject was instructed to move her head in the transverse plane (yaw or "no" motion) while continuing to fixate on the target. The head movements were paced by a metronome at 7 different frequencies ranging from 72 to 196 beats/min with increments of 20 beats/min. The subject was instructed to change the direction of head movement with each beat and to traverse an angular distance of approximately ±15°. This combination of tempos and angular range should result in angular velocities between 35 and100°/s. Each trial consisted of 5 complete rotation cycles lasting at most 20 seconds (with the lowest tempo). Subjects were verbally encouraged to maintain the prescribed angular head range during the higher tempos to provide a full range of angular head velocities. Following each trial, head range of motion was reviewed and the trial repeated if necessary.

Frequency (clicks/min)	Frequency (Hz)	Expected head velocity for a ±15° oscillation amplitude (deg/sec)
72	1.20	36.00
92	1.53	46.00
112	1.87	56.00
132	2.20	66.00
152	2.53	76.00
172	2.87	86.00
196	3.27	98.00

Table 3 – Table showing the different frequencies along with the respective expected head velocities.

Data Analysis

Transverse plane head angular data and horizontal eye angular data were used to analyze the VOR response. These data were zeroed based on start position by subtracting the mean of 500ms of non-movement initial data from the entire trial. A typical example of these data is illustrated in Figure 15a. A comparison plot of head angle vs. eye angle reveals the inverse relationship of these data (Figure 15b). Left and right head movement data were separated for analysis (positive head angles were leftwards and negative head angles were rightwards). Each variable and how it was computed is described below.

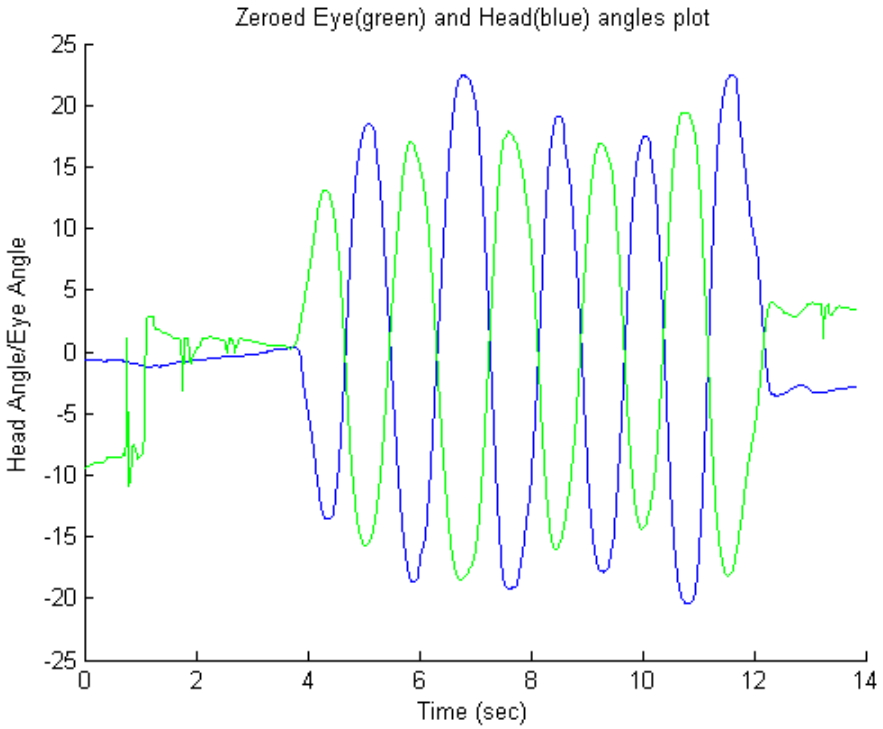

Figure 15a – Temporal plot of head and eye angles.

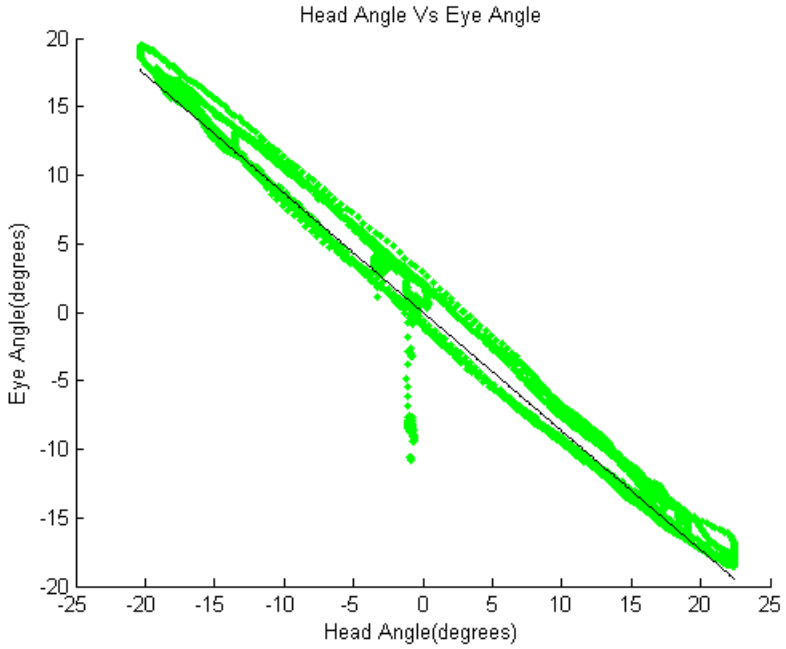

Figure 15b – Comparison plot between head and eye angles.

Angular Velocity Calculation

Angular Velocities of the head and eye were calculated from the head and eye angles respectively using the previously described central difference formula.

VOR gain Calculation

VOR gain is defined as the ratio between head velocity and eye velocity. Theoretically, for fixation targets at infinity, the VOR gain should be -1 since the head motion induces the eyes to move at the same angular velocity in the opposite direction [17]. This promotes stable target presentation on the retina. An illustration of typical gain data (plotting angular head velocity against angular eye velocity) can be seen is Figure 15c. In this figure, the gain is represented as the slope of a linear regression line through the data. Each trial has a singular total gain value calculated.

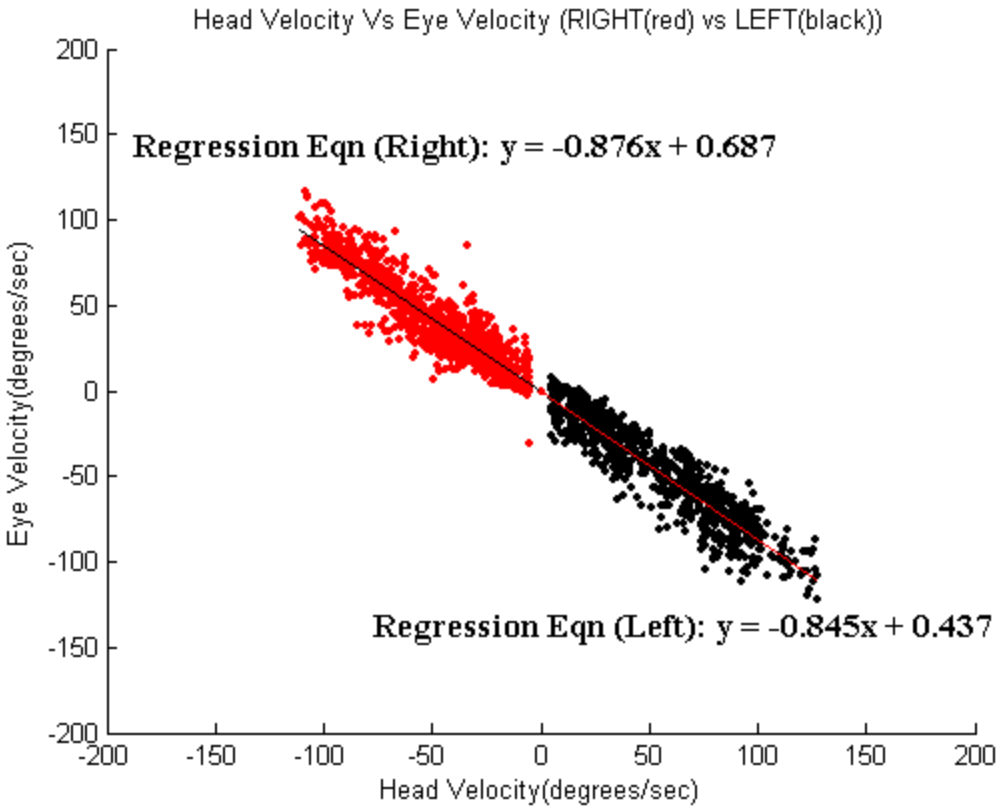

Figure 15c – Comparison plot between head and eye velocities. The regression lines illustrate the VOR gains. Data below 5°/s were excluded from the VOR gain calculation since the VOR response is negligible at low head velocities.

In addition to total gain, the gains for leftward movement and rightward head movements were also calculated. Each trial has a singular leftward movement gain and a singular rightward movement gain. The distinction of left and right movements were made based on the direction of head movement i.e. positive angular directions were labeled right and negative were labeled left.

VOR gains from all trials were compartmentalized and summed based on the peak head velocity achieved during that trial. Each compartment was 10°/s wide and ranged from 0°/s to the peak head velocity recorded for that subject.

Target Foveation

The fovea represents the central 1° of visual angle [10]. Subjects typically place images there since this area of the retina has the highest density of optical receptors [31]. The VOR promotes this. Since this experiment involves tracking the POI relative to a known target location and the location of the eye-in-space is known, the location of that target on the retina can be calculated with simple trigonometry (Figure 16). It is expected that this experiment will induce retinal smearing or a blurring of the image on the retina as the subject attempts to maintain foveation of the target. *Target foveation* time can be expressed as a percentage of total data collection time and was calculated for all trials. To account for the compound error associated with the eye tracker and kinematic systems, foveation was defined as targets within the central ±0.82° of visual angle (fovea ±0.5° + EyeLink II™ ±0.25° + Motion Monitor™ ±0.05°+ Target angle ±0.015).

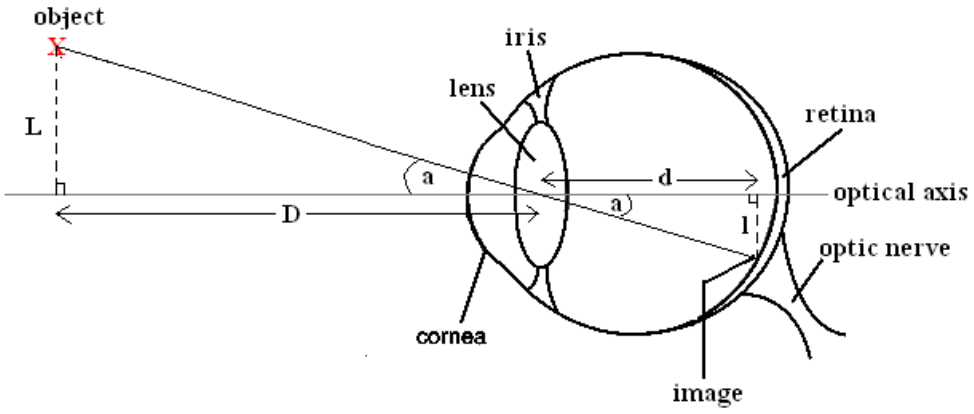

Figure 16 – Trigonometric relation to find the location of an image on the retina of an observer.

Experiment 4: Vestibulo-ocular Reflex Suppression/Cancellation (VORc).

In this experiment, the calibration display was replaced with a black tri-fold foam board (0.914m high X 1.22m wide). The center section was 0.914m x0.61m and subtended a visual angle of ±17° and was parallel to the calibration display. The two end folds were angled slightly toward the subject for support. A laser pointer was fitted on the subject's helmet such that when the subject faced the black board; a red dot appeared on the center area of the board. The dot was smaller than the fixation LED used in the VOR experiment and subtended an angle of ≈ 0.01°.

The subject was instructed **to** perform the same head movement described in the VOR experiment, but this time the subject was asked to focus on the laser target. Since the target was linked to head movement, successful visual fixation would result in VOR cancellation (VORc).

Data Analysis

Once again, transverse plane head angular data and horizontal eye angular data were used to analyze the VOR response. These data were zeroed based on start position by subtracting the mean of 500ms of non-movement initial data from the entire trial. A typical example of these data is illustrated in Figure 17a. A comparison plot of head angle vs. eye angle reveals an uncorrelated relationship between these data (Figure 17b). Left and right head movement data were separated for analysis (positive head angles were leftwards and negative head angles were rightwards). Each variable and how it was computed is described below.

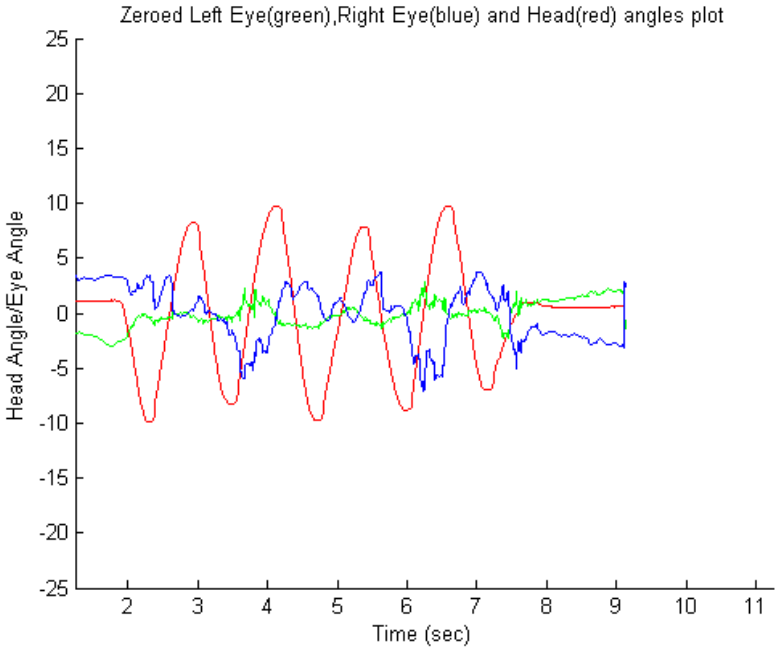

Figure 17a – Temporal plot of head and eye angles. Note the sinusoidal head movement similar to the VOR experiment, but in this case the eye angle remains relatively constant and near zero. This suggests that VOR suppression is achieved by the subject.

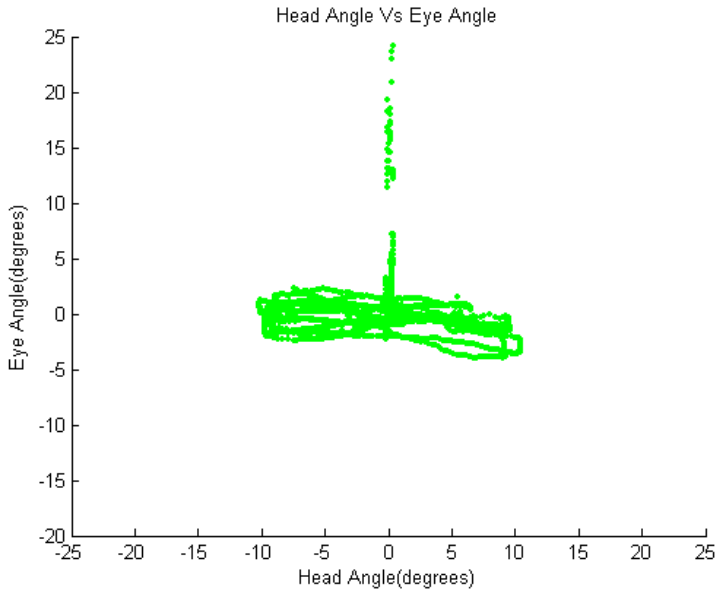

Figure 17b – Comparison plot between head and eye angles. Note the near constant eye angle throughout the range of head angles.

VORc gain Calculation

VORc gain is calculated just like the VOR gain. It is defined as the ratio between head velocity and eye velocity. Theoretically, for a head mounted target, the VORc gain should be zero [32].An illustration of typical suppressed VOR gain data (plotting angular head velocity against angular eye velocity) can be seen is Figure 17c. In this figure, the gain is represented as the slope of a linear regression line through the data. Each trial has a singular total gain value calculated.

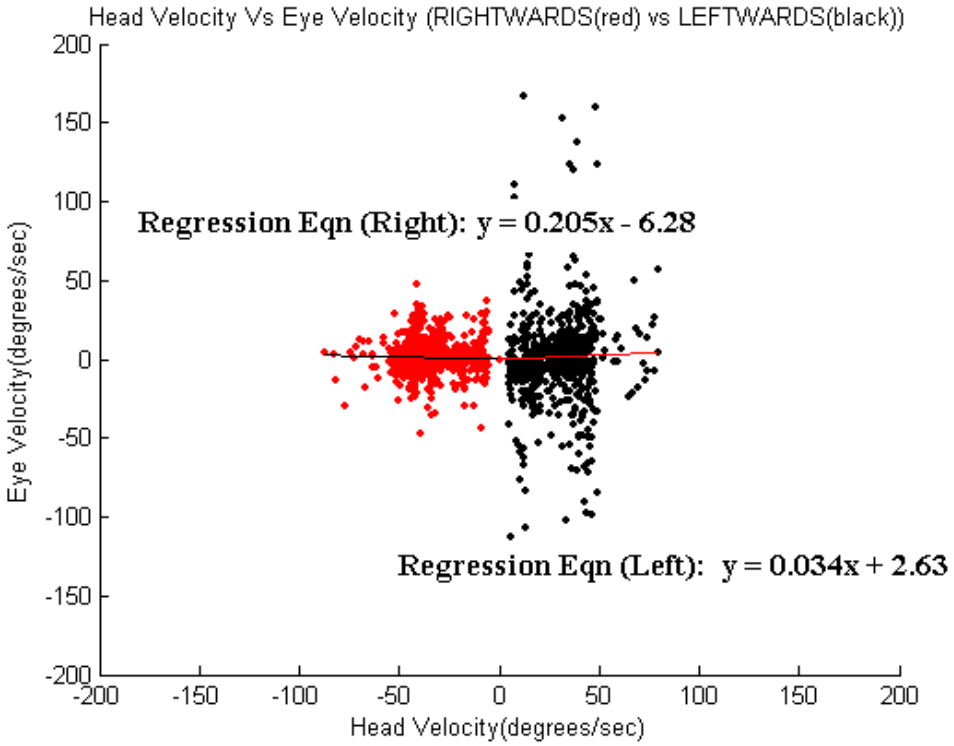

Figure 17c – Comparison plot between head and eye velocities. The regression lines illustrate the VORc gains.

VORc gains from all trials were compartmentalized and summed based on the peak head velocity achieved during that trial. Each compartment was 10°/s wide and ranged from 0°/s to the peak head velocity recorded for that subject.

Target Foveation

Target foveation was computed in a manner similar to the VOR experiment previously described.

VOR and VORc Indices Calculation

The main purpose of calculating the VOR and VORc indices was to estimate the time taken by the subject to begin suppressing the VOR. Each trial consisted of the subject moving their head in a yaw motion ten times. The peak head velocities from each end-to-end motion was calculated and data was classified in bins of 10°/s intervals from minimum head velocity to maximum head velocity range. Since all subjects had complete responses for both VOR and VORc within the 65-75°/s head velocity bracket, we chose to consider this bracket for further analysis. The mean head velocities and eye velocities for all trials within the range of 65-75°/s were passed through a cumulative integrator to calculate the area under their respective curves. Ideally the index for the VOR would be zero since the eye velocity response is equal and opposite head velocity. The index for VORc would be expected to start at zero indicating an initial VOR response and then increase almost linearly indicating the cancellation/suppression of VOR. The time at which the VORc index begins to deviate from zero (visually estimated) is compared to the trial start time and the difference is termed as the latency of suppression of VOR response. This is illustrated in Figure 18.

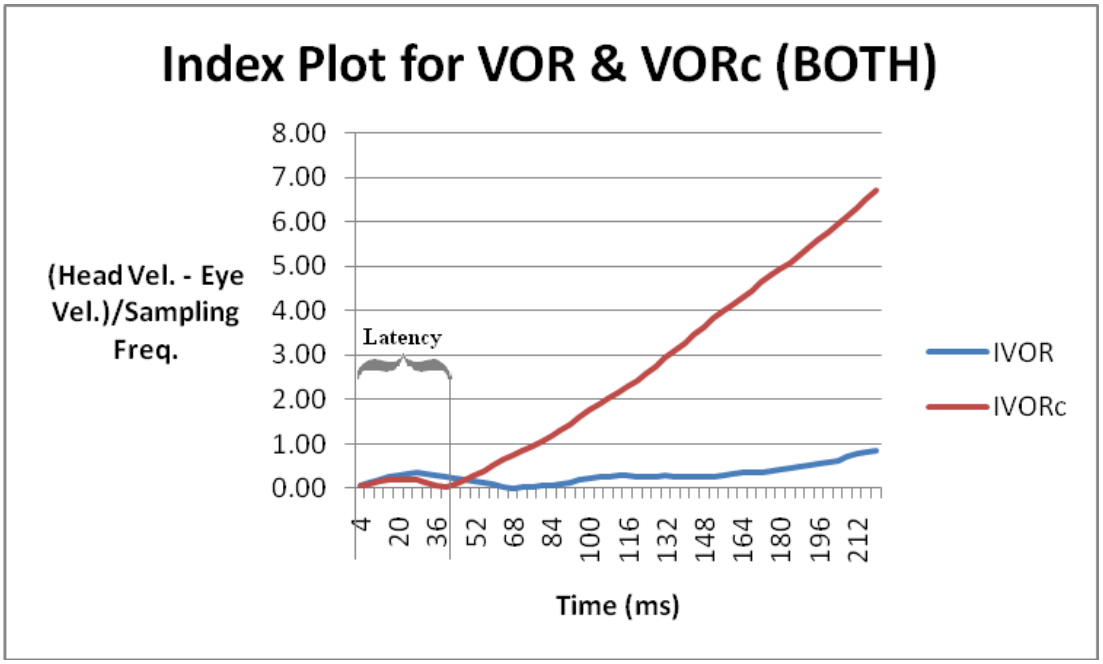

Figure 18 – Index Plot showing the latency period for the subject to initiate the cancellation of VOR.

Results

During post-hoc evaluation of the data it became apparent that there was a ranking in the VOR cancelation gain that paralleled performance metrics in the gymnasts. Gymnasts with higher performance levels had a greater ability to cancel VOR. As part of data analysis, the 10 subjects were subsequently divided into 3 groups arranged with respect to their VOR cancellation gain. These were labeled Group I (n=3), Group II (n=3), and Group III (n=4). Descriptive data are provided for individual and pooled variables in addition to these groups.

Experiment 1: Nine point data collection (NPV)

There are no results to present for this experiment since these data were used in quality control.

Experiment 2: Volitional Saccades

Latency

The temporally directed saccadic latencies were compared with their nasal counterparts using a paired t-test (SPSS). There were no significant differences found ($p>0.05$) when evaluating by subject or population. This implies that there was a nasal/temporal performance symmetry. These results are shown in Table 4a and Figures 19a and 19b. Grouped results also showed no significant difference ($p>0.05$) and can be

found in Table 4b. It has been suggested that there can be latency assymmetries between nasal and temporal eye movements[28]. This may be due to innervation differences or differences in the elastic properties of the optic muscles. This research has not been confirmed by others.

	Latency (sec)							
	Left Eye				Right Eye			
Subjects	Temporal		Nasal		Temporal		Nasal	
	Mean	s.d.	Mean	s.d.	Mean	s.d.	Mean	s.d.
AC	0.19	0.03	0.19	0.04	0.19	0.05	0.19	0.03
CR	0.17	0.02	0.19	0.03	0.19	0.03	0.18	0.02
KW	0.21	0.04	0.24	0.07	0.24	0.07	0.21	0.04
ML	0.20	0.05	0.21	0.05	0.24	0.09	0.23	0.09
KGB	0.19	0.04	0.18	0.06	0.19	0.04	0.19	0.06
RP	0.21	0.04	0.21	0.06	0.22	0.07	0.21	0.05
EC	0.21	0.06	0.20	0.05	0.20	0.05	0.21	0.05
LW	0.23	0.06	0.26	0.09	0.23	0.05	0.23	0.07
AD	0.21	0.05	0.23	0.07	0.22	0.07	0.22	0.08
KB	0.22	0.06	0.20	0.04	0.18	0.05	0.22	0.06
Mean	0.20		0.21		0.21		0.21	

Table 4a. Saccadic latencies for all ten subjects.

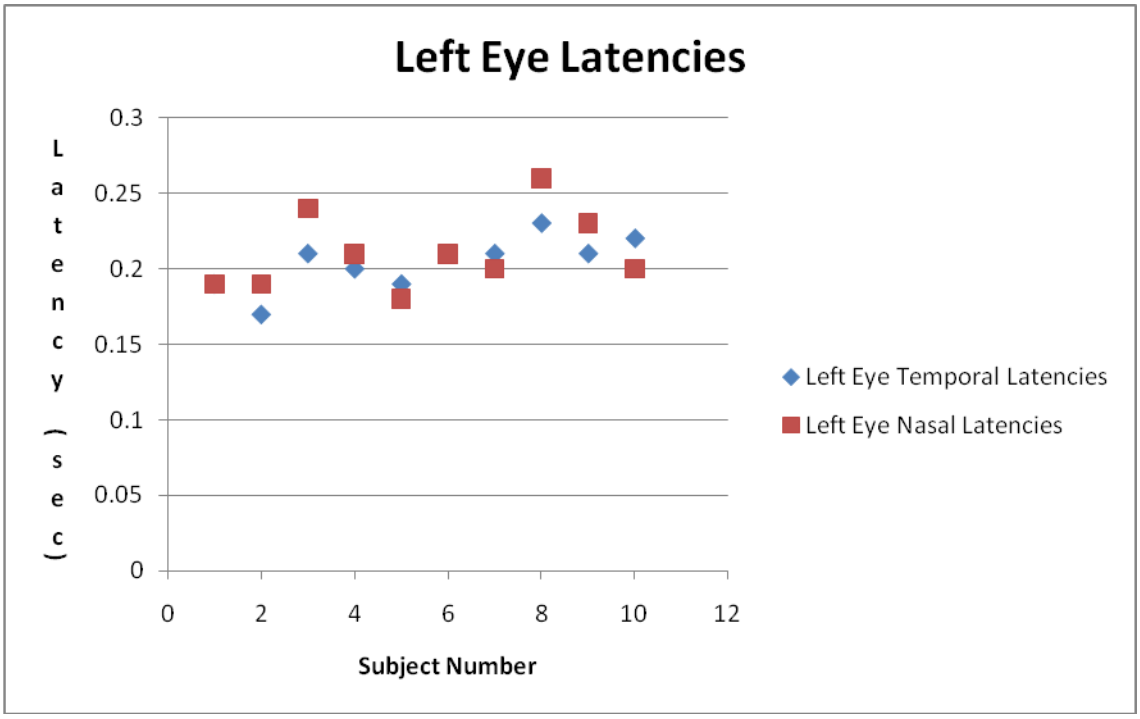

Figure 19a – Nasal and temporal latencies of the left eye of all subjects

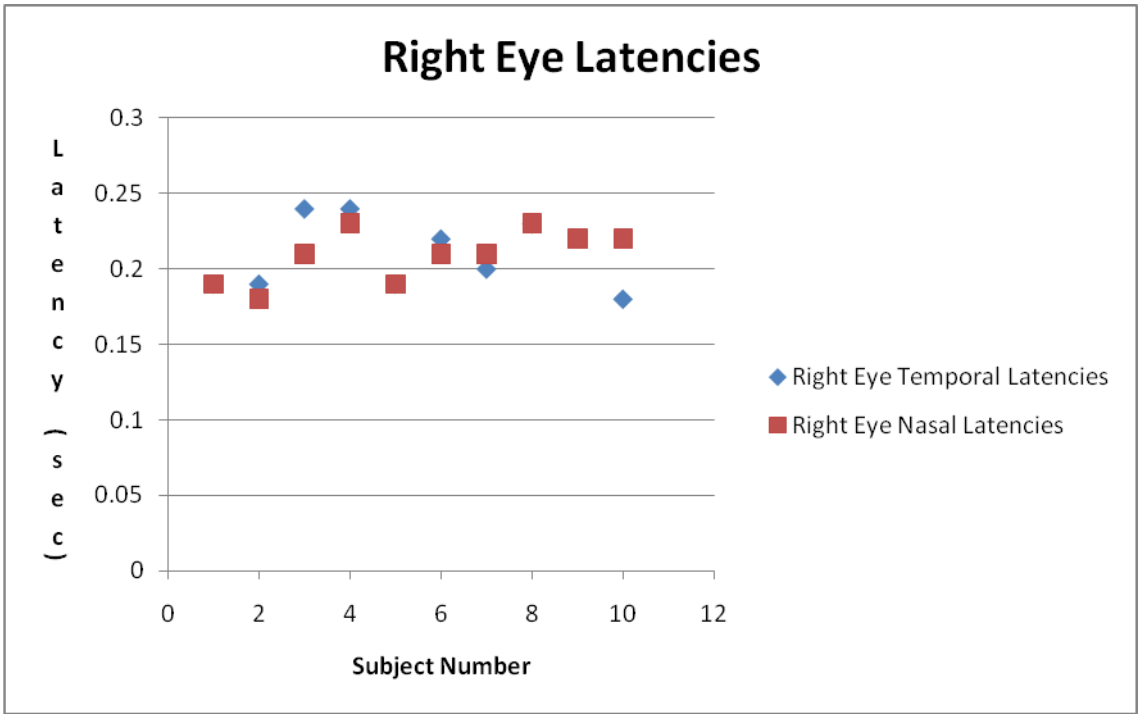

Figure 19b – Nasal and temporal latencies of the right eye of all subjects

Latency								
Groups	Left Eye				Right Eye			
	Temporal		Nasal		Temporal		Nasal	
	Mean	s.d.	Mean	s.d.	Mean	s.d.	Mean	s.d.
Group I	0.19	0.03	0.21	0.05	0.21	0.05	0.19	0.03
Group II	0.20	0.04	0.20	0.06	0.22	0.07	0.21	0.07
Group III	0.22	0.06	0.23	0.06	0.21	0.05	0.22	0.06

Table 4b. Mean saccade latencies for both eyes in both directions of subjects distributed in Groups.

Peak Velocity

The peak velocities of temporally directed saccades were compared with their nasal counterparts using a paired t-test (SPSS). There were no significant differences found ($p>0.05$) when evaluating by subject or population. This implies that there was a nasal/temporal performance symmetry. These results are shown in Table 5a and Figures 20a and 20b. Grouped results also showed no significant different ($p>0.05$) and can be found in Table 5b. Plots of saccadic peak velocities vs time for each subject showed no change in magnitude illustrating that fatigue had no effect on the data.

Peak Velocity (deg/sec)								
	Left Eye				Right Eye			
Subjects	Temporal		Nasal		Temporal		Nasal	
	Mean	s.d.	Mean	s.d.	Mean	s.d.	Mean	s.d.
AC	338.02	27.65	386.05	18.36	380.03	19.31	338.92	66.96
CR	364.33	19.54	356.85	19.36	374.83	19.15	350.56	20.04
KW	369.60	38.66	379.03	26.74	397.85	21.03	372.16	22.21
ML	434.71	46.73	415.24	34.08	411.04	21.82	413.58	27.72
KGB	455.97	30.44	463.28	27.97	457.84	31.18	474.32	34.86
RP	524.28	41.56	411.49	22.21	551.80	35.44	444.50	22.21
EC	361.19	24.86	337.86	19.69	341.49	17.30	366.65	18.99
LW	387.76	36.11	376.49	48.95	382.30	31.35	369.00	40.10
AD	442.39	33.27	436.86	38.89	436.57	37.13	440.36	31.74
KB	365.21	18.63	365.94	24.04	374.37	23.74	365.02	30.48
Mean	404.35		392.91		410.81		393.51	

Table 5a. Saccadic peak velocities for all ten subjects.

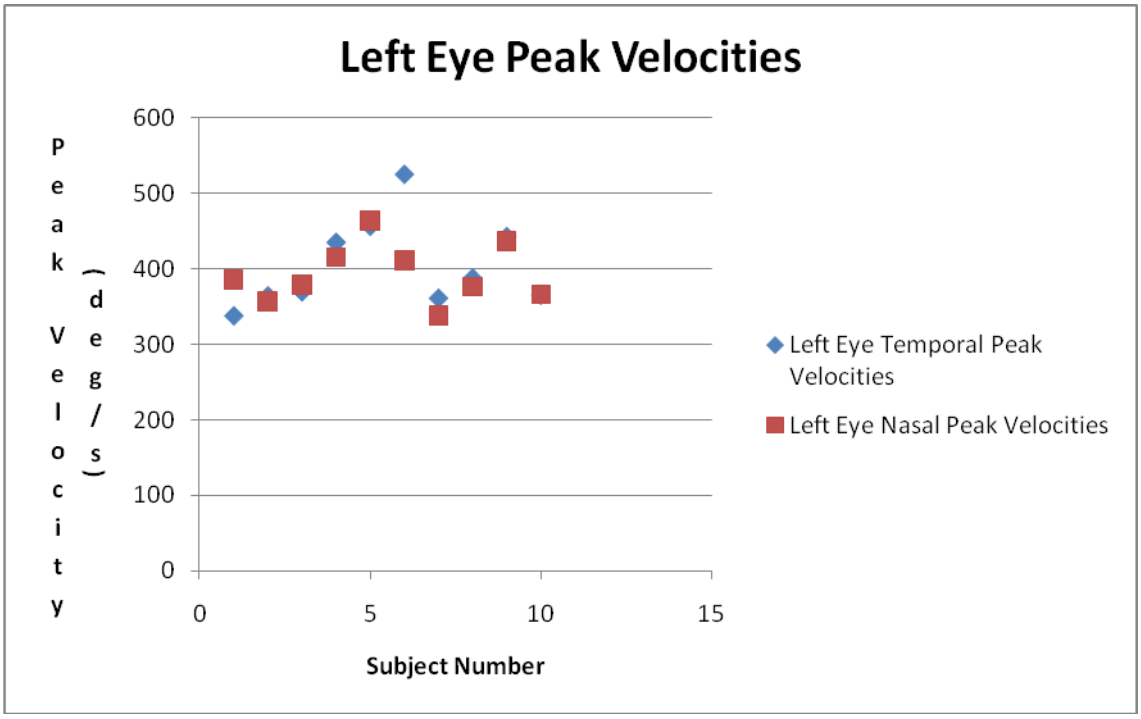

Figure 20a – Nasal and temporal peak velocities of the left eye of all subjects

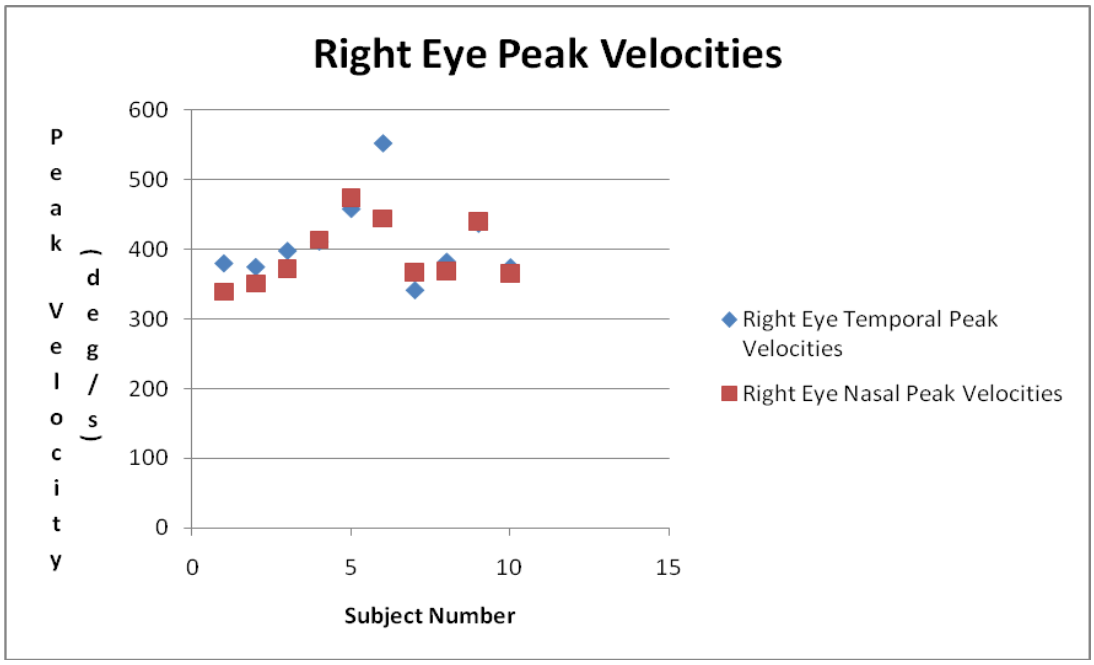

Figure 20b – Nasal and temporal peak velocities of the right eye of all subjects

Peak Velocity								
	Left Eye				Right Eye			
Groups	Temporal		Nasal		Temporal		Nasal	
	Mean	s.d.	Mean	s.d.	Mean	s.d.	Mean	s.d.
Group I	357.32	32.53	373.98	25.12	384.24	22.06	353.88	44.90
Group II	471.65	54.22	430.00	37.32	473.56	62.84	444.13	39.82
Group III	398.45	43.84	393.10	49.58	397.75	45.24	391.46	43.13

Table 5b. Mean peak velocities for both eyes in both directions of subjects distributed in groups.

Experiment 3: Vestibulo-ocular Reflex (VOR)

VOR gains were calculated for each subject as they rotated their heads around a superior/inferior axis (yaw) while visually fixating on a stationary target LED. These results have been separated into left and right rotations in addition to being presented as an overall composite (Table 5a & b). The expected value of VOR gain is -1 when performed under these conditions. All subjects demonstrated values close to this. A paired t-test (SPSS) was performed comparing the VOR gains for leftward and rightward head velocities of individual subjects. There were no significant differences ($p>0.05$) among the quantities. When the gains were analyzed by groups (ANOVA), the leftward VOR gains showed no significant difference ($p>0.05$) but the rightward gains showed a significant difference ($p<0.05$). The overall gain between the groups also showed no significant difference ($p>0.05$). The individual and grouped VOR gains are illustrated in tables below (6a and 6b).

VOR Gain						
	Leftwards		Rightwards		Overall	
Subjects	Mean	s.d.	Mean	s.d.	Mean	s.d.
AC	-1.04	0.08	-1.09	0.08	-1.06	0.06
CR	-0.82	0.16	-0.89	0.03	-0.85	0.09
KW	-0.82	0.03	-0.79	0.05	-0.80	0.04
ML	-0.95	0.16	-1.02	0.07	-0.94	0.06
KGB	-0.61	0.78	-1.25	0.87	-0.93	0.22
RP	-0.66	0.40	-0.80	0.10	-0.73	0.25
EC	-0.84	0.32	-0.76	0.22	-0.80	0.02
LW	-0.96	0.14	-0.85	0.17	-0.91	0.15
AD	-1.00	0.11	-1.02	0.10	-1.01	0.10
KB	-0.88	0.14	-0.96	0.08	-0.92	0.11

Table 6a – VOR gains by subject. The overall mean VOR gains were calculated to be -0.84±0.33 and -0.74±0.54 for leftwards and rightwards direction respectively.

VOR Gain						
	Leftwards		Rightwards		Overall	
Subjects	Mean	s.d.	Mean	s.d.	Mean	s.d.
Group I	-0.89	0.09	-0.92	0.05	-0.91	0.06
Group II	-0.74	0.45	-1.02	0.35	-0.87	0.18
Group III	-0.92	0.18	-0.90	0.14	-0.91	0.09

Table 6b – VOR gains by groups.

Experiment 4: Vestibulo-ocular Reflex Suppression/Cancellation (VORc)

VORc gains were calculated for each subject as they rotated their heads around a superior/inferior axis (yaw) while visually fixating on a laser target that was linked to their head movement. These results have been separated into left and right rotations in addition to being presented as an overall composite (Table 6a & b). The expected value of VORc gain is zero when performed under these conditions since the eye-in-head

velocity would be near zero if the subject were able to completely fixate on a target linked to head movement. A value of zero would represent complete suppression of the VOR system. All subjects demonstrated values close to this. A paired t-test (SPSS) was performed comparing the VORc gains for leftward and rightward head velocities. There were no significant differences ($p>0.05$) among the quantities. When the gains were analyzed by groups (ANOVA), the leftward VORc gains showed a significant difference ($p<0.05$) but the rightward gains showed no significant difference ($p>0.05$). The overall gain between the groups also showed a significant difference ($p<0.05$). The individual and grouped VOR gains are illustrated in tables below (7a and 7b).

VORc Gain						
	Leftwards		Rightwards		Overall	
Subjects	Mean	s.d.	Mean	s.d.	Mean	s.d.
AC	0.02	0.02	0.02	0.05	0.00	0.02
CR	-0.22	0.11	-0.19	0.09	-0.20	0.06
KW	-0.24	0.23	-0.18	0.16	-0.21	0.19
ML	-0.23	0.14	-0.24	0.09	-0.24	0.10
KGB	-0.25	0.20	-0.22	0.20	-0.24	0.20
RP	-0.30	0.15	-0.26	0.17	-0.28	0.15
EC	-0.40	0.16	-0.35	0.17	-0.37	0.16
LW	-0.39	0.25	-0.35	0.24	-0.37	0.22
AD	-0.41	0.35	-0.37	0.33	-0.39	0.34
KB	-0.65	0.40	-0.39	0.90	-0.52	0.63

Table 7a – VORc gains by subject. The overall mean VORc gains were calculated to be -0.21±0.16 and -0.18±0.15 for leftwards and rightwards directions respectively.

	VORc Gain					
	Leftwards		Rightwards		Overall	
Subjects	Mean	s.d.	Mean	s.d.	Mean	s.d.
Group I	-0.15	0.12	-0.12	0.10	-0.14	0.09
Group II	-0.26	0.16	-0.24	0.15	-0.25	0.15
Group III	-0.46	0.29	-0.36	0.41	-0.41	0.34

Table 7b – VORc gains of subjects in both directions distributed in groups.

Although there was no consistent statistically significant difference between the groups, it should be noted that there is a trend in these data. Positive VORc values represent the ability to suppress and counter the VOR. This only occurs in subject AC (#1) who is one of the highest ranked gymnasts in this subject pool (level 10). The subject order is a function of this VORc ranked hierarchy (ordered best to worst from top to bottom). The poorest performing subject (KB, #10) was one of the lowest ranked gymnasts (recreational) in the subject pool. VORc gains are illustrates in Figure 21.

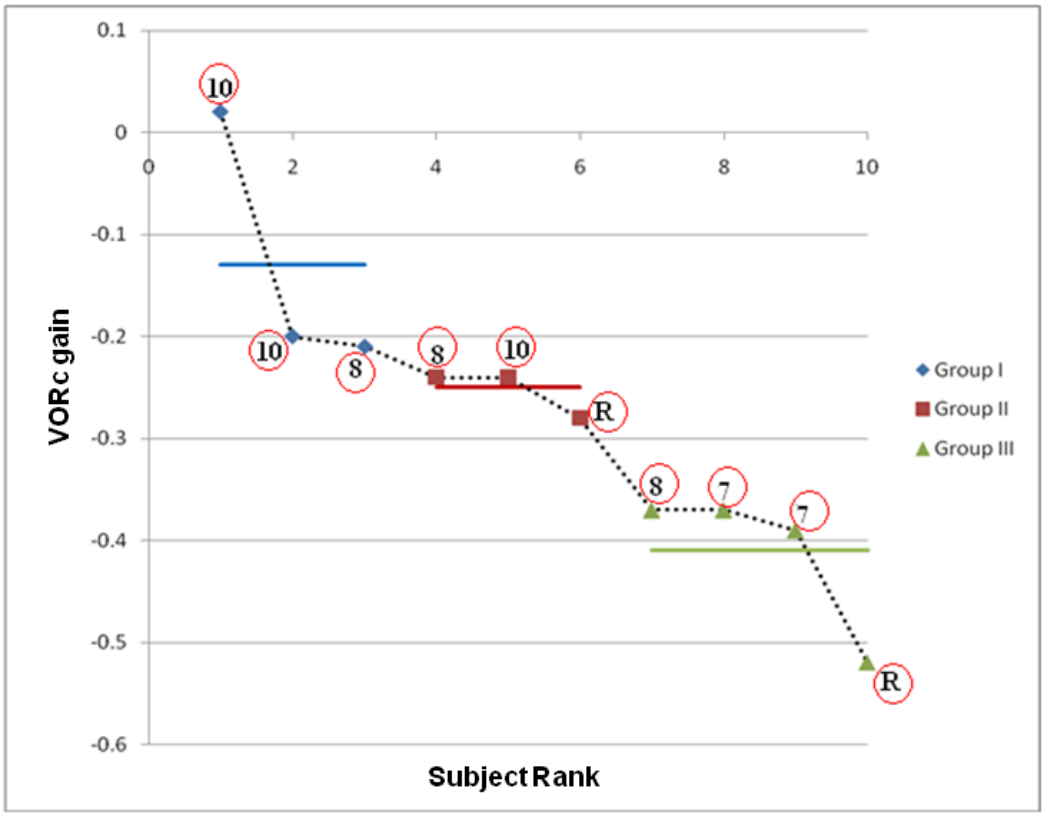

Figure 21 – VORc gains plotted by subject. Subject #1 (AC) had the highest level of VOR suppression. Subject #10 (KB) had the poorest performance. The horizontal lines represent the mean VORc for that group.

When VORc is plotted against the subjects gymnastics competitive level, there is a mild positive correlation (R^2=0.51). Although this is not a strong correlation, there does appear to be a trend in these results illustrating that the higher level gymnasts have increased ability to suppress the VOR (Figure 22). This figure also shows a trend for the higher level gymnasts to have VOR gains closer to the expected value of -1. In short, their performance in both the VOR and VORc experiments is better than the rest of the group.

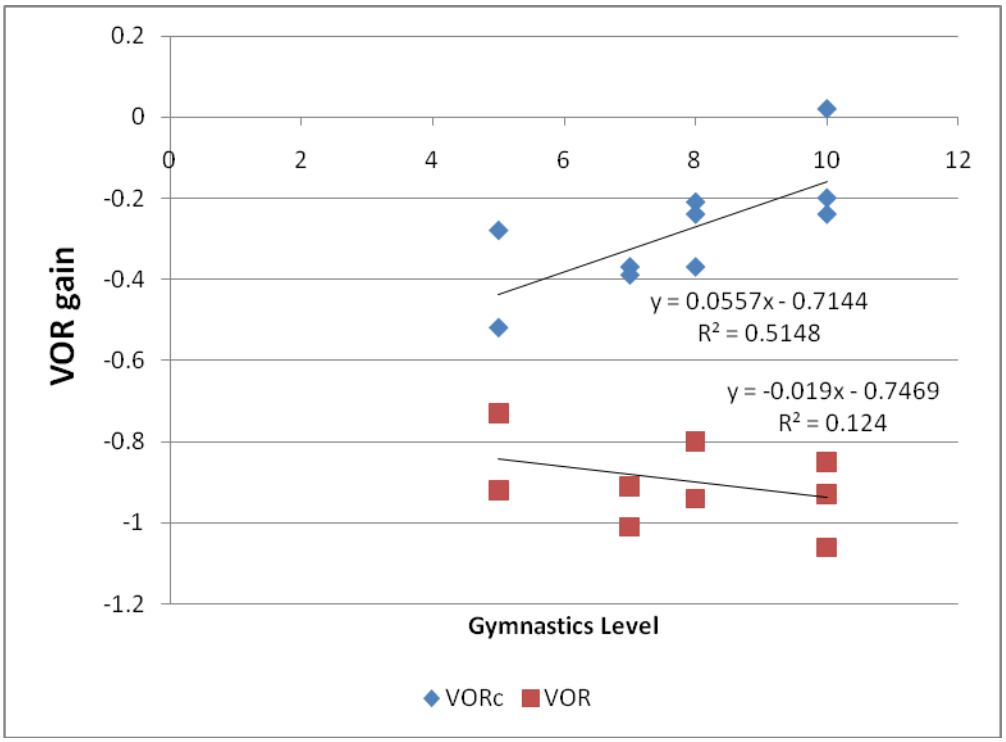

Figure 22 – VOR gains plotted by gymnastics level. The red data represents VOR gains with expected values of -1. The blue data represents VORc gains with the expected value of 0 for complete suppression of VOR. Recall that more positive gain values represent increased ability to suppress the VOR.

VOR / VORc Response Magnitudes

Recall that the VOR and VORc data were collected while pacing head rotation to a metronome. Head motion was cyclical and can be described as having a left-to-right *excursion* followed by a right-to-left *excursion* that ended at the start position (Figure 23). Based on the presented metronome pacing frequencies and a head range of motion of ±15°, head angular velocities were expected to range from 35 to 100°/s (if constant velocity movement is assumed). Head angular velocities actually ranged from 10 to 150°/s since constant velocity is not possible and time must be spent accelerating and

decelerating during each *excursion*. Clinical studies have shown that eye velocities under vestibular control can reach a maximum velocity of 315°/s [33]. The current study involves angular head velocities that are well below this limit and are therefore within the bandwidth of VOR response. The majority of the data for any given subject were in the mid-range of velocities since subjects tended to decrease their head rotational range of motion as the pacing frequencies increased.

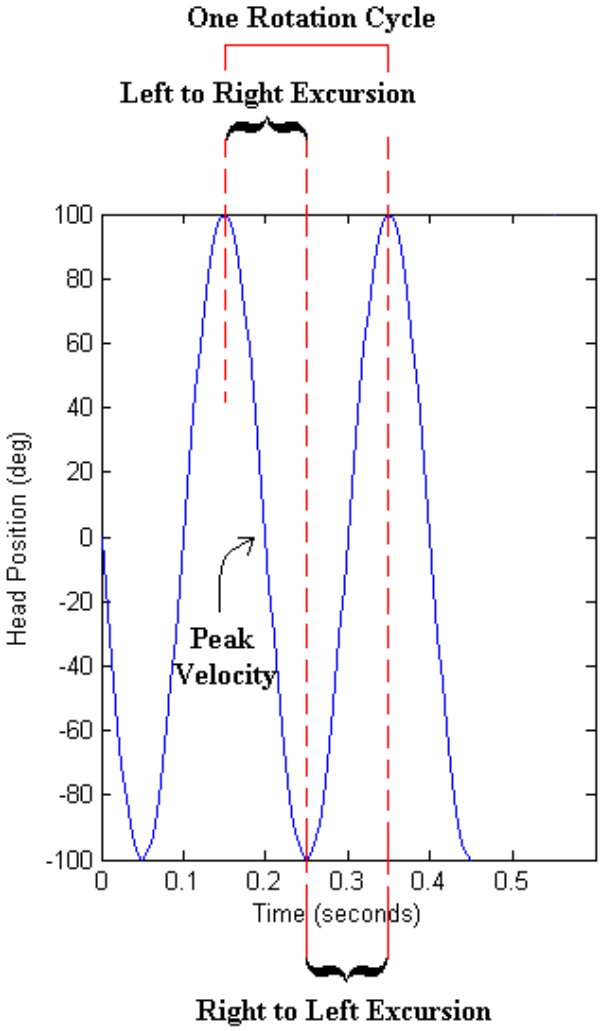

Figure 23 – Plot representation of the two excursions during one cycle of head motion (yaw). Note that the peak velocity occurs midway through an excursion and that two excursions make up one rotation cycle.

To analyze this mid-range velocity, sections of temporal data with peak angular velocities between 65-75°/s were averaged. This velocity range was selected because it can be easily compared to previous research [34]. The VOR and VORc responses for head and eye velocities within the 65-75°/s head velocity range were plotted. A typical subject is illustrated in Figures 24a, b, and c. These plots contain leftward, rightward, and overall excursion data from start to peak velocity. Note that the head and eye velocities have different signs consistent with a typical VOR response (the head and eyes move in opposite directions). Note also that the VORc eye velocity amplitude is less than the VOR. This was to be expected since VORc results in less movement of the eye relative to the head. Values near zero would be expected with VOR suppression. The reported "n" values denote the number of trials that met the inclusion criteria.

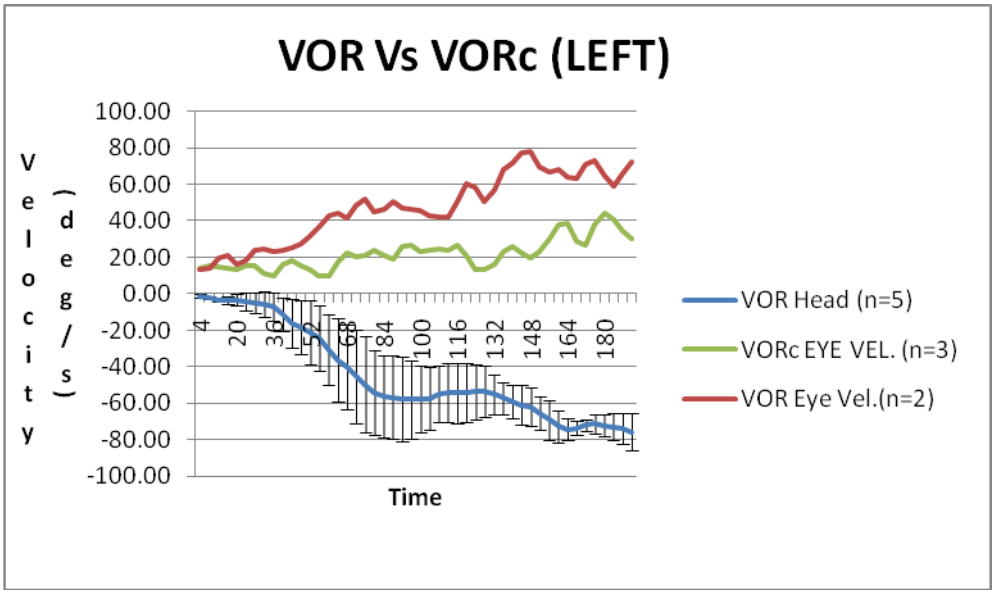

Figure 24a – Typical composite data for one subject. VOR vs VORc response while the subjects head is moving towards the left with a peak angular velocity of between 65-75°/s.

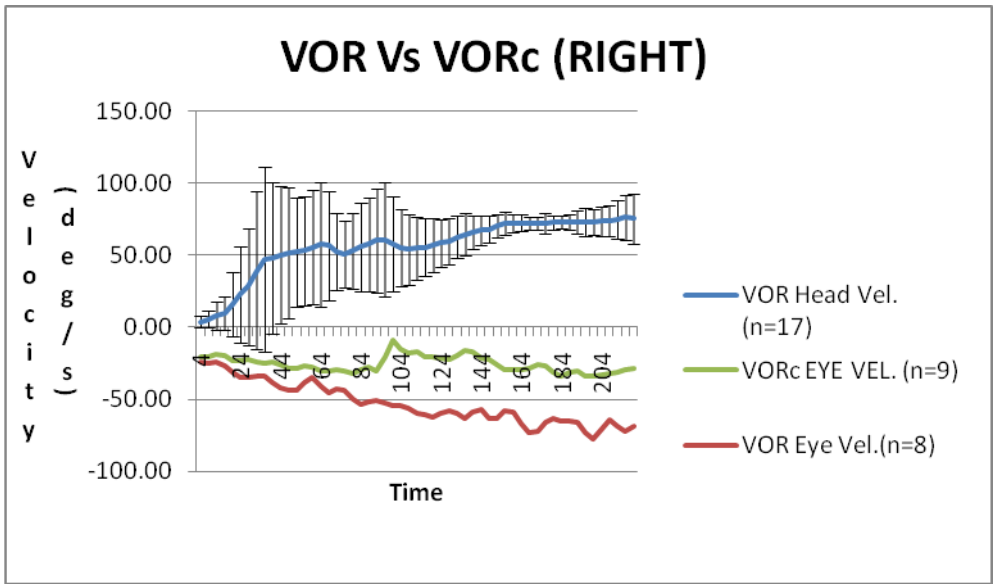

Figure 24b – Typical composite data for one subject. VOR vs VORc response while the subjects head is moving towards the right with a peak angular velocity of between 65-75°/s.

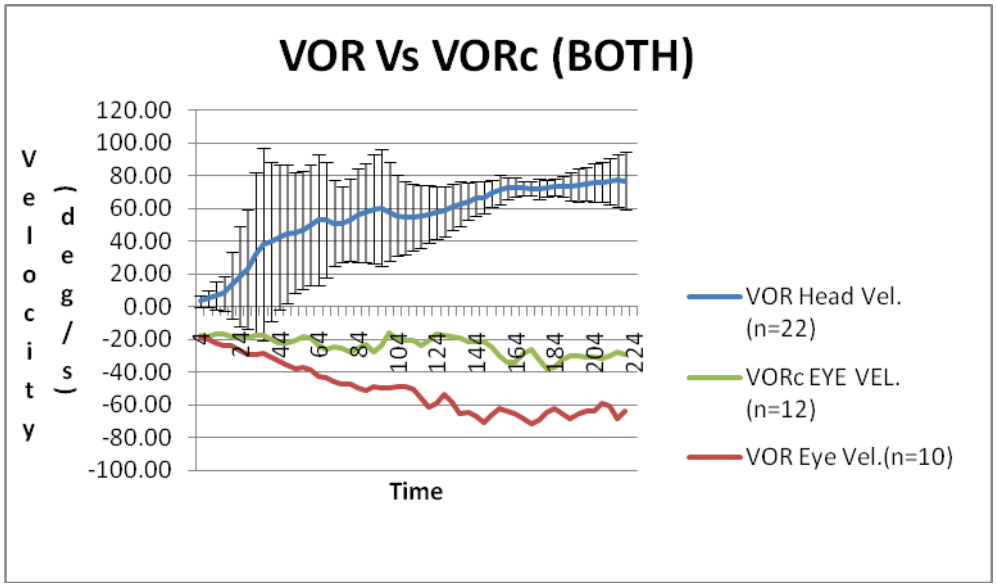

Figure 24c – Typical composite data for one subject. VOR vs VORc response while the subjects head is moving left and right with a peak angular velocity of between 65-75°/s. In this case the right-to-left data was inverted so that it could be mathematically averaged with the left-to-right data.

VOR / VORc Response Timing

In order to evaluate the VOR / VORc response timing, an *index* was developed that was sensitive to temporal changes in the data. This index differenced integrated velocity data by *excursion* (as described in the Methods section). Using this technique, it was expected that the VOR data would have an *index* close to zero since the head and eye velocities would be in opposite directions with near equal magnitudes. The VORc data was expected to follow the VOR data trend initially and then deviate from zero when suppression began. An example of VOR and VORc indices plotted against time as shown in Figures 25a, b, and c. From this plot, the VORc latency was determined by visual estimation. This latency was defined as the point at which the index began to deviate from zero in a linear fashion. In Fig. 25a, we can see the subject took around 68ms after head movement initiation (time = 0ms) to cancel VOR.

Figure 26 illustrates the average latencies for all ten subjects. Note that subjects 2 and 10 do not have average latency values on this plot because they did not have VOR and/or VORc within the 65-75°/s head velocity range. Table 8 numerically describes these results.

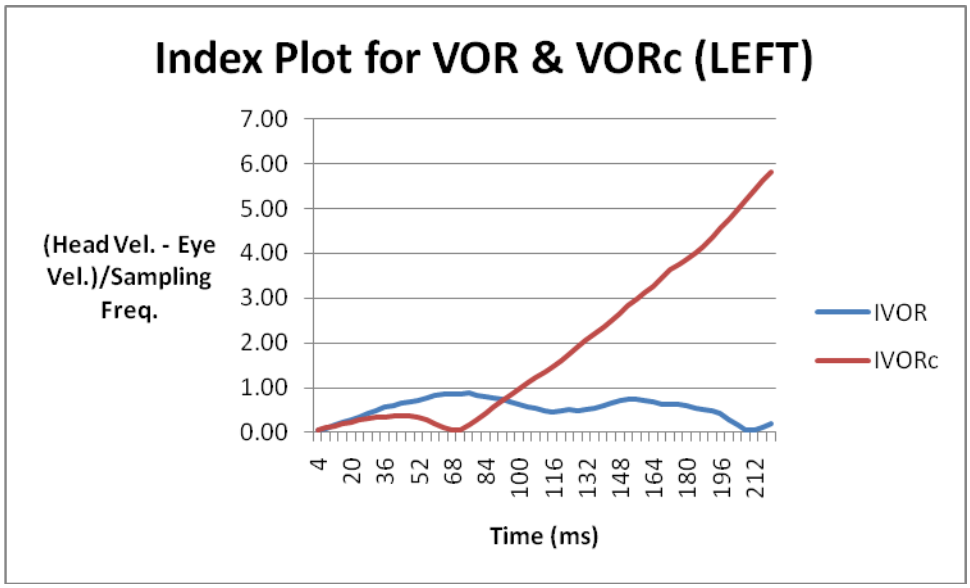

Figure 25a – Temporal index plot for VOR vs VORc plot while the head velocity is between 65-75 degrees/sec towards the left

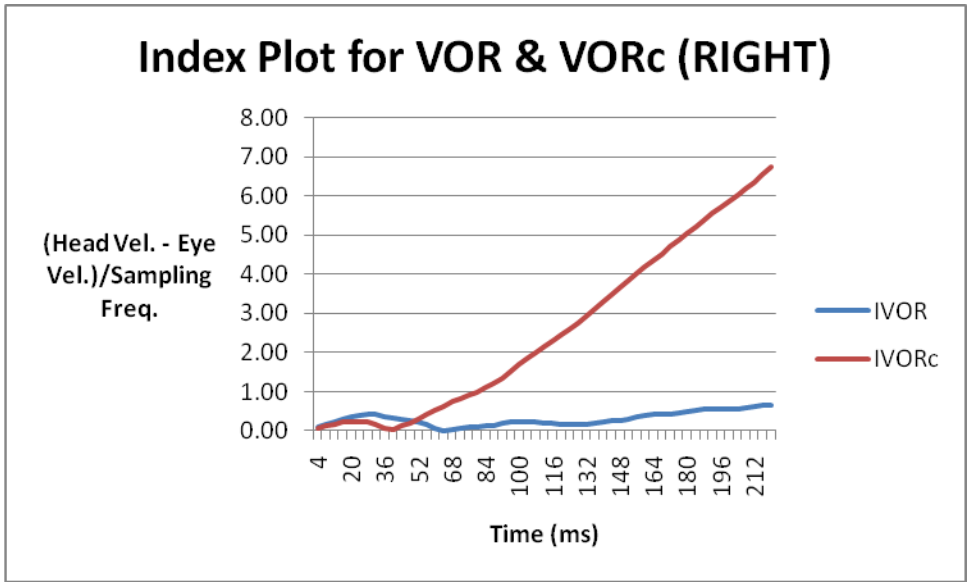

Figure 25b – Temporal index plot for VOR vs VORc plot while the head velocity is between 65-75 degrees/sec towards the right

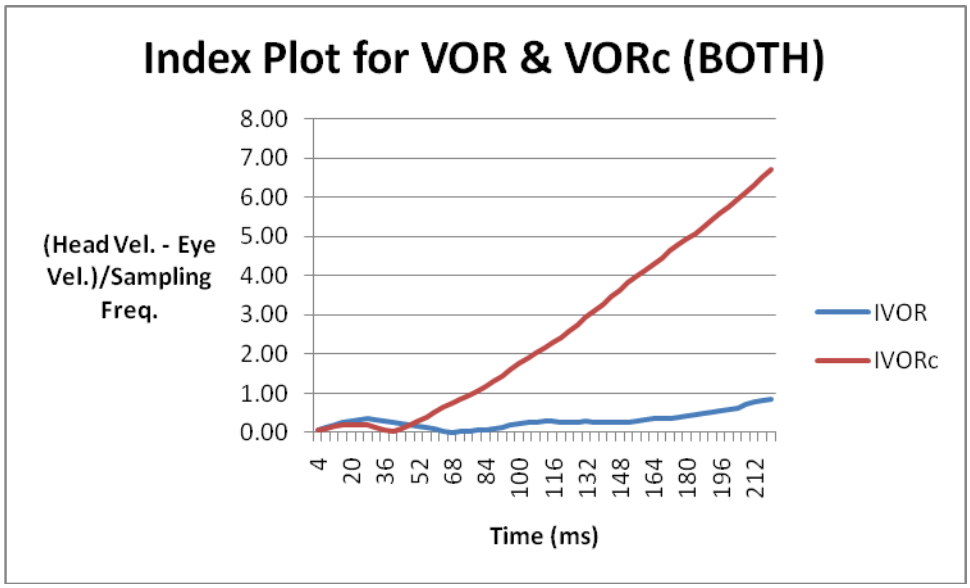

Figure 25c – Net temporal index plot for VOR vs VORc plot while the head velocity is between 65-75 degrees/sec.

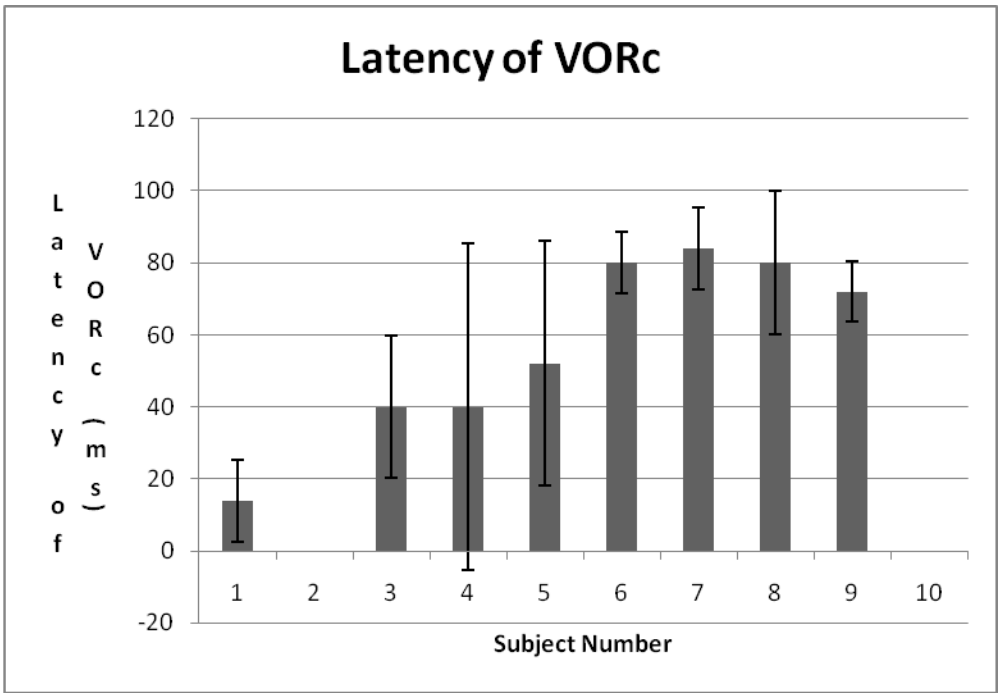

Figure 26 – Net latencies of the VORc for the subjects when their head velocities ranged from 65-75°/s.

Subject Rank	Subject	Net Latency	Std. Dev.
1	AC	14	11.31
2	CR	-	-
3	KW	40	19.8
4	KGB	40	45.25
5	ML	52	33.94
6	RP	80	8.49
7	EC	84	11.31
8	LW	80	19.8
9	AD	72	8.49
10	KB	-	-

Table 8 – Overall latency of VOR cancellation and standard deviation.

Discussion

Volitional Saccades

Amplitude

Studies regarding quantitative measurement of saccade amplitude, duration, and velocity on normal human subjects have shown a linear relationship between duration and amplitude [13, 3536]. The slope of this relationship lies between 1.5 to 3.0 ms/deg and is presented in a reproduction of these data in Figure 27. Pooled data from experiment #2 is also presented on this same plot. Recall that these data were collected as the subjects transitioned between two LED targets positioned approximately 20° apart along a horizontal path. The angular distance is described as "approximate" since the head was free to move and this would affect the angular displacement between the targets. Saccadic amplitude would also be impacted by head rotation during the experiment since there would be some tendency to move the head in the direction of the target being acquired. This would result in saccades smaller than the expected 20°. Although subjects were instructed to minimize head movement, these pooled data have eye amplitudes of approximately 15°. The duration of these saccades is less than that found in a normal population. This may be an artifact of data collect (since no bite bar was used), but it is worth considering that this population of gymnasts may simply differ from normal.

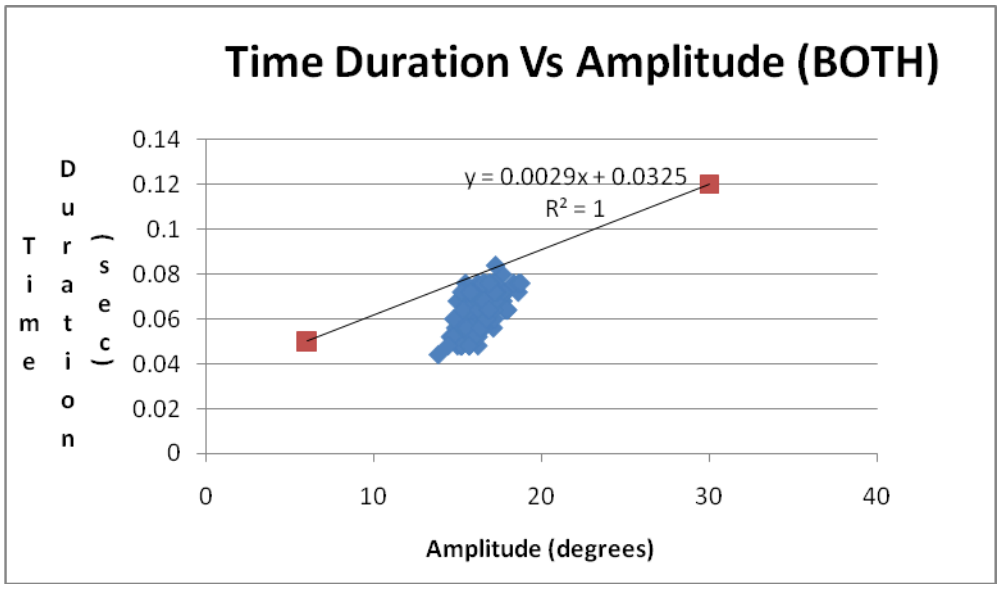

Figure 27 – Saccadic duration vs. amplitude plot for pooled data (n=75). Data from [36] is plotted in red. The blue data represents pooled data from gymnasts in this study.

Peak Velocity

Peak velocity was plotted against amplitude by Baloh et. al. as well. They fit their response data with an exponential equation : Time = $K(1-e^{-(Amplitude/L)})$ where K and L were constants with ranges 551 ± 65° and 14 ± 1.7° respectively. Figure 28 describes these boundaries graphically and includes pooled data from the gymnasts. The upper boundary represents the exponential fit using the maximum values of K and L. The lower boundary represents the minimum values of the same. The gymnasts' responses fall within the predicted range.

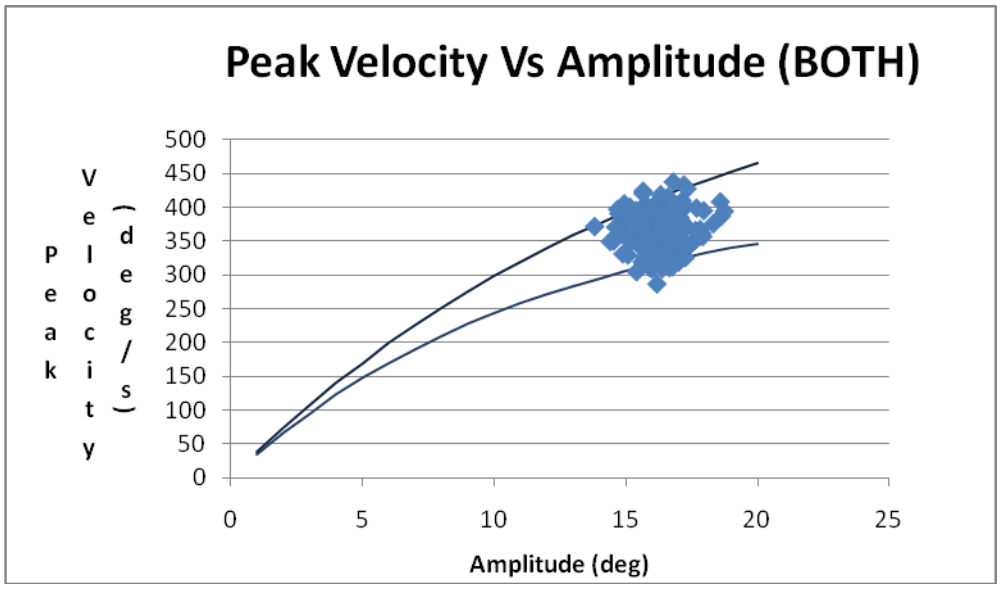

Figure 28 – Peak velocity vs. amplitude plot comparison for pooled data. The lines represent the upper and lower boundary for data obtained by Baloh et al. The blue dots represent data from the gymasnts.

Asymmetries

Studies have shown that nasal and temporal saccades of the same eye can have latency and velocity asymmetries [28, 37]. In these studies, subjects performed saccadic eye movements from a central target to eccentric targets when cued via the lighting of that target. Honda's work included two types of saccadic experiments: *reflexive saccade* and *voluntary saccade*. *Reflexive saccades* were performed by asking subjects to initially fixate on a central target. Illumination of a left or right eccentric target would cue the reflexive response to re-fixate. The sequence of the target illumination was randomized. *Voluntary saccades* were designed to require more cognitive input by forcing the subject to make a decision before re-fixating. In this experiment, the center target was replaced with a vertical line. The target was further enhanced with a small horizontal line that would momentarily appear to the left or right of the center line. This was the cue for the

subject to perform a saccade to fixate on the eccentric target to that side. Both eccentric targets were displayed throughout the experiment, so only the momentary appearance of the horizontal line provided any visual stimulus.

In Experiment 2: Volitional Saccades, there were two fixed targets illuminated in a randomized-alternating pattern cueing the fixations. Statistical analysis of the gymnasts' nasal and temporal saccadic latencies revealed no significant difference between them ($p>0.05$). One explanation for this finding may be protocol difference between the experiments. Honda's study utilized a central target as a starting point for each left or right saccade. The gymnasts were not provided with a central target and moved between the two eccentric targets. Honda's work also shows that saccadic responses varied and are unique to the individual. Although there exists some difference between our conclusions and that found by Honda, the range of latencies found in both studies (i.e. 150 – 250 ms) match, suggesting that the gymnasts belong to a normal population.

Now consider the angular peak velocity during horizontal eye movements. When a subject with normal eye movement control is made to perform a horizontally directed saccade to the right, the right eye will move in a temporal direction(abduction) and the left will move in a nasal direction (adduction). The peak velocity of each eye may be different [38, 39]. According to these studies, the abducting eye accelerates faster than the adducting eye and the peak velocity of the abducting eye is usually equal or higher than the adducting eye. In the gymnasts, seven out of ten subjects revealed these traits in the

data, but the standard deviations of both studies were too large to have any statistical significance.

Collewijn et all also found asymmetric time lags between the target activation and the initiation of the left and right eye[38]. The latency for the abducting eye was found to be shorter than the adducting eye. Flipse et al reported small time periods (0 to 6ms) with a system sampling rate of 1000 Hz[40]. The eye tracker used in the gymnasts' experiment had a sampling frequency of only 250 Hz. This did not provide enough temporal resolution to compare findings. Figures 29a and b illustrate these temporal delays and suggest some difference. Figure 29a illustrates that the abducting eye (left) and the adducting eye (right) respond symmetrically and fig. 29b illustrates that the abducting eye (left) leads the adducting eye (right) by around 4 ms. The slope of the eye movements in both graphs appear to be the same which suggests that peak velocities and acceleration of both eyes were equally matched. Thus the results arrived at through this experiment agrees with previous studies. Hence, the subjects used for this experiment could be considered as a part of the normal population.

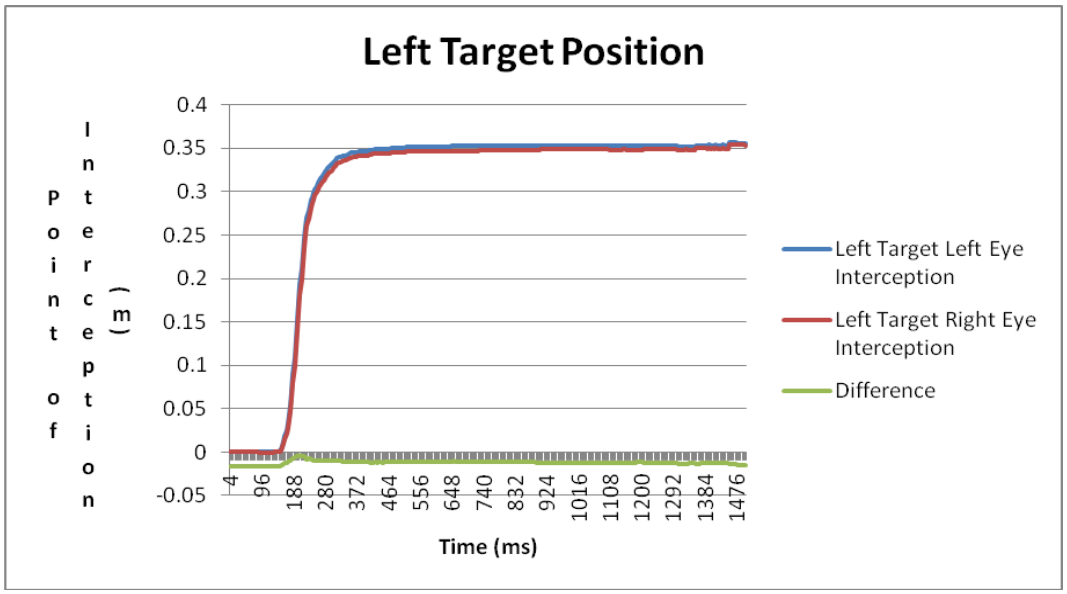

Figure 29a – Points of interception of gazes from both eyes and the left target. The position error between the two eyes is near zero and there is no difference at the initialtion of eye position change.

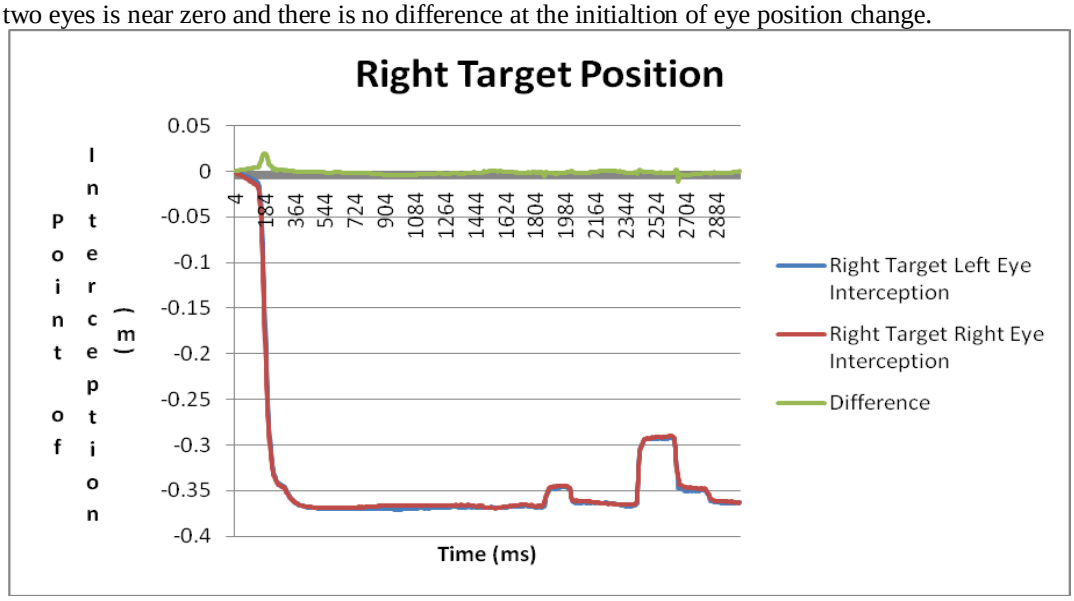

Figure 29b – Points of interception of gazes from both eyes and the right target. Although the position error between the two eyes is near zero, there is a slight difference at the initialtion of eye position change. This difference suggests a nasal lag in the left eye at 4 ms.

VOR and VORc

The VOR and VORc data revealed some interesting trends. The group analysis demonstrated that there was a significant difference across the groups when the subjects were grouped according to VORc performance. Figures 21 and 22 illustrate a difference as a function of gymnastics competitive level. From these data it appears that higher level gymnasts have VOR gains closer to -1 and VORc gains closer to 0. This suggests some VOR changes associated with elite performance. Since the higher level gymnasts were also the older gymnasts, it might be concluded that the VOR and VORc trends are due to the visual system maturity. This is probably not the case. There are a number of studies in literature characterizing the development of the VOR in children and adolescents. According to one, children aged 8 or above have a VOR similar to adults [41]. Since all of the gymnasts studied in this experiment were older than 14, it is expected that they had fully developed reflexes. In addition, the VOR gains measured in earlier studies vary from 0.6 – 1.11 depending on the head velocities used [41]. The gymnasts had a mean VOR gain of 0.79±0.45, fitting right into these previous works.

In short, the higher level gymnasts had better VOR gains (closer to -1) and better VORc gains (closer to 0). The gymnasts with the VORc gains closer to 0 were also the gymnasts capable of performing multiple twisting skills. Not only were the VORc gains of the higher level gymnasts closer to 0, but the suppression of the VOR system occurred more quickly. This would give these gymnasts a distinct advantage when performing aerial twisting skills that required visualizing the landing.

The VOR L/R gains (Table 7a) correlated with gymnasts twist direction (Table 2) in all but two competitive gymnasts. This result does not provide evidence that measured differences in VOR L/R gains can predict the expected direction of twist the gymnast will choose, but does provoke an interesting discussion. The twist direction of developing gymnasts has been a topic of controversy for a long time in the gymnastics community. Some feel that it should follow the direction of the cartwheel. Others think that it should be opposite that or should be determined by observing the child during jump twists. In either case, once it is selected, it becomes the standard for that gymnast as twisting techniques are taught.

In six of the competitive gymnasts tested in this study, their preferred direction of twist matched a VOR asymmetry that was biased in their preferred direction of twist. These six gymnasts routinely perform and practice multiple twisting skills. The two competitive gymnasts whose VOR asymmetry is opposite their preferred direction of twist also practice multiple twisting skills, but have not yet been able to master even single twisting flips. It is possible that they have been taught to twist in the wrong direction, or at least a direction that does not take advantage of their VOR asymmetry. Gymnasts who have protracted trouble twisting are often retrained in the opposite direction. This may be of benefit to these two. In fact, one of the two has a highly developed VORc, suggesting that if taught to twist in the opposite direction, may be very successful at multiple twisting skills.

The purpose of this research is to investigate differences in the VOR performance of young female gymnasts who routinely perform flipping twisting activities. The results

do show trended differences in this group, but care should be taken in interpreting these results. Limitations in this study include a small sample size and lack of normal subjects for control.

Literature Cited

Literature Cited

1. Clement G, Deguine O, Bourg M, Pavy-LeTraon A. Effects of vestibular training on motion sickness, nystagmus, and subjective vertical. *Journal of Vestibular Research*. 2007;17:227.

2. Sturnieks DL, St. George R, Lord SR. Balance disorders in elderly. *Clinical Neurophysiology*. 2008;38:467-478.

3. Roger MW, Hedman LD, Johnson ME, Martinez KM, Mille ML. Triggering of protective stepping for the control of human balance: Age and contextual dependence. *Cognitive brain research*. 2003;16:192.

4. Horak FB. Postural compensation for vestibular loss. *Annals of the New York Academy of Sciences*. 2009;1164.

5. Friedrich M, Grein H, Wicher C, et al. Influence of pathologic & simulated visual dysfunctions on the postural system. *Experimental Brain Research*. 2008;186:305.

6. Diehl MD, Pidcoe PE. The influence of gaze stabilization and fixation on stepping in younger and older adults. *Journal of Geriatric Physical Therapy*. 2010;33:19.

7. Keshner EA, Cohen H. Current concepts of the vestibular system reviewed: 1. the role of the vestibulospinal system in postural control. *The American Journal of Occupational Therapy*. 1989;43:320.

8. Flanders M, Daghestani L, Berthoz A. Reaching beyond reach. *Experimental Brain Research*. 1999;126:19.

9. Luis M, Tremblay L. Visual feedback use during a back tuck somersault: Evidence for optimal visual feedback utilization. *Motor contr*. 2008;12:210.

10. Campbell FW, Green DG. Optical and retinal factors affecting visual resolution. *The Journal of Physiology*. 1965;181:576.

11. Meyer CH, Lasker G, Robinson DA. The upper limit of human smooth pursuit velocity. *Vision Research*. 1985;25:561.

12. Collewijn H, Tamminga EP. Human smooth & saccadic eye movements during voluntary pursuit of different target motions on different backgrounds. *J Physiol*. 1984;351:217 - 250.

13. Baloh RW, Sills AW, Kumley WE, Honrubia V. Quantitative measurement of saccade amplitude, duration and velocity. . 1975;25:1065.

14. Black JL, Collins DWK, De Roach JN, Zubrick SR. Smooth pursuit eye movements in normal & dyslexic children. *Perceptual & Motor Skills*. 1984;59:91 - 100.

15. Schütz AC, Braun DI, Gegenfurtner KR. Object recognition during foveating eye movements. *Vision Research*. 2009;49:2241.

16. Baloh RW, Lyerly K, Yee RD, Honrubia V. Voluntary control of the human vestibulo-ocular reflex. *Acta oto-laryngologica*. 1984;97:1.

17. Schubert MC, Minor LB. Vestibulo-ocular physiology underlying vestibular hypofunction. *Phys Ther*. 2004;84:373-385.

18. Viirre ES, Demer JL. The vestibulo-ocular reflex during horizontal axis eccentric rotation and near target fixation. *Ann N Y Acad Sci*. 1996;781:706-708.

19. Crane BT, Demer JL. Human horizontal vestibulo-ocular reflex initiation: Effects of acceleration, target distance and unilateral deafferentation. *Journal of neurophysiology*. 1998;80:1151.

20. Fetter M. Vestibulo-ocular reflex. *Developments in Opthalmology*. 2007;40:35.

21. Barton JJ. Vertigo and Vestibular Function.

22. Peterka RJ. Sensorimotor integration in human postural control
. *Journal of Neurophysiology*. 2002;88:1097.

23. Dieterich M. Central vestibular disorders. *Journal of Neurology*. 2007;254:559-68.

24. Judge JO, King MB, Whipple R, Clive J, Wolfson LI. Dynamic balance in older persons: Effects of reduced visual and proprioceptive input. *Journal of Gerontology: Medical Sciences*. 1995;50A:M263-M270.

25. Lee DN, Aronson E. Visual proprioceptive control in standing human infants. *Perception & Psychophysics*. 1974;15:529-532.

26. Perrin P, Schneider D, Deviterne D, Perrot C, Constantinescu L. Training improves the adaptation to changing visual conditions in maintaining human posture control in a test of sinusoidal oscillation of the support. *Neuroscience Letters*. 1998;245:155-158.

27. Pidcoe PE, Diehl MD. Development and validation of an eye tracking system for use in balance recovery
research.

28. Honda H. Idiosyncratic left-right asymmetries of saccadic latencies: Examination in a gap paradigm. *Vision Research*. 2002;42:1437.

29. Träisk F, Bolzani R, Ygge J. Intra-individual variability of saccadic velocity measured with the infrared reflection and magnetic scleral search coil methods. . 2006;14:137-46.

30. Irving EL, Steinbach MJ, Lillakas L, Babu RJ, Hutchings N. Horizontal saccades dynamics across the human life.
. . 2006;47:2478.

31. Penfold PL, Provis JM. *Macular Degeneration.* ; 2005.

32. Paige GD, Telford L, Seidman SH, Barnes GR. Human vestibuloocular reflex and its interactions with vision and fixation distance during linear and angular head movement. *Journal of Neurophysiology*. 1998;80:2391.

33. Reker U. Extent of reaction capacity of the vestibulo-ocular reflex. *Archives of oto-rhino-laryngology*. 1980;228:35.

34. Crane BT, Demer JL. Latency of voluntary cancellation of the human vestibulo-ocular reflex during transient yaw rotation. *Experimental Brain Research*. 1999;127:67.

35. Freedman EG. Coordination of the eyes and head during visual orienting. *Experimental Brain Research*. 2008;190:369.

36. Bahill AT, Brockenbrough A, Troost BT.
Variability and development of a normative data base for saccadic eye movements. *Investigative Ophthalmology and Visual Science*. 1981;21:116.

37. Fricker SJ. Dynamic measurements of horizontal eye motion: Acceleration and velocity matrices. *Investigative ophthalmology*. 1971;10:724.

38. Collewijn H. Interocular timing differences in the horizontal components of human saccades. *Vision Research*. 2001;41:3413.

39. Kloke WB, Jaschinski W. Individual differences in the asymmetry of binocular saccades, analysed with mixed-effects models. *Biological Psychology*. 2006;73:220.

40. Flipse JP, Straathof CS, Van der Steen J, et al. Binocular saccadic eye movements in multiple sclerosis. *Journal of the Neurological Sciences*. 1997;148:53.

41. Salman MS, Lillakas L, Dennis M, Steinbach MJ, Sharpe JA. The vestibulo-ocular reflex during active head motion in children and adolescents. . *Child's nervous system*;23:1269.

APPENDIX A: The Checklist used while performing the Experiments.

Date:_____

Experiment Checklist.
Initials:_____

1) Calib n Valid (Ctrl F8)
2) 9 point
3) Saccade……L L R R
4) 9 point
5) VOR 72 92 112 132 152 176 192
6) 9 point
7) Calib n Valid (Ctrl F8)
8) 9 point
9) VORc 72 92 112 132 152 176 192
10) 9 point

APPENDIX B: MATLAB code for NPV

```
close all
clear

filename=input('Select File to Run: ','s')
Y=load(filename);

LE_X=Y(:,2);    LE_Y=Y(:,3);    LE_Z=Y(:,4);
RE_X=Y(:,5);    RE_Y=Y(:,6);    RE_Z=Y(:,7);

Lg_X=Y(:,8);   Lg_Y=Y(:,9);   Lg_Z=Y(:,10);
Rg_X=Y(:,11);   Rg_Y=Y(:,12);   Rg_Z=Y(:,13);

Lhorz_eye_angle=Y(:,14); Lvert_eye_angle=Y(:,15);


file_len=length(LE_X);
RADIAN=3.14/180;
sample_freq = 250;
sample_rate = 1/sample_freq;

%-----------------------------------------------
%  create arrays
%-----------------------------------------------
pog_X=zeros(file_len,1);
pog_Y=zeros(file_len,1);
pog_Z=zeros(file_len,1);
X_error=zeros(file_len,1);
Y_error=zeros(file_len,1);
Z_error=zeros(file_len,1);

%smooth velocity data
filt_size = 51;
ihalf = cast(fix(filt_size / 2),'int16');   %truncate answer
sLhorz_eye_angle=zeros(file_len,1);

for i=ihalf+1:file_len-ihalf
    sum = 0;
    for j=i-ihalf:i+ihalf
        sum = sum + Lhorz_eye_angle(j);
    end
    sLhorz_eye_angle(i) = sum/filt_size;
end

for i=1:ihalf                                   %fill begining of array
    sLhorz_eye_angle(i) = sLhorz_eye_angle(ihalf+1);
end
```

```
for i=file_len-ihalf+1:file_len             %fill end of array
    sLhorz_eye_angle(i) = sLhorz_eye_angle(file_len-ihalf);
end


figure(1)
hold on
title('Left Horiz eye angle')
plot(Lhorz_eye_angle(:,1), 'b')
xlabel('Frames');
ylabel('Eye Angle(degrees)');
%-------------------------------------------
%   Selecting Data Points from the Graph
%-------------------------------------------
query = input('Use existing data?Y/N =','s')

if query == 'y'
    F = input( 'Filename containing Frames:','s')
    x = load(F);
%     dlmread('y.out',';','C1');
    G = input( 'Filename containing Eye Angles:','s')
    y = load(G);
elseif query == 'n'
    [x,y] = ginput(18);
    for i =1:2:18
        start = x(i);
        stop = x(i+1);
        start = round(start);
        stop = round(stop);

        m = mean(sLhorz_eye_angle(start:stop));
        for j=start:stop
            if abs(m - sLhorz_eye_angle(j)) > 0.5
                start = start+1;
                stop = stop - 1;
            end
        end
    end
    %-------------------------------------------
    %   Exporting Frame and Eye data values
    %-------------------------------------------
    save x.out x -ASCII;
    dlmwrite ('x',x,';');
    save y.out y -ASCII;
    dlmwrite ('y',y,';');
end
%-------------------------------------------------------------------------
-
%      Error Calculation
%-------------------------------------------------------------------------
-
 x = round(x);
XYZ_error=zeros(file_len,1);
```

```
LED_X=zeros(file_len,1);
LED_Y=zeros(file_len,1);
LED_Z=zeros(file_len,1);
flag = 0;
for i=1:2:18
    start = x(i);
    stop = x(i+1);
    flag = flag+1;
    for j=start:stop

        %compute eye point-of-gaze within stationary eye-position
segments
        ml = Lg_X(j)/Lg_Y(j);% Calculating m1 in excel file
        mr = Rg_X(j)/Rg_Y(j);% Calculating m2 in excel file
        bl = abs(LE_Y(j))* ml + LE_X(j);% Calculating b1 in excel file
        br = abs(RE_Y(j))* mr + RE_X(j);% Calculating b2 in excel file
        pog_Y(j) = (br - bl)/(ml - mr);% Calculating Y in excel file
        pog_X(j) = ml*pog_Y(j) + bl;% Calculating X in excel file

        lgQ = atan(Lg_Z(j)/Lg_Y(j))*180/pi;% Calculating Vertical(left)
gaze angle in excel file
        rgQ = atan(Rg_Z(j)/Rg_Y(j))*180/pi;% Calculating
Vertical(right) gaze angle in excel file
        lvo = (pog_Y(j)- LE_Y(j))* tan(lgQ*pi/180);% Calculating
Vertical(left) offset in excel file
        rvo = (pog_Y(j)- RE_Y(j))* tan(rgQ*pi/180);% Calculating
Vertical(right) offset in excel file
        lvi = lvo+ LE_Z(j);% Calculating Vertical(left) intercept in
excel file
        rvi = rvo+ RE_Z(j);% Calculating Vertical(right) intercept in
excel file
        pog_Z(j) = (lvi+ rvi)/2;% Calculating Z in excel file

        if flag == 1
            LED_X(j) = 0.554;
            LED_Y(j) = 1;
            LED_Z(j) = -0.6;
        end
        if flag == 2
            LED_X(j) = 0.364;
            LED_Y(j) = 1;
            LED_Z(j) = -0.6;
        end

        if flag == 3
            LED_X(j) = 0.173;
            LED_Y(j) = 1;
            LED_Z(j) = -0.6;
        end
        if flag == 4
            LED_X(j) = 0.554;
            LED_Y(j) = 1;
            LED_Z(j) = -0.409;
```

```
        end
        if flag == 5
            LED_X(j) = 0.364;
            LED_Y(j) = 1;
            LED_Z(j) = -0.409;
        end
        if flag == 6
            LED_X(j) = 0.173;
            LED_Y(j) = 1;
            LED_Z(j) = -0.409;
        end
        if flag == 7
            LED_X(j) = 0.554;
            LED_Y(j) = 1;
            LED_Z(j) = -0.219;
        end
        if flag == 8
            LED_X(j) = 0.364;
            LED_Y(j) = 1;
            LED_Z(j) = -0.219;
        end
        if flag == 9
            LED_X(j) = 0.173;
            LED_Y(j) = 1;
            LED_Z(j) = -0.219;
        end

        %compute error
        X_error(j) = pog_X(j) - LED_X(j);
        Y_error(j) = pog_Y(j) - LED_Y(j);
        Z_error(j) = pog_Z(j) - LED_Z(j);

        XYZ_error(j) = sqrt(X_error(j)^2+Y_error(j)^2+Z_error(j)^2);
    end
end

figure(2)
hold on
title('XYZ (3D) error')
xlabel('Frames');
ylabel('3D-Error');
%plot(XYZ_error(:,1), 'b')  %plot whole array
for i=1:2:16
    start = x(i);
    stop = x(i+1);
    plot(XYZ_error(start:stop),'b')  %plot array in pieces
end

%-------------------------------------------------------------------------
----
%  Compute intercept (2D) error
%-------------------------------------------------------------------------
----
```

```
X_cyc=zeros(file_len,1);
Y_cyc=zeros(file_len,1);
Z_cyc=zeros(file_len,1);
X_cal=zeros(file_len,1);
Z_cal=zeros(file_len,1);
Int_err_X=zeros(file_len,1);
Int_err_Z=zeros(file_len,1);
XZ_error=zeros(file_len,1);
X_left=zeros(file_len,1);
Z_left=zeros(file_len,1);
X_right=zeros(file_len,1);
Z_right=zeros(file_len,1);
L_Int_err_X=zeros(file_len,1);
R_Int_err_X=zeros(file_len,1);
L_Int_err_Z=zeros(file_len,1);
R_Int_err_Z=zeros(file_len,1);
L_XZ_error=zeros(file_len,1);
R_XZ_error=zeros(file_len,1);
errorL1x = zeros(35,1);
errorL1z = zeros(35,1);
errorL2x = zeros(35,1);
errorL2z = zeros(35,1);
errorL3x = zeros(35,1);
errorL3z = zeros(35,1);
errorL4x = zeros(35,1);
errorL4z = zeros(35,1);
errorL5x = zeros(35,1);
errorL5z = zeros(35,1);
errorL6x = zeros(35,1);
errorL6z = zeros(35,1);
errorL7x = zeros(35,1);
errorL7z = zeros(35,1);
errorL8x = zeros(35,1);
errorL8z = zeros(35,1);
errorL9x = zeros(35,1);
errorL9z = zeros(35,1);
errorR1x = zeros(35,1);
errorR1z = zeros(35,1);
errorR2x = zeros(35,1);
errorR2z = zeros(35,1);
errorR3x = zeros(35,1);
errorR3z = zeros(35,1);
errorR4x = zeros(35,1);
errorR4z = zeros(35,1);
errorR5x = zeros(35,1);
errorR5z = zeros(35,1);
errorR6x = zeros(35,1);
errorR6z = zeros(35,1);
errorR7x = zeros(35,1);
errorR7z = zeros(35,1);
errorR8x = zeros(35,1);
errorR8z = zeros(35,1);
errorR9x = zeros(35,1);
errorR9z = zeros(35,1);
```

```
error1x = zeros(35,1);
error1z = zeros(35,1);
error2x = zeros(35,1);
error2z = zeros(35,1);
error3x = zeros(35,1);
error3z = zeros(35,1);
error4x = zeros(35,1);
error4z = zeros(35,1);
error5x = zeros(35,1);
error5z = zeros(35,1);
error6x = zeros(35,1);
error6z = zeros(35,1);
error7x = zeros(35,1);
error7z = zeros(35,1);
error8x = zeros(35,1);
error8z = zeros(35,1);
error9x = zeros(35,1);
error9z = zeros(35,1);

for i=1:2:18
    start = x(i);
    stop = x(i+1);

    for j=start:stop
        % Calculating Cyclopian Eye;
        X_cyc(j)= (LE_X(j)+ RE_X(j))/2;
        Z_cyc(j)= (LE_Z(j)+ RE_Z(j))/2;
        Y_cyc(j)= (LE_Y(j)+ RE_Y(j))/2;

        % Calculating Intercept error (based on target depth);

        r = (LED_Y(j) - Y_cyc(j))/(pog_Y(j) - Y_cyc(j));
        X_cal(j)= X_cyc(j) + r*(pog_X(j)-X_cyc(j));
        Z_cal(j)= Z_cyc(j) + r*(pog_Z(j)-Z_cyc(j));
        X_left(j)= LE_X(j)+ r*(pog_X(j)- LE_X(j)) ;
        X_right(j)= RE_X(j)+ r*(pog_X(j)- RE_X(j));
        Z_left(j) = LE_Z(j) + r*(pog_Z(j)-LE_Z(j));
        Z_right(j) = RE_Z(j) + r*(pog_Z(j)-RE_Z(j));

        Int_err_X(j) = X_cal(j)-LED_X(j);
        Int_err_Z(j) = Z_cal(j)-LED_Z(j);
        L_Int_err_X(j) = X_left(j)-LED_X(j);
        R_Int_err_X(j) = X_right(j)-LED_X(j);
        L_Int_err_Z(j) = Z_left(j)+LED_Z(j);
        R_Int_err_Z(j) = Z_right(j)+LED_Z(j);

        XZ_error(j) = sqrt(Int_err_X(j)^2+Int_err_Z(j)^2);
        L_XZ_error(j) = sqrt(L_Int_err_X(j)^2+L_Int_err_Z(j)^2);
        R_XZ_error(j) = sqrt(R_Int_err_X(j)^2+R_Int_err_Z(j)^2);
        Mean_LX_error = mean(L_Int_err_X);
        Mean_LZ_error = mean(L_Int_err_Z);
        Mean_RX_error = mean(R_Int_err_X);
```

```
        Mean_RZ_error = mean(R_Int_err_Z);

    end
end
%-------------------------------------------------------------------------
-
%      Offset Calculation
%-------------------------------------------------------------------------
-
flag = 0;
for i=1:2:18
    start = x(i);
    stop = x(i+1);
    flag = flag+1;
    k=1;
    for j=start:stop

         if flag == 1
            errorL1x(k)= L_Int_err_X(j);
            errorL1z(k)= L_Int_err_Z(j);
            errorR1x(k)= R_Int_err_X(j);
            errorR1z(k)= R_Int_err_Z(j);
            error1x(k) = Int_err_X(j);
            error1z(k) = Int_err_Z(j);
            k = k+1;
            cnt = 0;
         end

        if flag == 2
            errorL2x(k)= L_Int_err_X(j);
            errorL2z(k)= L_Int_err_Z(j);
            errorR2x(k)= R_Int_err_X(j);
            errorR2z(k)= R_Int_err_Z(j);
            error2x(k) = Int_err_X(j);
            error2z(k) = Int_err_Z(j);
            k = k+1;
        end

        if flag == 3
            errorL3x(k)= L_Int_err_X(j);
            errorL3z(k)= L_Int_err_Z(j);
            errorR3x(k)= R_Int_err_X(j);
            errorR3z(k)= R_Int_err_Z(j);
            error3x(k) = Int_err_X(j);
            error3z(k) = Int_err_Z(j);
            k = k+1;
        end

        if flag == 4
            errorL4x(k)= L_Int_err_X(j);
            errorL4z(k)= L_Int_err_Z(j);
            errorR4x(k)= R_Int_err_X(j);
            errorR4z(k)= R_Int_err_Z(j);
```

```
    error4x(k) = Int_err_X(j);
    error4z(k) = Int_err_Z(j);
    k = k+1;
end

if flag == 5
    errorL5x(k)= L_Int_err_X(j);
    errorL5z(k)= L_Int_err_Z(j);
    errorR5x(k)= R_Int_err_X(j);
    errorR5z(k)= R_Int_err_Z(j);
    error5x(k) = Int_err_X(j);
    error5z(k) = Int_err_Z(j);
    k = k+1;
end

if flag == 6
    errorL6x(k)= L_Int_err_X(j);
    errorL6z(k)= L_Int_err_Z(j);
    errorR6x(k)= R_Int_err_X(j);
    errorR6z(k)= R_Int_err_Z(j);
    error6x(k) = Int_err_X(j);
    error6z(k) = Int_err_Z(j);
    k = k+1;

end

if flag == 7
    errorL7x(k)= L_Int_err_X(j);
    errorL7z(k)= L_Int_err_Z(j);
    errorR7x(k)= R_Int_err_X(j);
    errorR7z(k)= R_Int_err_Z(j);
    error7x(k) = Int_err_X(j);
    error7z(k) = Int_err_Z(j);
    k = k+1;
end

if flag == 8
    errorL8x(k)= L_Int_err_X(j);
    errorL8z(k)= L_Int_err_Z(j);
    errorR8x(k)= R_Int_err_X(j);
    errorR8z(k)= R_Int_err_Z(j);
    error8x(k) = Int_err_X(j);
    error8z(k) = Int_err_Z(j);
    k = k+1;
end

if flag == 9
    errorL9x(k)= L_Int_err_X(j);
    errorL9z(k)= L_Int_err_Z(j);
    errorR9x(k)= R_Int_err_X(j);
    errorR9z(k)= R_Int_err_Z(j);
    error9x(k) = Int_err_X(j);
```

```
            error9z(k) = Int_err_Z(j);
            k = k+1;
        end

    end
end

offsetL1x=mean(errorL1x);
offsetL1z=mean(errorL1z);
offsetR1x=mean(errorR1x);
offsetR1z=mean(errorR1z);

offsetL2x=mean(errorL2x);
offsetL2z=mean(errorL2z);
offsetR2x=mean(errorR2x);
offsetR2z=mean(errorR2z);

offsetL3x=mean(errorL3x);
offsetL3z=mean(errorL3z);
offsetR3x=mean(errorR3x);
offsetR3z=mean(errorR3z);

offsetL4x=mean(errorL4x);
offsetL4z=mean(errorL4z);
offsetR4x=mean(errorR4x);
offsetR4z=mean(errorR4z);

offsetL5x=mean(errorL5x);
offsetL5z=mean(errorL5z);
offsetR5x=mean(errorR5x);
offsetR5z=mean(errorR5z);

offsetL6x=mean(errorL6x);
offsetL6z=mean(errorL6z);
offsetR6x=mean(errorR6x);
offsetR6z=mean(errorR6z);

offsetL7x=mean(errorL7x);
offsetL7z=mean(errorL7z);
offsetR7x=mean(errorR7x);
offsetR7z=mean(errorR7z);

offsetL8x=mean(errorL8x);
offsetL8z=mean(errorL8z);
offsetR8x=mean(errorR8x);
offsetR8z=mean(errorR8z);

offsetL9x=mean(errorL9x);
offsetL9z=mean(errorL9z);
offsetR9x=mean(errorR9x);
offsetR9z=mean(errorR9z);
```

```
offset1x=mean(error1x);
offset1z=mean(error1z);
offset2x=mean(error2x);
offset2z=mean(error2z);
offset3x=mean(error3x);
offset3z=mean(error3z);
offset4x=mean(error4x);
offset4z=mean(error4z);
offset5x=mean(error5x);
offset5z=mean(error5z);
offset6x=mean(error6x);
offset6z=mean(error6z);
offset7x=mean(error7x);
offset7z=mean(error7z);
offset8x=mean(error8x);
offset8z=mean(error8z);
offset9x=mean(error9x);
offset9z=mean(error9z);
offsetx =
(offset1x+offset2x+offset3x+offset4x+offset5x+offset6x+offset7x+offset8
x+offset9x)/9;
offsetz =
(offset1z+offset2z+offset3z+offset4z+offset5z+offset6z+offset7z+offset8
z+offset9z)/9;
flag = 0;
for i=1:2:18
    start = x(i);
    stop = x(i+1);
    flag = flag+1;

    for j=start:stop
        if flag == 1
            X_cal(j) = X_cal(j)- offsetx;
            Z_cal(j) = Z_cal(j)- offsetz;
            X_left(j)= X_left(j)- offsetx;
            X_right(j)= X_right(j)- offsetx;
            Z_left(j)= Z_left(j)- offsetz;
            Z_right(j)= Z_right(j)- offsetz;
        end
        if flag == 2
            X_cal(j) = X_cal(j)- offsetx;
            Z_cal(j) = Z_cal(j)- offsetz;
            X_left(j)= X_left(j)- offsetx;
            X_right(j)= X_right(j)- offsetx;
            Z_left(j)= Z_left(j)- offsetz;
            Z_right(j)= Z_right(j)- offsetz;
        end

        if flag == 3
            X_cal(j) = X_cal(j)- offsetx;
            Z_cal(j) = Z_cal(j)- offsetz;
            X_left(j)= X_left(j)- offsetx;
```

```
    X_right(j)= X_right(j)- offsetx;
    Z_left(j)= Z_left(j)- offsetz;
    Z_right(j)= Z_right(j)- offsetz;
end
if flag == 4
    X_cal(j) = X_cal(j)- offsetx;
    Z_cal(j) = Z_cal(j)- offsetz;
    X_left(j)= X_left(j)- offsetx;
    X_right(j)= X_right(j)- offsetx;
    Z_left(j)= Z_left(j)- offsetz;
    Z_right(j)= Z_right(j)- offsetz;
end
if flag == 5
    X_cal(j) = X_cal(j)- offsetx;
    Z_cal(j) = Z_cal(j)- offsetz;
    X_left(j)= X_left(j)- offsetx;
    X_right(j)= X_right(j)- offsetx;
    Z_left(j)= Z_left(j)- offsetz;
    Z_right(j)= Z_right(j)- offsetz;
end
if flag == 6
    X_cal(j) = X_cal(j)- offsetx;
    Z_cal(j) = Z_cal(j)- offsetz;
    X_left(j)= X_left(j)- offsetx;
    X_right(j)= X_right(j)- offsetx;
    Z_left(j)= Z_left(j)- offsetz;
    Z_right(j)= Z_right(j)- offsetz;
end
if flag == 7
    X_cal(j) = X_cal(j)- offsetx;
    Z_cal(j) = Z_cal(j)- offsetz;
    X_left(j)= X_left(j)- offsetx;
    X_right(j)= X_right(j)- offsetx;
    Z_left(j)= Z_left(j)- offsetz;
    Z_right(j)= Z_right(j)- offsetz;
end
if flag == 8
    X_cal(j) = X_cal(j)- offsetx;
    Z_cal(j) = Z_cal(j)- offsetz;
    X_left(j)= X_left(j)- offsetx;
    X_right(j)= X_right(j)- offsetx;
    Z_left(j)= Z_left(j)- offsetz;
    Z_right(j)= Z_right(j)- offsetz;
end
if flag == 9
    X_cal(j) = X_cal(j)- offsetx;
    Z_cal(j) = Z_cal(j)- offsetz;
    X_left(j)= X_left(j)- offsetx;
    X_right(j)= X_right(j)- offsetx;
    Z_left(j)= Z_left(j)- offsetz;
    Z_right(j)= Z_right(j)- offsetz;
end
```

```
    end
end

figure(3)
hold on
title('XZ (2D) error')
xlabel('Frames)');
ylabel('2-D error');

for i=1:2:18
    start = x(i);
    stop = x(i+1);
    plot(XZ_error(start:stop),'b')  %plot array in pieces
end

figure (4)
hold on
title('Scatter plot (LEDs vs. point-of-interception)')
xlabel('X co-ordinate');
ylabel('-Z co-ordinate');
display(sprintf('%s','2D Intercept Error'));
fix = 1;
for i=1:2:18
    start = x(i);
    stop = x(i+1);
    plot(X_cal(start:stop),-Z_cal(start:stop),'b','LineWidth',5)
    plot(LED_X(start:stop),-LED_Z(start:stop),'rs','LineWidth',2,
'MarkerSize' ,5)

    ave_err = mean(XZ_error(start:stop))*1000;     %convert to mm
    stdev = std(XZ_error(start:stop))*1000;        %convert to mm
    X_err = mean(Int_err_X(start:stop))*1000;      %convert to mm
    X_stdev = std(Int_err_X(start:stop))*1000;     %convert to mm
    Z_err = mean(Int_err_Z(start:stop))*1000;      %convert to mm
    Z_stdev = std(Int_err_Z(start:stop))*1000;     %convert to mm

    display(sprintf('fixation(%d) = %0.3f ± %0.3f (X-> %0.3f ± %0.3f)
(Z-> %0.3f ± %0.3f)',fix,ave_err,stdev,X_err,X_stdev,Z_err,Z_stdev));

    fix = fix + 1;
end

display(sprintf('\n'));

figure (5)
hold on
title('Scatter plot (LEDs vs. point-of-interception)[Left Eye]')
xlabel('X co-ordinate');
ylabel('-Z co-ordinate');
display(sprintf('%s','2D Intercept Error[Left Eye]'));
fix = 1;
for i=1:2:18
```

```
    start = x(i);
    stop = x(i+1);
    plot(X_left(start:stop),-Z_left(start:stop),'b','LineWidth',5)
    plot(LED_X(start:stop),-LED_Z(start:stop),'rs','LineWidth',2,
'MarkerSize' ,5)

    ave_err_left = mean(L_XZ_error(start:stop))*1000;     %convert to
mm
    stdev_left = std(L_XZ_error(start:stop))*1000;        %convert to
mm
    X_err_left = mean(L_Int_err_X(start:stop))*1000;      %convert to
mm
    X_stdev_left = std(L_Int_err_X(start:stop))*1000;     %convert to
mm
    Z_err_left = mean(L_Int_err_Z(start:stop))*1000;      %convert to
mm
    Z_stdev_left = std(L_Int_err_Z(start:stop))*1000;     %convert to
mm

    display(sprintf('fixation(%d) = %0.3f ± %0.3f (X-> %0.3f ± %0.3f)
(Z-> %0.3f ±
%0.3f)',fix,ave_err_left,stdev_left,X_err_left,X_stdev_left,Z_err_left,
Z_stdev_left));

    fix = fix + 1;
end

display(sprintf('\n'));


figure (6)
hold on
title('Scatter plot (LEDs vs. point-of-interception)[Right Eye]')
xlabel('X co-ordinate');
ylabel('-Z co-ordinate');
display(sprintf('%s','2D Intercept Error[Right Eye]'));
fix = 1;
for i=1:2:18
    start = x(i);
    stop = x(i+1);
    plot(X_right(start:stop),-Z_right(start:stop),'b','LineWidth',5)
    plot(LED_X(start:stop),-LED_Z(start:stop),'rs','LineWidth',2,
'MarkerSize' ,5)

    ave_err_right = mean(R_XZ_error(start:stop))*1000;     %convert to
mm
    stdev_right = std(R_XZ_error(start:stop))*1000;        %convert to
mm
    X_err_right = mean(R_Int_err_X(start:stop))*1000;      %convert to
mm
    X_stdev_right = std(R_Int_err_X(start:stop))*1000;     %convert to
mm
```

```
    Z_err_right = mean(R_Int_err_Z(start:stop))*1000;       %convert to
mm
    Z_stdev_right = std(R_Int_err_Z(start:stop))*1000;      %convert to
mm

    display(sprintf('fixation(%d) = %0.3f ± %0.3f (X-> %0.3f ± %0.3f)
(Z-> %0.3f ±
%0.3f)',fix,ave_err_right,stdev_right,X_err_right,X_stdev_right,Z_err_r
ight,Z_stdev_right));

    fix = fix + 1;
end

display(sprintf('\n'));
```

APPENDIX C: MATLAB code for analyzing saccadic data.

```
%-------------------------------------------------------------------------
----
%  SACCADE.M -- this program...
%-------------------------------------------------------------------------
----
close all
clear

filename=input('Select File to Run: ','s')
Y=load(filename);
frames=Y(:,1);
LE_X=Y(:,2);    LE_Y=Y(:,3);    LE_Z=Y(:,4);
RE_X=Y(:,5);    RE_Y=Y(:,6);    RE_Z=Y(:,7);

Lg_X=Y(:,8);   Lg_Y=Y(:,9);   Lg_Z=Y(:,10);
Rg_X=Y(:,11);   Rg_Y=Y(:,12);   Rg_Z=Y(:,13);

Lhorz_eye_angle=Y(:,14); Lvert_eye_angle=Y(:,15);
LED_data=Y(:,16);

file_len=length(LE_X);
RADIAN=3.14/180;
sample_freq = 250;
sample_rate = 1/sample_freq;
Z=0;% Control for Plotting the Diagnostic Graphs

%-------------------------------------------------------------------------
----
%  Plot of LED blinks and Eye Response
%-------------------------------------------------------------------------
----
figure(1)
hold on
title('Eye Angle')
xlabel('Frames');
ylabel('Eye Angle(degrees)');
plot(-Lhorz_eye_angle(:,1), 'b')

query = input('Mark Blinks?Y/N =','s')

if query == 'n'
    F = input( 'Eye angle file:','s')
    Lhorz_eye_angle = load(F);
%     dlmread('y.out',';','C1');

elseif query == 'y'
    [Eye_angle] = ginput(6);
    for i =1:2:6
        start = Eye_angle(i);
```

```
        stop = Eye_angle(i+1);
        start = round(start);
        stop = round(stop);
        for j = start:stop
            Lhorz_eye_angle(j)=Lhorz_eye_angle (j-1);
        end
    end
    %--------------------------------------------
    %   Exporting Eye Angle values
    %--------------------------------------------
    save eye_angle.out Lhorz_eye_angle -ASCII;
    dlmwrite ('Lhorz_eye_angle',Lhorz_eye_angle,';');
end



figure(2)
hold on
title('Eye(blue) and LED(green) plot')
xlabel('Time(seconds)');
ylabel('Eye Angle(degrees)/LED switching pattern');
plot(frames/250,-Lhorz_eye_angle(:,1), 'b')
plot(frames/250,LED_data(:,1), 'g')

vel_L = zeros(file_len,1);
vel_E = zeros(file_len,1);
ax_E = zeros(file_len,1);

% directional variables

directiona = zeros(file_len,1);
directionb = zeros(file_len,1);
velT = 10;                    % set velocity threshold to 10 deg/sec
vel_Lr = zeros(file_len,1);
vel_Ll = zeros(file_len,1);
vel_Er = zeros(file_len,1);
vel_El = zeros(file_len,1);
ax_Er = zeros(file_len,1);
ax_El = zeros(file_len,1);
Eye_set = zeros(39,1);
rcnt = 1;
lcnt = 1;
r=1;f=0;g=0;

for i=2:file_len-10
    vel_L(i)=((LED_data(i+1)-LED_data(i-1))/(2*sample_rate));
    vel_E(i)=((Lhorz_eye_angle(i+1)-Lhorz_eye_angle(i-
1))/(2*sample_rate));
end
for i=2:file_len-10
    ax_E(i)= (abs((vel_E(i+1))-abs(vel_E(i-1)))/(2*sample_rate));
    if (vel_L(i) > velT)
```

```
        directiona(i) = -500;
        vel_Ll(lcnt) = vel_L(i);
        lcnt = lcnt + 1;
        f=-1;
    end
    if (vel_L(i) < -velT)
        directiona(i) = 500;
        vel_Lr(rcnt) = vel_L(i);
        rcnt = rcnt + 1;
        f=1;
    end
    if (abs(vel_E(i)) > 3*velT)&& f==1
        if max(abs(vel_E(i:i+10)))>100
            directionb(i) = +500;
            vel_Er(i) = vel_E(i);
            ax_Er(i) = ax_E(i);
            g=1;
    elseif max(abs(vel_E(i:i+10)))<100 && g==1
            directionb(i) = +500;
            vel_Er(i) = vel_E(i);
            ax_Er(i) = ax_E(i);
        else
                g=0;
        end
    end
    if (abs(vel_E(i)) > 3*velT)&& f==-1
         if max(abs(vel_E(i:i+10)))>100
             directionb(i) = -500;
             vel_El(i) = vel_E(i);
             ax_El(i) = ax_E(i);
             g=-1;
         elseif max(abs(vel_E(i:i+10)))<100 && g==-1
             directionb(i) = -500;
             vel_El(i) = vel_E(i);
             ax_Er(i) = ax_E(i);
         else
                 g=0;
         end
    end

end
%-------------------------------------------------------------------------
----
%  Find peak, mean and std. dev of left n right saccade velocites
%-------------------------------------------------------------------------
----
flag = 0;
for i = 1:file_len
    if directiona(i) ~= 0 && flag == 0
        start_point=i;
        flag = 1;
    end
end
```

```
rsac=zeros(100,1);
lsac=zeros(100,1);
j=1;k=1;cnt1=0;lcnt=0;
for i=start_point:file_len
    if vel_Er(i)~=0
        rsac(j) = i;
        j=j+1;
        rcnt=rcnt+1;
    end
    if vel_Er(i) == 0 && rcnt < 10
        j = j-rcnt;
        rcnt = 0;
    end
    if vel_Er(i) == 0 && rcnt > 10
        rcnt=0;
    end
    if vel_El(i)~=0
        lsac(k) = i;
        k=k+1;
        lcnt=lcnt+1;
    end
    if vel_El(i) == 0 && lcnt < 10
        k = k-lcnt;
        lcnt = 0;
    end
    if vel_El(i) == 0 && lcnt > 10
        lcnt=0;
    end
end
n=0;j=1;k=1;
r_odd=zeros(file_len,1);
r_marker=zeros(10,1);
R_Saccade=zeros(10,1);
max_velR = zeros(10,1);
l_odd=zeros(file_len,1);
l_marker=zeros(10,1);
L_Saccade=zeros(10,1);
max_velL = zeros(10,1);
durationL = zeros(10,1);
durationR = zeros(10,1);
max_axR = zeros(10,1);
max_axL = zeros(10,1);
ax_timeR = zeros(10,1);
ax_timeL = zeros(10,1);
R_Saccade(1) = rsac(1);
L_Saccade(1) = lsac(1);
rdiff = diff(rsac);
ldiff = diff(lsac);
for i = 1:length(rdiff)
    if rdiff(i)~=1
        r_odd(j) =i;
        j=j+1;
    end
end
```

```
if rdiff(length(rdiff))==1 % for including last saccade
    r_odd(j)=r_odd(j-1);
    r_odd(j+1)=length(rsac);
end
a = r_odd(1);
R_Saccade(2) = rsac(a);
for i = 1:length(ldiff)
    if ldiff(i)~=1
        l_odd(k) =i;
        k=k+1;
    end
end
if ldiff(length(ldiff))==1 % for including last saccade
    l_odd(k)=l_odd(k-1);
    l_odd(k+1)=length(lsac);
end
b = l_odd(1);
L_Saccade(2) = lsac(b);
j=1;k=1;
for i = 1:length(rdiff)
    if r_odd(i+1) - r_odd(i)>=2
        r_marker(j)= r_odd(i)+1;
        r_marker(j+1)= r_odd(i+1);
        j=j+2;
    end
%     if r_odd(i+1) - r_odd(i)==1
%         r_marker(j)= r_odd(i+1);
%         r_marker(j+1)= r_odd(i+1);
%         j=j+2;
%     end
end
for i = 1:length(ldiff)
    if l_odd(i+1) - l_odd(i)>=2
        l_marker(k)= l_odd(i)+1;
        l_marker(k+1)= l_odd(i+1);
        k=k+2;
    end
%     if l_odd(i+1) - l_odd(i)==1
%         l_marker(k)= l_odd(i+1);
%         l_marker(k+1)= l_odd(i+1);
%         k=k+2;
%     end
end
j=3;k=3;
for i=1:2:length(r_marker)
    a = r_marker(i);
    b = r_marker(i+1);
    R_Saccade(j) = rsac(a);
    R_Saccade(j+1)=rsac(b);
    j=j+2;
end
for i=1:2:length(l_marker)
    c = l_marker(i);
    d = l_marker(i+1);
```

```
    L_Saccade(k) = lsac(c);
    L_Saccade(k+1)=lsac(d);
    k=k+2;
end
j=1;k=1;
for i = 1:2:length(R_Saccade)
    start = R_Saccade(i);
    stop = R_Saccade(i+1);
    max_velR(j) = max(vel_Er(start:stop));
    max_axR(j) = max(ax_Er(start+1:stop-1));
    durationR(j) = (stop-start)/250;
    for q = start:stop
        if ax_Er(q) == max_axR(j)
            ax_timeR(j) = (q-start)/250;
        end
    end
    j=j+1;
    avg_max_velR = mean(max_velR);
    std_max_velR = std(max_velR);

end
for i = 1:2:length(L_Saccade)
    start = L_Saccade(i);
    stop = L_Saccade(i+1);
    max_velL(k) = max(abs(vel_El(start:stop)));
    max_axL(k) = max(ax_El(start+1:stop-1));
    durationL(k) = (stop-start)/250;
    for q = start:stop
        if ax_El(q) == max_axL(k)
            ax_timeL(k) = (q-start)/250;
        end
    end
    k=k+1;
    avg_max_velL = mean(max_velL);
    std_max_velL = std(max_velL);

end

figure(3)
hold on
title('Onset of LED illumination(Green) & Eye Velocity(Blue)')
plot (-vel_L(:,1),'g')
plot (vel_E(:,1),'b')
% plot (directiona(:,1),'r')   % plot velocity direction array to test
% plot (directionb(:,1),'k')
xlabel('Frames');
ylabel('LED blinks / Eye velocity')

figure(4)
hold on
title('Left Saccade velocity(Green) & Right Saccade Velocity(Blue)')
plot (frames*sample_rate,vel_El(:,1),'g')
plot (frames*sample_rate,vel_Er(:,1),'b')
```

```
xlabel('Time');
ylabel('Eye Velocity (deg/sec)')

%-------------------------------------------------------------------------
----
%  Time Lag Calculation.
%-------------------------------------------------------------------------
----
LED_blink = zeros(file_len,1);
Eye_fix = zeros(file_len,1);
led_blink = zeros(file_len,1);
eye_fix = zeros(file_len,1);
j=1;k=1;
for i=1:file_len
    if abs(directiona(i)) == 500
    led_blink(j)=i;
    j=j+1;
    end
    if abs(directionb(i)) == 500 && abs(mean (directionb(i:i+5)))==500
    eye_fix(k)=i;
    k=k+1;
    end
end
LED_blink(1) = led_blink(1);
j=2;k=1;l=2;
for i=2:file_len-1
    if led_blink(i+1)-led_blink(i) >1
        LED_blink(j) = led_blink(i+1);
        j=j+1;
    end
    if eye_fix(i)>led_blink(1)
        eye_fix(k) = eye_fix(i);
        k=k+1;
    end
end
Eye_fix(1) = eye_fix(1);
for i=2:file_len-1
    if eye_fix(i+1)-eye_fix(i) >1
        Eye_fix(l) = eye_fix(i+1);
        l=l+1;
    end
end
%-------------------------------------------------------------------------
----
%  Settling Time Calculation.
%-------------------------------------------------------------------------
----
k=1;
for i = 1:file_len
    if Eye_fix(i)~=0
        flag = 0;
        a = Eye_fix(i);
        for j=a:file_len
            if directionb(j)==0 && flag==0
```

```
                Eye_set(k) = j;
                flag = 1;
                k = k+1;
            end
        end
    end
end

time_lag = zeros(34,1);
L_time_lag = zeros(17,1);
R_time_lag = zeros(17,1);
settle_time = zeros(39,1);
settle_timeR = zeros(20,1);
settle_timeL = zeros(19,1);
saccade_number = (1:1:20);
set_ampL = zeros(19,1);
set_ampR = zeros(19,1);
set_amp = zeros(38,1);
for i=1:40
    time_lag(i)= (Eye_fix(i)-LED_blink(i))*sample_rate;
end
j = 1;
for i = 1:2:39
    R_time_lag(j) = time_lag(i);
    L_time_lag(j) = time_lag(i+1);
    j = j+1;
end
for i = 1:39
    settle_time(i) = (Eye_set(i)-Eye_fix(i))*sample_rate;
end
j=1;
for i = 1:2:38
    settle_timeR(j) = settle_time(i);
    settle_timeL(j) = settle_time(i+1);
    j=j+1;
end
for i = 1:length(L_time_lag)
    j = L_time_lag(i);
    set_ampL(i) = j;
    j=j+1;
end
for i = 1:length(R_time_lag)
    k = L_time_lag(i);
    set_ampR(i) = k;
    k=k+1;
end
for i = 1:length(time_lag)
    l = time_lag(i);
    set_amp(i) = l;
    l=l+1;
end
avg_settle_timeL = mean(settle_timeL);
avg_settle_timeR = mean(settle_timeR);
```

```
std_settle_timeL = std(settle_timeL);
std_settle_timeR = std(settle_timeR);
Left_Latency = mean(L_time_lag);
Right_Latency = mean(R_time_lag);
std_Left_Latency = std(L_time_lag);
std_Right_Latency = std(R_time_lag);
figure(5)
hold on
title('Time Delay for Left(Green) & Right(Blue) Saccades')
plot (saccade_number,L_time_lag,'.g')
plot (saccade_number,R_time_lag,'.b')
xlabel('Saccade Number');
ylabel('Time Delay');

%-------------------------------------------------------------------------
----
%  calculate error between POG and LED
%-------------------------------------------------------------------------
----
RX_error=zeros(file_len,1);
RY_error=zeros(file_len,1);
RZ_error=zeros(file_len,1);
LX_error=zeros(file_len,1);
LY_error=zeros(file_len,1);
LZ_error=zeros(file_len,1);
pog_X=zeros(file_len,1);
pog_Y=zeros(file_len,1);
pog_Z=zeros(file_len,1);
LXYZ_error=zeros(file_len,1);
RXYZ_error=zeros(file_len,1);

for j=1:file_len

        %compute eye point-of-gaze
        if Lg_Y(j) ~= 0
            ml = Lg_X(j)/Lg_Y(j);% Calculating m1 in excel file
        end
        if Rg_Y(j) ~= 0
            mr = Rg_X(j)/Rg_Y(j);% Calculating m2 in excel file
        end
        bl = abs(LE_Y(j))* ml + LE_X(j);% Calculating b1 in excel file
        br = abs(RE_Y(j))* mr + RE_X(j);% Calculating b2 in excel file
        pog_Y(j) = (br - bl)/(ml - mr);% Calculating Y in excel file
        pog_X(j) = ml*pog_Y(j) + bl;% Calculating X in excel file

        if Lg_Y(j) ~= 0
            lgQ = atan(Lg_Z(j)/Lg_Y(j))*180/pi;% Calculating
Vertical(left) gaze angle in excel file
        end
        if Rg_Y(j) ~= 0
            rgQ = atan(Rg_Z(j)/Rg_Y(j))*180/pi;% Calculating
Vertical(right) gaze angle in excel file
        end
```

```
        lvo = (pog_Y(j)- LE_Y(j))*tan(lgQ*pi/180);% Calculating
Vertical(left) offset in excel file
        rvo = (pog_Y(j)- RE_Y(j))*tan(rgQ*pi/180);% Calculating
Vertical(right) offset in excel file
        lvi = lvo+ LE_Z(j);% Calculating Vertical(left) intercept in
excel file
        rvi = rvo+ RE_Z(j);% Calculating Vertical(right) intercept in
excel file
        pog_Z(j) = (lvi+ rvi)/2;% Calculating Z in excel file

        %compute error
        for i=1:file_len
            if directionb(i) == 500
                RX_error(i) = pog_X(i) - 0.173;
                RY_error(i) = pog_Y(i) - 1;
                RZ_error(i) = pog_Z(i) - 0.409;
                RXYZ_error(i) =
sqrt(RX_error(i)^2+RY_error(i)^2+RZ_error(i)^2);
            end

            if  directionb(i) == -500
                LX_error(i) = pog_X(i) - 0.554;
                LY_error(i) = pog_Y(i) - 1;
                LZ_error(i) = pog_Z(i) - 0.409;
                LXYZ_error(i) =
sqrt(LX_error(i)^2+LY_error(i)^2+LZ_error(i)^2);
            end
        end
end

LED_positionLX=zeros(file_len,1);
LED_positionLZ=zeros(file_len,1);
LED_positionRX=zeros(file_len,1);
LED_positionRZ=zeros(file_len,1);

X_cyc=zeros(file_len,1);
Y_cyc=zeros(file_len,1);
Z_cyc=zeros(file_len,1);
X_cal=zeros(file_len,1);
Z_cal=zeros(file_len,1);
Int_err_X=zeros(file_len,1);
Int_err_Z=zeros(file_len,1);
XZ_error=zeros(file_len,1);
X_left=zeros(file_len,1);
Z_left=zeros(file_len,1);
X_right=zeros(file_len,1);
Z_right=zeros(file_len,1);
L_Int_err_X=zeros(file_len,1);
R_Int_err_X=zeros(file_len,1);
L_Int_err_Z=zeros(file_len,1);
R_Int_err_Z=zeros(file_len,1);
L_XZ_error=zeros(file_len,1);
R_XZ_error=zeros(file_len,1);
```

```
XL_eye=zeros(file_len,1);
XR_eye=zeros(file_len,1);
ZL_eye=zeros(file_len,1);
ZR_eye=zeros(file_len,1);
XRR = zeros(file_len,1);
ZRR = zeros(file_len,1);
XRL = zeros(file_len,1);
ZRL = zeros(file_len,1);
XLL = zeros(file_len,1);
ZLL = zeros(file_len,1);
XLR = zeros(file_len,1);
ZLR = zeros(file_len,1);
yL=ones(file_len,1);
yR=ones(file_len,1);
frameL = zeros(file_len,1);
frameR = zeros(file_len,1);
ampL = zeros(100,1);
ampR = zeros(100,1);

for j = 1:file_len
    if LED_data(j) < 0   % Right LED negative Left LED positive
        LED_positionRX(j) = 0.554;
        LED_positionRZ(j) = 0.409;

    end
    if LED_data(j) > 0
        LED_positionLX(j) = 0.173;
        LED_positionLZ(j) = 0.409;

    end

        % Calculating Cyclopian Eye;
        X_cyc(j)= (LE_X(j)+ RE_X(j))/2;
        Z_cyc(j)= (LE_Z(j)+ RE_Z(j))/2;
        Y_cyc(j)= (LE_Y(j)+ RE_Y(j))/2;

        % Calculating Intercept error (based on target depth);

        r = (1 - Y_cyc(j))/(pog_Y(j) - Y_cyc(j));
        X_cal(j)= X_cyc(j) + r*(pog_X(j)-X_cyc(j));
        Z_cal(j)= Z_cyc(j) + r*(pog_Z(j)-Z_cyc(j));
        X_left(j)= LE_X(j)+ r*(pog_X(j)- LE_X(j));
        X_right(j)= RE_X(j)+ r*(pog_X(j)- RE_X(j));
        Z_left(j) = LE_Z(j) + r*(pog_Z(j)-LE_Z(j));
        Z_right(j) = RE_Z(j) + r*(pog_Z(j)-RE_Z(j));
        L_Int_err_X(j) = X_left(j)-LED_positionLX(j);
        R_Int_err_X(j) = X_right(j)-LED_positionRX(j);
        L_Int_err_Z(j) = Z_left(j)-LED_positionLZ(j);
        R_Int_err_Z(j) = Z_right(j)-LED_positionRZ(j);

        L_XZ_error(j) = sqrt(L_Int_err_X(j)^2+L_Int_err_Z(j)^2);
        R_XZ_error(j) = sqrt(R_Int_err_X(j)^2+R_Int_err_Z(j)^2);
```

```
end
LED_data = round(LED_data)-1;
for j = 2:file_len
    if LED_data(j) < 0% Right LED negative Left LED positive
        XR_eye(j) = X_right(j);
        ZR_eye(j) = Z_right(j);
        XL_eye(j) = X_left(j);
        ZL_eye(j) = Z_left(j);
        XRR(j)=XR_eye(j)-3.5*mean(XR_eye(2:100));% zeroed value[1)-
3.5;2)-3.3;3)-3.3;4)-1]
        ZRR(j)=ZR_eye(j);
        XRL(j)=XL_eye(j)-3.5*mean(XL_eye(2:100));% zeroed value[1)-
3.5;2)-3.3;3)-3.3;4)-1]
        ZRL(j)=ZL_eye(j);
        frameR(j) = frameR(j-1)+1;
        if XRR(j)==0
            frameR(j)=0;
        end

    end
    if LED_data(j) > 0% Right LED negative Left LED positive
        XL_eye(j) = X_left(j);
        ZL_eye(j) = Z_left(j);
        XR_eye(j) = X_right(j);
        ZR_eye(j) = Z_right(j);
        XLL(j) = XL_eye(j)-1*mean(XL_eye(2:100));% zeroed value[1)-
1;2)-1;3)-1;4)-0.3]
        ZLL(j) = ZL_eye(j);
        XLR(j) = XR_eye(j)-1*mean(XR_eye(2:100));% zeroed value[1)-
1;2)-1;3)-1;4)-0.3]
        ZLR(j) = ZR_eye(j);
        frameL(j) = frameL(j-1)+1;
        if XLR(j)==0
            frameL(j)=0;
        end
    end
end

figure (6)
hold on
title('LED position Vs Point of Interception (PoI) [blue=left eye,
green=right eye]')
xlabel('X co-ordinate');
ylabel('Z co-ordinate');
    plot(XL_eye,-ZL_eye,'.b')
    plot(XR_eye,-ZR_eye,'.g')
    plot (0.554, 0.409,'rs','LineWidth',2,'MarkerSize' ,5)
    plot (0.173, 0.409,'rs','LineWidth',2,'MarkerSize' ,5)

for j = 1:file_len
    yL(j) = 0.554*yL(j)-1*mean(XL_eye(2:100));% zeroed value[1)-1;2)-
1;3)3)-1;4)-0.3]
```

```
    yR(j) = 0.173*yR(j)-3.5*mean(XR_eye(2:100));% zeroed value[1)-
3.5;2)-3.3;3)-3.3;4)-1]
end

figure (7)
hold on
title('Temporal Plot of LED position[Red] Vs Point of Interception
(PoI) [blue=left eye, green=right eye]& error(black)')
xlabel('Time (seconds)');
ylabel('X co-ordinate (Top=left LED;Bottom=right LED)');
plot(frameR*sample_rate,XRR,'.g')% Right LED negative Left LED positive
plot(frameR*sample_rate,XRL,'.b')

plot(frameL*sample_rate,XLR,'.g')% Right LED negative Left LED positive
plot(frameL*sample_rate,XLL,'.b')

plot (frameL*sample_rate,yL,'r')
plot (frameR*sample_rate,yR,'r')
axis([0,1000*sample_rate,-0.7,0.7])

figure (8)
hold on
title('Temporal Plot of LED position[Red] Vs Point of Interception
(PoI) [blue=left eye, green=right eye]& error(black)')
xlabel('Time (seconds)');
ylabel('X co-ordinate (Top=left LED;Bottom=right LED)');
plot(frameR*sample_rate,XRR,'.g')% Right LED negative Left LED positive
plot(frameR*sample_rate,XRL,'.b')

plot(frameL*sample_rate,XLR,'.g')% Right LED negative Left LED positive
plot(frameL*sample_rate,XLL,'.b')

plot (frameL*sample_rate,yL,'r')
plot (frameR*sample_rate,yR,'r')
axis([0,500*sample_rate,-0.7,0.7])
j=1;k=1;

for i = 100:file_len
   if XLR(i) ~= 0
      ampL(j) = XLR(i);
      j = j+1;
   end
   if XRL(i) ~= 0
      ampR(k) = XRL(i);
      k = k+1;
   end
end

avg_ampL = mean(ampL);
std_ampL = std(ampL);
avg_ampR = mean(ampR);
std_ampR = std(ampR);
```

```
max_ampL = max(ampL);
min_ampL = min(ampL);
min_ampR = max(ampR);
max_ampR = min(ampR);
no_of_saccades = ((length(settle_time)-1)/2);
%-------------------------------------------------------------------------
----
%  calculate eye angles while staring target
%-------------------------------------------------------------------------
----
L_Eye_angle = zeros(39,1);
%R_Eye_angle = zeros(39,1);
L_Eye_angleL = zeros(18,1);
L_Eye_angleR = zeros(18,1);
%R_Eye_angleR = zeros(file_len,1);
%R_Eye_angleL = zeros(file_len,1);
j = 1;k = 1;l = 1;
% m = 1; n = 1;
for i = 1:39
    a = Eye_set(i);
    b = a+100;
    L_Eye_angle(j) = mean(Lhorz_eye_angle (a:b));
    %R_Eye_angle(j) = mean(Rhorz_eye_angle (a:b));

    if L_Eye_angle(j)>0
        L_Eye_angleL(k) = L_Eye_angle(j);
        k=k+1;
    end
    if L_Eye_angle(j)<0
        L_Eye_angleR(l) = L_Eye_angle(j);
        l=l+1;
    end
end



avg_L_Eye_angleL = mean(L_Eye_angleL);
avg_L_Eye_angleR = mean(L_Eye_angleR);
% avg_R_Eye_angleL = mean(R_Eye_angleL);
% avg_R_Eye_angleR = mean(R_Eye_angleR);

figure(9)
hold on
title('Eye(blue) and LED(green) plot')
xlabel('Frames');
ylabel('Eye Angle(degrees)/LED switching pattern');
plot(-Lhorz_eye_angle(:,1), 'b')
plot(LED_data(:,1), 'g')
plot (vel_El(:,1),'k')
plot (vel_Er(:,1),'k')

%-------------------------------------------------------------------------
----
```

```
%  calculate saccadic error
%-------------------------------------------------------------------------
----
Error_R=zeros(20,1);
er_R=zeros(file_len,1);
Error_L=zeros(20,1);
er_L=zeros(file_len,1);
Max_sac_velR=zeros(20,1);
Max_sac_velL=zeros(20,1);
Sac_ampR=zeros(20,1);
Sac_ampL=zeros(20,1);

for i = 1:file_len
    if XRR(i) ~=0
        er_R(i)= XRR(i)-yR(i);
    end
    if XLR(i) ~=0
        er_L(i)= XLR(i)-yL(i);
    end
end
k=1;l=1;
for i = 1:2:length(R_Saccade)
    start = R_Saccade(i);
    stop = R_Saccade(i+1);
    Max_sac_velR(k) = max(abs(vel_Er(start:stop)));
    Sac_ampR(k) = abs(Lhorz_eye_angle(stop)-Lhorz_eye_angle(start));
    Error_R(k)= (mean(er_R(stop:stop+10)))*1000;
    k=k+1;
end
for i = 1:2:length(L_Saccade)
    start = L_Saccade(i);
    stop = L_Saccade(i+1);
    Max_sac_velL(l) = max(abs(vel_El(start:stop)));
    Sac_ampL(l) = abs(Lhorz_eye_angle(stop)-Lhorz_eye_angle(start));
    Error_L(l)= (mean(er_L(stop:stop+10)))*1000;
    l=l+1;
end
average_Error_L = mean(Error_L);
average_Error_R = mean(Error_R);
std_Error_L = std(Error_L);
std_Error_R = std(Error_R);

%-------------------------------------------------------------------------
----
%  Plot saccadic error
%-------------------------------------------------------------------------
----
if Z == 1
    sac_markL=zeros(20,1);
    sac_markR=zeros(20,1);
    flag1=0;flag2=0;
    j=1;k=1;
    for i = start_point+1: file_len
```

```
    if XRR(i)~=0 && flag1==0;
        sac_markR(j) = i;
        flag1=1;
        j=j+1;
    end
    if XRR(i)==0 && flag1 == 1
        sac_markR(j)= i-1;
        flag1 = 0;
        j = j+1;
    elseif i == file_len && flag1 == 1
        sac_markR(j) = i;
    end
 end
 for i = start_point+1: file_len
    if XLR(i)~=0 && flag2==0;
        sac_markL(k) = i;
        flag2=1;
        k=k+1;
    end
    if XLR(i)==0 && flag2 == 1
        sac_markL(k)= i-1;
        flag2 = 0;
        k = k+1;
    elseif i == file_len && flag2 == 1
        sac_markL(k) = i;
    end
 end
 j=1;k=1;
for i = 1:2:length(sac_markR)-1
    startR=sac_markR(i);
    stopR=sac_markR(i+1);
    figure (10)
    hold on
    title('Point of intercept (Blue) Vs Right Target(Red)')
    xlabel('Time');
    ylabel('Eye PoI/LED position');
    plot (frameR(startR:stopR)*sample_rate,yR(startR:stopR),'r')
    plot (frameR(startR:stopR)*sample_rate,XRR(startR:stopR),'b')
    a=num2str(startR);
    b=num2str(stopR);
    text(frameR(startR)*sample_rate+0.05,0,a);
    text(frameR(stopR)*sample_rate-0.05,0,b);
    pause
    clf
 end
 for i = 1:2:length(sac_markL)-1
    startL=sac_markL(i);
    stopL=sac_markL(i+1);
    figure (11)
    hold on
    title('Point of intercept (Blue) Vs Left Target(Red)')
    xlabel('Time');
    ylabel('Eye PoI/LED position');
    plot (frameL(startL:stopL)*sample_rate,yL(startL:stopL),'r')
```

```
        plot (frameL(startL:stopL)*sample_rate,XLR(startL:stopL),'b')
        a=num2str(startL);
        b=num2str(stopL);
        text(frameL(startL)*sample_rate+0.05,0.1,a);
        text(frameL(stopL)*sample_rate-0.05,0.1,b);
        pause
        clf
    end
end

figure(12)
hold on
title('Time Duration for Left(Green) & Right(Blue) Saccades Vs
Amplitude')
plot (Sac_ampL(1:18),durationL(1:18),'.g')
plot (Sac_ampR(1:18),durationR(1:18),'.b')
xlabel('Amplitude');
ylabel('Time Duration');

figure(13)
hold on
title('Peak Velocity for Left(Green) & Right(Blue) Saccades Vs
Amplitude')
plot (Sac_ampL(1:18),max_velL(1:18),'.g')
plot (Sac_ampR(1:18),max_velR(1:18),'.b')
xlabel('Amplitude');
ylabel('Peak Velocity');
```

APPENDIX D: MATLAB code for analyzing VOR data.

```
%-------------------------------------------------------------------------
----
%  This program is designed to plot the head angle vs eye angle for VOR
%-------------------------------------------------------------------------
----
close all
clear

filename=input('Select File to Run: ','s')
Y=load(filename);

frame=Y(:,1); head_angle=Y(:,2);
LE_X=Y(:,3);    LE_Y=Y(:,4);    LE_Z=Y(:,5);
RE_X=Y(:,6);    RE_Y=Y(:,7);    RE_Z=Y(:,8);

Lg_X=Y(:,9);   Lg_Y=Y(:,10);   Lg_Z=Y(:,11);
Rg_X=Y(:,12);   Rg_Y=Y(:,13);   Rg_Z=Y(:,14);

Lhorz_eye_angle=Y(:,15); Lvert_eye_angle=Y(:,16);
eye_angle=Y(:,17);

%-----------------------------------------
%  create arrays
%-----------------------------------------
file_len=length(head_angle);
zeroed_head_angle = zeros(file_len,1);
zeroed_eye_angle = zeros(file_len,1);
time = frame/250;
%-------------------------------------------------------------------------
----
%  Calculation of Mean
%-------------------------------------------------------------------------
----
for i = 1:file_len
    zeroed_head_angle(i) = head_angle(i)-mean(head_angle);
    zeroed_eye_angle(i) = eye_angle(i)-mean(eye_angle);
end
%-------------------------------------------------------------------------
----
%  Plot of Zeroed Eye and Head angles
%-------------------------------------------------------------------------
----
figure(1)
hold on
title('Zeroed Eye(green) and Head(blue) angles plot')
xlabel('Time (sec)');
ylabel('Head Angle/Eye Angle');
plot((frame/250), zeroed_head_angle(:,1), 'b')
plot((frame/250), zeroed_eye_angle(:,1), 'g')
```

```
%-------------------------------------------------------------------------
----
%  Calculation of Head Angle Vs Eye Angle
%-------------------------------------------------------------------------
----
figure(2)
hold on
title('Head Angle Vs Eye Angle')
plot (zeroed_head_angle, zeroed_eye_angle,'.g')
xlabel('Head Angle(degrees)');
ylabel('Eye Angle(degrees)');
p=polyfit(zeroed_head_angle, zeroed_eye_angle, 1);
t2= min(zeroed_head_angle):0.1:max(zeroed_head_angle);
y2=polyval(p,t2);
plot (t2,y2, 'k');
%-------------------------------------------------------------------------
----
%  Sorting of Left & Right Head & Eye Movements
%-------------------------------------------------------------------------
----

% left_head = zeros(file_len,1);
% right_head = zeros(file_len,1);
% left_eye = zeros(file_len,1);
% right_eye = zeros(file_len,1);

left_head = find (zeroed_head_angle > 0);
right_head = find (zeroed_head_angle < 0);

left_eye = find (zeroed_eye_angle < 0);
right_eye = find (zeroed_eye_angle > 0);
%-------------------------------------------------------------------------
----
%  Calculation of Head Velocity Vs Eye Velocity
%  Also create array noting left (-100) or right (+100) head movement
and
%  place these data in separate left and right arrays
%-------------------------------------------------------------------------
----
vel_H = zeros(file_len,1);
vel_E = zeros(file_len,1);

% directional variables
direction = zeros(file_len,1);
velT = 5;                   % set velocity threshold to 10 deg/sec
vel_Hr = zeros(file_len,1);
vel_Hl = zeros(file_len,1);
vel_Er = zeros(file_len,1);
vel_El = zeros(file_len,1);
Vel_HR = zeros(20,1);
Vel_HL = zeros(20,1);
Vel_ER = zeros(20,1);
Vel_EL = zeros(20,1);
```

```
R_gain = zeros(10,1);
L_gain = zeros(10,1);
l=1;m=1;

sample_freq = 250;
sample_rate = 1/sample_freq;
for i=2:file_len-1
    vel_H(i)=((zeroed_head_angle(i+1)-zeroed_head_angle(i-
1))/(2*sample_rate));
    vel_E(i)=((zeroed_eye_angle(i+1)-zeroed_eye_angle(i-
1))/(2*sample_rate));
    if (vel_H(i) > velT)
        direction(i) = 100;
        vel_Hr(i) = vel_H(i);
        vel_Er(i) = vel_E(i);
        Vel_HR(l) = vel_Hr(i);
        Vel_ER(l) = vel_Er(i);
        l=l+1;
    end
    if (vel_H(i) < -velT)
        direction(i) = -100;
        vel_Hl(i) = vel_H(i);
        vel_El(i) = vel_E(i);
        Vel_HL(m) = vel_Hl(i);
        Vel_EL(m) = vel_El(i);
        m=m+1;
    end
end

j=1;k=1;n=0;flag=0;
for i = 1:length(Vel_ER)
    R_gain(j)= Vel_ER(i)/Vel_HR(i);
    j=j+1;
end

for i = 1:length(Vel_EL)
    L_gain(k)= Vel_EL(i)/Vel_HL(i);
    k=k+1;
end
Gain_R = mean(R_gain);
Gain_L = mean(L_gain);
Gain = (Gain_R+Gain_L)/2;

Vel_HR10 = zeros(10,1);
Vel_HR20 = zeros(10,1);
Vel_HR30 = zeros(10,1);
Vel_HR40 = zeros(10,1);
Vel_HR50 = zeros(10,1);
Vel_HR60 = zeros(10,1);
Vel_HR70 = zeros(10,1);
Vel_HR80 = zeros(10,1);
Vel_HR90 = zeros(10,1);
Vel_HR100 = zeros(10,1);
```

```
Vel_HR110 = zeros(10,1);
Vel_HR120 = zeros(10,1);
Vel_HR130 = zeros(10,1);
Vel_HR140 = zeros(10,1);
Vel_HR150 = zeros(10,1);
Vel_HR160 = zeros(10,1);
Vel_HR170 = zeros(10,1);
Vel_HR180 = zeros(10,1);
Vel_HR190 = zeros(10,1);
Vel_HR200 = zeros(10,1);
Vel_HR_Hi = zeros(10,1);
Vel_ER10 = zeros(10,1);
Vel_ER20 = zeros(10,1);
Vel_ER30 = zeros(10,1);
Vel_ER40 = zeros(10,1);
Vel_ER50 = zeros(10,1);
Vel_ER60 = zeros(10,1);
Vel_ER70 = zeros(10,1);
Vel_ER80 = zeros(10,1);
Vel_ER90 = zeros(10,1);
Vel_ER100 = zeros(10,1);
Vel_ER110 = zeros(10,1);
Vel_ER120 = zeros(10,1);
Vel_ER130 = zeros(10,1);
Vel_ER140 = zeros(10,1);
Vel_ER150 = zeros(10,1);
Vel_ER160 = zeros(10,1);
Vel_ER170 = zeros(10,1);
Vel_ER180 = zeros(10,1);
Vel_ER190 = zeros(10,1);
Vel_ER200 = zeros(10,1);
Vel_ER_Hi = zeros(10,1);
Gain_R10 = zeros(10,1);
Gain_R20 = zeros(10,1);
Gain_R30 = zeros(10,1);
Gain_R40 = zeros(10,1);
Gain_R50 = zeros(10,1);
Gain_R60 = zeros(10,1);
Gain_R70 = zeros(10,1);
Gain_R80 = zeros(10,1);
Gain_R90 = zeros(10,1);
Gain_R100 = zeros(10,1);
Gain_R110 = zeros(10,1);
Gain_R120 = zeros(10,1);
Gain_R130 = zeros(10,1);
Gain_R140 = zeros(10,1);
Gain_R150 = zeros(10,1);
Gain_R160 = zeros(10,1);
Gain_R170 = zeros(10,1);
Gain_R180 = zeros(10,1);
Gain_R190 = zeros(10,1);
Gain_R200 = zeros(10,1);
Gain_R_Hi = zeros(10,1);
Vel_HL10 = zeros(10,1);
```

```
Vel_HL20 = zeros(10,1);
Vel_HL30 = zeros(10,1);
Vel_HL40 = zeros(10,1);
Vel_HL50 = zeros(10,1);
Vel_HL60 = zeros(10,1);
Vel_HL70 = zeros(10,1);
Vel_HL80 = zeros(10,1);
Vel_HL90 = zeros(10,1);
Vel_HL100 = zeros(10,1);
Vel_HL110 = zeros(10,1);
Vel_HL120 = zeros(10,1);
Vel_HL130 = zeros(10,1);
Vel_HL140 = zeros(10,1);
Vel_HL150 = zeros(10,1);
Vel_HL160 = zeros(10,1);
Vel_HL170 = zeros(10,1);
Vel_HL180 = zeros(10,1);
Vel_HL190 = zeros(10,1);
Vel_HL200 = zeros(10,1);
Vel_HL_Hi = zeros(10,1);
Vel_EL10 = zeros(10,1);
Vel_EL20 = zeros(10,1);
Vel_EL30 = zeros(10,1);
Vel_EL40 = zeros(10,1);
Vel_EL50 = zeros(10,1);
Vel_EL60 = zeros(10,1);
Vel_EL70 = zeros(10,1);
Vel_EL80 = zeros(10,1);
Vel_EL90 = zeros(10,1);
Vel_EL100 = zeros(10,1);
Vel_EL110 = zeros(10,1);
Vel_EL120 = zeros(10,1);
Vel_EL130 = zeros(10,1);
Vel_EL140 = zeros(10,1);
Vel_EL150 = zeros(10,1);
Vel_EL160 = zeros(10,1);
Vel_EL170 = zeros(10,1);
Vel_EL180 = zeros(10,1);
Vel_EL190 = zeros(10,1);
Vel_EL200 = zeros(10,1);
Vel_EL_Hi = zeros(10,1);
Gain_L10 = zeros(10,1);
Gain_L20 = zeros(10,1);
Gain_L30 = zeros(10,1);
Gain_L40 = zeros(10,1);
Gain_L50 = zeros(10,1);
Gain_L60 = zeros(10,1);
Gain_L70 = zeros(10,1);
Gain_L80 = zeros(10,1);
Gain_L90 = zeros(10,1);
Gain_L100 = zeros(10,1);
Gain_L110 = zeros(10,1);
Gain_L120 = zeros(10,1);
Gain_L130 = zeros(10,1);
```

```
Gain_L140 = zeros(10,1);
Gain_L150 = zeros(10,1);
Gain_L160 = zeros(10,1);
Gain_L170 = zeros(10,1);
Gain_L180 = zeros(10,1);
Gain_L190 = zeros(10,1);
Gain_L200 = zeros(10,1);
Gain_L_Hi = zeros(10,1);
j=1;k=1;l=1;m=1;n=1;o=1;p=1;q=1;r=1;s=1;t=1;u=1;v=1;w=1;x=1;y=1;z=1;
J=1;K=1;L=1;M=1;
 for i = 1:length(Vel_HR)
     if abs(Vel_HR(i)) < 10
         Vel_HR10(j)=Vel_HR(i);
         Vel_ER10(j)=Vel_ER(i);
         Gain_R10(j)=Vel_ER10(j)/Vel_HR10(j);
         j=j+1;
     end
     if abs(Vel_HR(i)) < 20 && abs(Vel_HR(i)) >10
         Vel_HR20(k)=Vel_HR(i);
         Vel_ER20(k)=Vel_ER(i);
         Gain_R20(k)=Vel_ER20(k)/Vel_HR20(k);
         k=k+1;
     end
     if abs(Vel_HR(i)) < 30 && abs(Vel_HR(i)) >20
         Vel_HR30(l)=Vel_HR(i);
         Vel_ER30(l)=Vel_ER(i);
         Gain_R30(l)=Vel_ER30(l)/Vel_HR30(l);
         l=l+1;
     end
     if abs(Vel_HR(i)) < 40 && abs(Vel_HR(i)) >30
         Vel_HR40(m)=Vel_HR(i);
         Vel_ER40(m)=Vel_ER(i);
         Gain_R40(m)=Vel_ER40(m)/Vel_HR40(m);
         m=m+1;
     end
      if abs(Vel_HR(i)) < 50 && abs(Vel_HR(i)) >40
         Vel_HR50(n)=Vel_HR(i);
         Vel_ER50(n)=Vel_ER(i);
         Gain_R50(n)=Vel_ER50(n)/Vel_HR50(n);
         n=n+1;
      end
      if abs(Vel_HR(i)) < 60 && abs(Vel_HR(i)) >50
         Vel_HR60(o)=Vel_HR(i);
         Vel_ER60(o)=Vel_ER(i);
         Gain_R60(o)=Vel_ER60(o)/Vel_HR60(o);
         o=o+1;
      end
      if abs(Vel_HR(i)) < 70 && abs(Vel_HR(i)) >60
         Vel_HR70(p)=Vel_HR(i);
         Vel_ER70(p)=Vel_ER(i);
         Gain_R70(p)=Vel_ER70(p)/Vel_HR70(p);
         p=p+1;
      end
      if abs(Vel_HR(i)) < 80 && abs(Vel_HR(i)) >70
```

```
    Vel_HR80(q)=Vel_HR(i);
    Vel_ER80(q)=Vel_ER(i);
    Gain_R80(q)=Vel_ER80(q)/Vel_HR80(q);
    q=q+1;
end
if abs(Vel_HR(i)) < 90 && abs(Vel_HR(i)) >80
    Vel_HR90(r)=Vel_HR(i);
    Vel_ER90(r)=Vel_ER(i);
    Gain_R90(r)=Vel_ER90(r)/Vel_HR90(r);
    r=r+1;
end
if abs(Vel_HR(i)) < 100 && abs(Vel_HR(i)) >90
    Vel_HR100(s)=Vel_HR(i);
    Vel_ER100(s)=Vel_ER(i);
    Gain_R100(s)=Vel_ER100(s)/Vel_HR100(s);
    s=s+1;
end
if abs(Vel_HR(i)) < 110 && abs(Vel_HR(i)) >100
    Vel_HR110(t)=Vel_HR(i);
    Vel_ER110(t)=Vel_ER(i);
    Gain_R110(t)=Vel_ER110(t)/Vel_HR110(t);
    t=t+1;
end
if abs(Vel_HR(i)) < 120 && abs(Vel_HR(i)) >110
    Vel_HR120(u)=Vel_HR(i);
    Vel_ER120(u)=Vel_ER(i);
    Gain_R120(u)=Vel_ER120(u)/Vel_HR120(u);
    u=u+1;
end
if abs(Vel_HR(i)) < 130 && abs(Vel_HR(i)) >120
    Vel_HR130(v)=Vel_HR(i);
    Vel_ER130(v)=Vel_ER(i);
    Gain_R130(v)=Vel_ER130(v)/Vel_HR130(v);
    v=v+1;
end
if abs(Vel_HR(i)) < 140 && abs(Vel_HR(i)) >130
    Vel_HR140(w)=Vel_HR(i);
    Vel_ER140(w)=Vel_ER(i);
    Gain_R140(w)=Vel_ER140(w)/Vel_HR140(w);
    w=w+1;
end
if abs(Vel_HR(i)) < 150 && abs(Vel_HR(i)) >140
    Vel_HR150(x)=Vel_HR(i);
    Vel_ER150(x)=Vel_ER(i);
    Gain_R150(x)=Vel_ER150(x)/Vel_HR150(x);
    x=x+1;
end
if abs(Vel_HR(i)) < 160 && abs(Vel_HR(i)) >150
    Vel_HR160(y)=Vel_HR(i);
    Vel_ER160(y)=Vel_ER(i);
    Gain_R160(y)=Vel_ER160(y)/Vel_HR160(y);
    y=y+1;
end
if abs(Vel_HR(i)) < 170 && abs(Vel_HR(i)) >160
```

```
         Vel_HR170(z)=Vel_HR(i);
         Vel_ER170(z)=Vel_ER(i);
         Gain_R170(z)=Vel_ER170(z)/Vel_HR170(z);
         z=z+1;
      end
      if abs(Vel_HR(i)) < 180 && abs(Vel_HR(i)) >170
         Vel_HR180(M)=Vel_HR(i);
         Vel_ER180(M)=Vel_ER(i);
         Gain_R180(M)=Vel_ER180(M)/Vel_HR180(M);
         M=M+1;
      end
      if abs(Vel_HR(i)) < 190 && abs(Vel_HR(i)) >180
         Vel_HR190(J)=Vel_HR(i);
         Vel_ER190(J)=Vel_ER(i);
         Gain_R190(J)=Vel_ER190(J)/Vel_HR190(J);
         J=J+1;
      end
      if abs(Vel_HR(i)) < 200 && abs(Vel_HR(i)) >190
         Vel_HR200(K)=Vel_HR(i);
         Vel_ER200(K)=Vel_ER(i);
         Gain_R200(K)=Vel_ER200(K)/Vel_HR200(K);
         K=K+1;
      end
      if 200 < abs(Vel_HR(i))
         Vel_HR_Hi(L)=Vel_HR(i);
         Vel_ER_Hi(L)=Vel_ER(i);
         Gain_R_Hi(L)=Vel_ER_Hi(L)/Vel_HR_Hi(L);
         L=L+1;
      end
 end
Avg_Vel_HR=[mean(Vel_HR10);mean(Vel_HR20);mean(Vel_HR30);mean(Vel_HR40)
;mean(Vel_HR50);

mean(Vel_HR60);mean(Vel_HR70);mean(Vel_HR80);mean(Vel_HR90);mean(Vel_HR
100);

mean(Vel_HR110);mean(Vel_HR120);mean(Vel_HR130);mean(Vel_HR140);mean(Ve
l_HR150);

mean(Vel_HR160);mean(Vel_HR170);mean(Vel_HR180);mean(Vel_HR190);mean(Ve
l_HR200);mean(Vel_HR_Hi)];
Avg_Gain_R=[mean(Gain_R10);mean(Gain_R20);mean(Gain_R30);mean(Gain_R40)
;mean(Gain_R50);

mean(Gain_R60);mean(Gain_R70);mean(Gain_R80);mean(Gain_R90);mean(Gain_R
100);mean(Gain_R110);

mean(Gain_R120);mean(Gain_R130);mean(Gain_R140);mean(Gain_R150);mean(Ga
in_R160);mean(Gain_R170);
    mean(Gain_R180);mean(Gain_R190);mean(Gain_R200);mean(Gain_R_Hi)];

j=1;k=1;l=1;m=1;n=1;o=1;p=1;q=1;r=1;s=1;t=1;u=1;v=1;w=1;x=1;y=1;z=1;
J=1;K=1;L=1;M=1;
```

```
for i = 1:length(Vel_HL)
    if abs(Vel_HL(i)) < 10
        Vel_HL10(j)=Vel_HL(i);
        Vel_EL10(j)=Vel_EL(i);
        Gain_L10(j)=Vel_EL10(j)/Vel_HL10(j);
        j=j+1;
    end
    if abs(Vel_HL(i)) < 20 && abs(Vel_HL(i)) >10
        Vel_HL20(k)=Vel_HL(i);
        Vel_EL20(k)=Vel_EL(i);
        Gain_L20(k)=Vel_EL20(k)/Vel_HL20(k);
        k=k+1;
    end
    if abs(Vel_HL(i)) < 30 && abs(Vel_HL(i)) >20
        Vel_HL30(l)=Vel_HL(i);
        Vel_EL30(l)=Vel_EL(i);
        Gain_L30(l)=Vel_EL30(l)/Vel_HL30(l);
        l=l+1;
    end
    if abs(Vel_HL(i)) < 40 && abs(Vel_HL(i)) >30
        Vel_HL40(m)=Vel_HL(i);
        Vel_EL40(m)=Vel_EL(i);
        Gain_L40(m)=Vel_EL40(m)/Vel_HL40(m);
        m=m+1;
    end
     if abs(Vel_HL(i)) < 50 && abs(Vel_HL(i)) >40
        Vel_HL50(n)=Vel_HL(i);
        Vel_EL50(n)=Vel_EL(i);
        Gain_L50(n)=Vel_EL50(n)/Vel_HL50(n);
        n=n+1;
     end
     if abs(Vel_HL(i)) < 60 && abs(Vel_HL(i)) >50
        Vel_HL60(o)=Vel_HL(i);
        Vel_EL60(o)=Vel_EL(i);
        Gain_L60(o)=Vel_EL60(o)/Vel_HL60(o);
        o=o+1;
     end
     if abs(Vel_HL(i)) < 70 && abs(Vel_HL(i)) >60
        Vel_HL70(p)=Vel_HL(i);
        Vel_EL70(p)=Vel_EL(i);
        Gain_L70(p)=Vel_EL70(p)/Vel_HL70(p);
        p=p+1;
     end
     if abs(Vel_HL(i)) < 80 && abs(Vel_HL(i)) >70
        Vel_HL80(q)=Vel_HL(i);
        Vel_EL80(q)=Vel_EL(i);
        Gain_L80(q)=Vel_EL80(q)/Vel_HL80(q);
        q=q+1;
     end
     if abs(Vel_HL(i)) < 90 && abs(Vel_HL(i)) >80
        Vel_HL90(r)=Vel_HL(i);
        Vel_EL90(r)=Vel_EL(i);
        Gain_L90(r)=Vel_EL90(r)/Vel_HL90(r);
        L=L+1;
```

```
end
if abs(Vel_HL(i)) < 100 && abs(Vel_HL(i)) >90
   Vel_HL100(s)=Vel_HL(i);
   Vel_EL100(s)=Vel_EL(i);
   Gain_L100(s)=Vel_EL100(s)/Vel_HL100(s);
   s=s+1;
end
if abs(Vel_HL(i)) < 110 && abs(Vel_HL(i)) >100
   Vel_HL110(t)=Vel_HL(i);
   Vel_EL110(t)=Vel_EL(i);
   Gain_L110(t)=Vel_EL110(t)/Vel_HL110(t);
   t=t+1;
end
if abs(Vel_HL(i)) < 120 && abs(Vel_HL(i)) >110
   Vel_HL120(u)=Vel_HL(i);
   Vel_EL120(u)=Vel_EL(i);
   Gain_L120(u)=Vel_EL120(u)/Vel_HL120(u);
   u=u+1;
end
if abs(Vel_HL(i)) < 130 && abs(Vel_HL(i)) >120
   Vel_HL130(v)=Vel_HL(i);
   Vel_EL130(v)=Vel_EL(i);
   Gain_L130(v)=Vel_EL130(v)/Vel_HL130(v);
   v=v+1;
end
if abs(Vel_HL(i)) < 140 && abs(Vel_HL(i)) >130
   Vel_HL140(w)=Vel_HL(i);
   Vel_EL140(w)=Vel_EL(i);
   Gain_L140(w)=Vel_EL140(w)/Vel_HL140(w);
   w=w+1;
end
if abs(Vel_HL(i)) < 150 && abs(Vel_HL(i)) >140
   Vel_HL150(x)=Vel_HL(i);
   Vel_EL150(x)=Vel_EL(i);
   Gain_L150(x)=Vel_EL150(x)/Vel_HL150(x);
   x=x+1;
end
if abs(Vel_HL(i)) < 160 && abs(Vel_HL(i)) >150
   Vel_HL160(y)=Vel_HL(i);
   Vel_EL160(y)=Vel_EL(i);
   Gain_L160(y)=Vel_EL160(y)/Vel_HL160(y);
   y=y+1;
end
if abs(Vel_HL(i)) < 170 && abs(Vel_HL(i)) >160
   Vel_HL170(z)=Vel_HL(i);
   Vel_EL170(z)=Vel_EL(i);
   Gain_L170(z)=Vel_EL170(z)/Vel_HL170(z);
   z=z+1;
end
if abs(Vel_HL(i)) < 180 && abs(Vel_HL(i)) >170
   Vel_HL180(M)=Vel_HL(i);
   Vel_EL180(M)=Vel_EL(i);
   Gain_L180(M)=Vel_EL180(M)/Vel_HL180(M);
   M=M+1;
```

```
      end
      if abs(Vel_HL(i)) < 190 && abs(Vel_HL(i)) >180
         Vel_HL190(J)=Vel_HL(i);
         Vel_EL190(J)=Vel_EL(i);
         Gain_L190(J)=Vel_EL190(J)/Vel_HL190(J);
         J=J+1;
      end
      if abs(Vel_HL(i)) < 200 && abs(Vel_HL(i)) >190
         Vel_HL200(K)=Vel_HL(i);
         Vel_EL200(K)=Vel_EL(i);
         Gain_L200(K)=Vel_EL200(K)/Vel_HL200(K);
         K=K+1;
      end
      if 200 < abs(Vel_HL(i))
         Vel_HL_Hi(L)=Vel_HL(i);
         Vel_EL_Hi(L)=Vel_EL(i);
         Gain_L_Hi(L)=Vel_EL_Hi(L)/Vel_HL_Hi(L);
         L=L+1;
      end
 end
Avg_Vel_HL=[mean(Vel_HL10);mean(Vel_HL20);mean(Vel_HL30);mean(Vel_HL40)
;mean(Vel_HL50);

mean(Vel_HL60);mean(Vel_HL70);mean(Vel_HL80);mean(Vel_HL90);mean(Vel_HL
100);

mean(Vel_HL110);mean(Vel_HL120);mean(Vel_HL130);mean(Vel_HL140);mean(Ve
l_HL150);

mean(Vel_HL160);mean(Vel_HL170);mean(Vel_HL180);mean(Vel_HL190);mean(Ve
l_HL200);mean(Vel_HL_Hi)];
Avg_Gain_L=[mean(Gain_L10);mean(Gain_L20);mean(Gain_L30);mean(Gain_L40)
;mean(Gain_L50);

mean(Gain_L60);mean(Gain_L70);mean(Gain_L80);mean(Gain_L90);mean(Gain_L
100);mean(Gain_L110);

mean(Gain_L120);mean(Gain_L130);mean(Gain_L140);mean(Gain_L150);mean(Ga
in_L160);mean(Gain_L170);
    mean(Gain_L180);mean(Gain_L190);mean(Gain_L200);mean(Gain_L_Hi)];

figure(3)
hold on
title('Head Velocity(Blue), Eye Velocity(Green),Velocity Direction
(Red)')
plot (vel_H(:,1),'b')
plot (vel_E(:,1),'g')
plot (direction(:,1),'r')   % plot velocity direction array to test

figure(4)
hold on
title('Head Velocity Vs Eye Velocity')
plot (vel_H, vel_E,'k.')
```

```
xlabel('Head Velocity(degrees/sec)');
ylabel('Eye Velocity(degrees/sec)');
axis([-100,100,-300,300]);

X = xlim;
Y = ylim;

%-------------------------------------------------------------------------
----
%  Determine Left vs Right VOR gains
%-------------------------------------------------------------------------
----
figure(5)
hold on
title('Head Velocity Vs Eye Velocity (RIGHT(red) vs LEFT(black))')

axis([-200,200,-200,200]);   % set axis limits to previous plot
plot (vel_Hr, vel_Er,'k.')
xlabel('Head Velocity(degrees/sec)');
ylabel('Eye Velocity(degrees/sec)');
    p=polyfit(vel_Hr, vel_Er, 1);
    t2= min(vel_Hr):0.1:max(vel_Hr);
    y2=polyval(p,t2);
    plot (t2,y2, 'r');
    [R,P] = corrcoef(vel_Hr, vel_Er);
%R(2)
%P(2)
p(1)   %gain

%figure(7)
%hold on
%title('Head Velocity Vs Eye Velocity (LEFT)')
plot (vel_Hl, vel_El,'r.')
xlabel('Head Velocity(degrees/sec)');
ylabel('Eye Velocity(degrees/sec)');
    p=polyfit(vel_Hl, vel_El, 1);
    t2= min(vel_Hl):0.1:max(vel_Hl);
    y2=polyval(p,t2);
    plot (t2,y2, 'k');
    [R,P] = corrcoef(vel_Hl, vel_El);
p(1)   %gain



%-------------------------------------------------------------------------
----
%   PoG & PoI Calculations
%-------------------------------------------------------------------------
--

pog_X=zeros(file_len,1);
pog_Y=zeros(file_len,1);
```

```
pog_Z=zeros(file_len,1);
X_error=zeros(file_len,1);
Y_error=zeros(file_len,1);
Z_error=zeros(file_len,1);

XYZ_error=zeros(file_len,1);
LED_X = 0.364;
LED_Y = 1;
LED_Z = -0.409;
%compute eye point-of-gaze within stationary eye-position segments
for j = 1:file_len
    ml = Lg_X(j)/Lg_Y(j);% Calculating m1 in excel file
    mr = Rg_X(j)/Rg_Y(j);% Calculating m2 in excel file
    bl = abs(LE_Y(j))* ml + LE_X(j);% Calculating b1 in excel file
    br = abs(RE_Y(j))* mr + RE_X(j);% Calculating b2 in excel file
    pog_Y(j) = (br - bl)/(ml - mr);% Calculating Y in excel file
    pog_X(j) = ml*pog_Y(j) + bl;% Calculating X in excel file

    lgQ = atan(Lg_Z(j)/Lg_Y(j))*180/pi;% Calculating Vertical(left)
gaze angle in excel file
    rgQ = atan(Rg_Z(j)/Rg_Y(j))*180/pi;% Calculating Vertical(right)
gaze angle in excel file
    lvo = (pog_Y(j)- LE_Y(j))*tan(lgQ*pi/180);% Calculating
Vertical(left) offset in excel file
    rvo = (pog_Y(j)- RE_Y(j))*tan(rgQ*pi/180);% Calculating
Vertical(right) offset in excel file
    lvi = lvo+ LE_Z(j);% Calculating Vertical(left) intercept in excel
file
    rvi = rvo+ RE_Z(j);% Calculating Vertical(right) intercept in excel
file
    pog_Z(j) = (lvi+ rvi)/2;% Calculating Z in excel file
 %compute error
        X_error(j) = pog_X(j) - LED_X;
        Y_error(j) = pog_Y(j) - LED_Y;
        Z_error(j) = pog_Z(j) - LED_Z;
        XYZ_error(j) = sqrt(X_error(j)^2+Y_error(j)^2+Z_error(j)^2);
end




%-------------------------------------------------------------------------
----
%  Compute intercept (2D) error
%-------------------------------------------------------------------------
----
X_cyc=zeros(file_len,1);
Y_cyc=zeros(file_len,1);
Z_cyc=zeros(file_len,1);
X_cal=zeros(file_len,1);
Z_cal=zeros(file_len,1);
Int_err_X=zeros(file_len,1);
Int_err_Z=zeros(file_len,1);
```

```
XZ_error=zeros(file_len,1);
X_left=zeros(file_len,1);
Z_left=zeros(file_len,1);
X_right=zeros(file_len,1);
Z_right=zeros(file_len,1);
L_Int_err_X=zeros(file_len,1);
R_Int_err_X=zeros(file_len,1);
L_Int_err_Z=zeros(file_len,1);
R_Int_err_Z=zeros(file_len,1);
L_XZ_error=zeros(file_len,1);
R_XZ_error=zeros(file_len,1);
X_fov=zeros(file_len,1);
Z_fov=zeros(file_len,1);

for j = 1:file_len
    % Calculating Cyclopian Eye;
    X_cyc(j)= (LE_X(j)+ RE_X(j))/2;
    Z_cyc(j)= (LE_Z(j)+ RE_Z(j))/2;
    Y_cyc(j)= (LE_Y(j)+ RE_Y(j))/2;

    % Calculating Intercept error (based on target depth);
    r = (LED_Y - Y_cyc(j))/(pog_Y(j) - Y_cyc(j));
    X_cal(j)= X_cyc(j) + r*(pog_X(j)-X_cyc(j));% - offsetx[vor=;vorc=]
    Z_cal(j)= Z_cyc(j) + r*(pog_Z(j)-Z_cyc(j));% - offsetz[vor=;vorc=]]
    X_left(j)= LE_X(j)+ r*(pog_X(j)- LE_X(j)) ;
    X_right(j)= RE_X(j)+ r*(pog_X(j)- RE_X(j));
    Z_left(j) = LE_Z(j) + r*(pog_Z(j)-LE_Z(j));
    Z_right(j) = RE_Z(j) + r*(pog_Z(j)-RE_Z(j));
    Int_err_X(j) = X_cal(j)-LED_X;
    Int_err_Z(j) = Z_cal(j)-LED_Z;
    L_Int_err_X(j) = X_left(j)-LED_X;
    R_Int_err_X(j) = X_right(j)-LED_X;
    L_Int_err_Z(j) = Z_left(j)-LED_Z;
    R_Int_err_Z(j) = Z_right(j)-LED_Z;

    XZ_error(j) = sqrt(Int_err_X(j)^2+Int_err_Z(j)^2);
    L_XZ_error(j) = sqrt(L_Int_err_X(j)^2+L_Int_err_Z(j)^2);
    R_XZ_error(j) = sqrt(R_Int_err_X(j)^2+R_Int_err_Z(j)^2);

end
%-------------------------------------------------------------------------
----
%Eliminating NaNs
%-------------------------------------------------------------------------
----
XZ_err=zeros(10,1);
check=zeros(10,1);
for j=1:length(XZ_error)
    check(j) = isfinite(XZ_error(j));
end
i=1;
for j=1:length(check)
    if check(j) == 1
```

```
        XZ_err(i) = XZ_error(j);
        i=i+1;
    end
end

std_XZ_error = std(XZ_err);
avg_XZ_error = mean(XZ_err);
std_zeroed_head_angle = std(zeroed_head_angle);

%-------------------------------------------------------------------------
----
%  Target Foveation
%-------------------------------------------------------------------------
----
for j =1:file_len
   if atan(sqrt((X_cal(j)-LED_X)^2))*180/pi < 3.26   % in degrees =
1(fovea)+0.03(target)+0.5(eyelink)+0.1(motion monitor)...double it 2
count for errors during head movement.
       X_fov(j) = X_cal(j);
       Z_fov(j) = Z_cal(j);
   end
end
f = 0;
for j = 1:file_len
   if X_fov(j)~=0
       f = f +1;
   end
c = length(X_cal);

end
FF = f/c*100;
figure (6)
hold on
title('Scatter plot (Target vs. point-of-gaze)')
xlabel('X co-ordinate');
ylabel('-Z co-ordinate');
plot(X_cal,-Z_cal,'.b')
plot(X_fov,-Z_fov,'.g')
plot(LED_X,-LED_Z,'--rs','LineWidth',2, 'MarkerSize' ,5)
%-------------------------------------------------------------------------
----
%  Final Temporal Plot of both Velocities
%-------------------------------------------------------------------------
----
a =400; %1)925;2)325;3)350;4)350 ;5)1100;6)550;7)400
z =1650; %1)3100;2)1900;3)1900;4)1900;5)2900;6)1650;7)1650
flag=-1;
slope = diff(head_angle)./diff(time);
for i =a:z
    if slope(i)<0 && flag==-1
        if max(slope(i:i+55))<0
            left_start1=i;
            left_end1 = i+55;
```

```
        flag=1;
    end
end
if slope(i)>0 && flag==1
    if min(slope(i:i+55))>0
        right_start1=i;
        right_end1 = i+55;
        flag=-2;
    end
end
if slope(i)<0 && flag==-2
    if max(slope(i:i+55))<0
        left_start2=i;
        left_end2 = i+55;
        flag=2;
    end
end
if slope(i)>0 && flag==2
    if min(slope(i:i+55))>0
        right_start2=i;
        right_end2 = i+55;
        flag=-3;
    end
end
if slope(i)<0 && flag==-3
    if max(slope(i:i+55))<0
        left_start3=i;
        left_end3 = i+55;
        flag=3;
    end
end
if slope(i)>0 && flag==3
    if min(slope(i:i+55))>0
        right_start3=i;
        right_end3 = i+55;
        flag=-4;
    end
end
if slope(i)<0 && flag==-4
    if max(slope(i:i+55))<0
        left_start4=i;
        left_end4 = i+55;
        flag=4;
    end
end
if slope(i)>0 && flag==4
    if min(slope(i:i+55))>0
        right_start4=i;
        right_end4 = i+55;
        flag=-5;
    end
end
if slope(i)<0 && flag==-5
    if max(slope(i:i+55))<0
```

```
            left_start5=i;
            left_end5 = i+55;
            flag=5;
        end
    end
    if slope(i)>0 && flag==5
        if min(slope(i:i+55))>0
            right_start5=i;
            right_end5 = i+55;
            flag=-6;
        end
    end
    if slope(i)<0 && flag==-6
        if max(slope(i:i+55))<0
            left_start6=i;
            left_end6 = i+55;
        end
    end
end
A=vel_H(left_start1:left_end1);
B=vel_H(left_start2:left_end2);
C=vel_H(left_start3:left_end3);
D=vel_H(left_start4:left_end4);
E=vel_H(left_start5:left_end5);
Hl=[A';B';C';D';E'];
lh_vel=mean(Hl);
lh_vel=lh_vel';
F=vel_H(right_start1:right_end1);
G=vel_H(right_start2:right_end2);
H=vel_H(right_start3:right_end3);
I=vel_H(right_start4:right_end4);
J=vel_H(right_start5:right_end5);
Hr=[F';G';H';I';J'];
rh_vel=mean(Hr);
rh_vel=rh_vel';

K=vel_E(left_start1:left_end1);
L=vel_E(left_start2:left_end2);
M=vel_E(left_start3:left_end3);
N=vel_E(left_start4:left_end4);
O=vel_E(left_start5:left_end5);
El=[K';L';M';N';O'];
le_vel=mean(El);
le_vel=le_vel';
P=vel_E(right_start1:right_end1);
Q=vel_E(right_start2:right_end2);
R=vel_E(right_start3:right_end3);
S=vel_E(right_start4:right_end4);
T=vel_E(right_start5:right_end5);
Er=[P';Q';R';S';T'];
re_vel=mean(Er);
re_vel=re_vel';
```

```
figure (7)
hold on
title('Leftwards Velocity plot (Head(blue) and eye(green))')
xlabel('Time');
ylabel('Velocities');
plot(time(1:56),lh_vel,'b')
plot(time(1:56),le_vel,'g')
axis([0,0.28,-200,250]);

figure (8)
hold on
title('Rightwards Velocity plot (Head(blue) and eye(green))')
xlabel('Time');
ylabel('Velocities');
plot(time(1:56),rh_vel,'b')
plot(time(1:56),re_vel,'g')
axis([0,0.28,-200,250]);
%-------------------------------------------------------------------------
----
%  10 deg/s range head velocity sorting
%-------------------------------------------------------------------------
----
% rvel_H=round(vel_H);
peak_A=min(vel_H(left_start1:left_end1));
% mode_A=mode(vel_H(left_start1:left_end1));
mean_A=mean(vel_H(left_end1-25:left_end1));
peak_F=max(vel_H(right_start1:right_end1));
% mode_F=mode(vel_H(right_start1:right_end1));
mean_F=mean(vel_H(right_end1-25:right_end1));


peak_B=min(vel_H(left_start2:left_end2));
mean_B=mean(vel_H(left_end2-25:left_end2));
% mode_B=mode(vel_H(left_start2:left_end2));
peak_G=max(vel_H(right_start2:right_end2));
mean_G=mean(vel_H(right_end2-25:right_end2));
% mode_G=mode(vel_H(right_start2:right_end2));

peak_C=min(vel_H(left_start3:left_end3));
mean_C=mean(vel_H(left_end3-25:left_end3));
peak_H=max(vel_H(right_start3:right_end3));
mean_H=mean(vel_H(right_end3-25:right_end3));
peak_D=min(vel_H(left_start4:left_end4));
mean_D=mean(vel_H(left_end4-25:left_end4));
peak_I=max(vel_H(right_start4:right_end4));
mean_I=mean(vel_H(right_end4-25:right_end4));

peak_E=min(vel_H(left_start5:left_end5));
mean_E=mean(vel_H(left_end5-25:left_end5));
peak_J=max(vel_H(right_start5:right_end5));
mean_J=mean(vel_H(right_end5-25:right_end5));
Mean_Left_Velocities = [mean_A;mean_B;mean_C;mean_D;mean_E];
Mean_Right_Velocities = [mean_F;mean_G;mean_H;mean_I;mean_J];
```

```
Wave_Data_H = [A,B,C,D,E,F,G,H,I,J];
Wave_Data_E = [K,L,M,N,O,P,Q,R,S,T];
```

APPENDIX E: MATLAB code for analyzing VORc data.

```
%-------------------------------------------------------------------------
----
%  This program is designed to plot the head angle vs eye angle for VOR
%-------------------------------------------------------------------------
----
close all
clear

filename=input('Select File to Run: ','s');
Y=load(filename);

frames=Y(:,1);head_angle=Y(:,2);
LE_X=Y(:,3);    LE_Y=Y(:,4);    LE_Z=Y(:,5);
RE_X=Y(:,6);    RE_Y=Y(:,7);    RE_Z=Y(:,8);

Lg_X=Y(:,9);   Lg_Y=Y(:,10);   Lg_Z=Y(:,11);
Rg_X=Y(:,12);   Rg_Y=Y(:,13);   Rg_Z=Y(:,14);

Lhorz_eye_angle=Y(:,15); Lvert_eye_angle=Y(:,16);
eye_angle=Y(:,17);eye_angleR=Y(:,18);

%-----------------------------------------
%  create arrays
%-----------------------------------------
file_len=length(head_angle);
zeroed_head_angle = zeros(file_len,1);
zeroed_eye_angle = zeros(file_len,1);
zeroed_eye_angleR = zeros(file_len,1);
pos_head = zeros(20,1);
pos_eye = zeros(20,1);
neg_head = zeros(20,1);
neg_eye = zeros(20,1);
m = zeros(file_len,1);
time = frames/250;
%-------------------------------------------------------------------------
----
%  Calculation of Mean
%-------------------------------------------------------------------------
----
for i = 1:file_len
    zeroed_head_angle(i) = head_angle(i)-mean(head_angle);
    zeroed_eye_angle(i) = eye_angle(i)-mean(eye_angle);
    zeroed_eye_angleR(i) = eye_angleR(i)-mean(eye_angleR);
end
%-------------------------------------------------------------------------
----
%  Plot of Zeroed Eye and Head angles
```

```
%-------------------------------------------------------------------------
----
figure(1)
hold on
title('Zeroed Left Eye(green),Right Eye(blue) and Head(red) angles
plot')
xlabel('Frames');
ylabel('Head Angle/Eye Angle');
plot(zeroed_head_angle(:,1), 'r')
plot(zeroed_eye_angle(:,1), 'g')
plot(zeroed_eye_angleR(:,1), 'b')

%-------------------------------------------------------------------------
----
%  Calculation of Head Angle Vs Eye Angle
%-------------------------------------------------------------------------
----
figure(2)
hold on
title('Head Angle Vs Eye Angle')
plot (zeroed_head_angle, zeroed_eye_angle,'.g')
xlabel('Head Angle(degrees)');
ylabel('Eye Angle(degrees)');
%-------------------------------------------------------------------------
----
%  Differentiation between positive and negative head angles
%-------------------------------------------------------------------------
----
j = 1;
k = 1;
for i = 1:file_len-1
    m = diff(zeroed_head_angle)./diff(frames);
    if m(i)>0
       pos_head(j) = zeroed_head_angle(i);
       pos_eye(j) = zeroed_eye_angle(i);
       j = j+1;
    end
    if m(i)<0
       neg_head(k) = zeroed_head_angle(i);
       neg_eye(k) = zeroed_eye_angle(i);
       k = k+1;
    end
end
figure(3)
hold on
title('Head Angle Vs Eye Angle left(green),right(blue)')
plot (pos_head(1), pos_eye(1),'rs','LineWidth',2,'MarkerSize' ,5)
plot (pos_head(1:300), pos_eye(1:300),'.b')
plot (neg_head(1:300), neg_eye(1:300),'.g')
xlabel('Head Angle(degrees)');
ylabel('Eye Angle(degrees)');
```

```
%-------------------------------------------------------------------------
----
%  Calculation of Head Velocity Vs Eye Velocity
%  Also create array noting left (-100) or right (+100) head movement
and
%  place these data in separate left and right arrays
%-------------------------------------------------------------------------
----
vel_H = zeros(file_len,1);
vel_E = zeros(file_len,1);

% directional variables
direction = zeros(file_len,1);
velT = 5;                    % set velocity threshold to 10 deg/sec
vel_Hr = zeros(file_len,1);
vel_Hl = zeros(file_len,1);
vel_Er = zeros(file_len,1);
vel_El = zeros(file_len,1);
Vel_HR = zeros(20,1);
Vel_HL = zeros(20,1);
Vel_ER = zeros(20,1);
Vel_EL = zeros(20,1);
R_gain = zeros(10,1);
L_gain = zeros(10,1);
l=1;m=1;

sample_freq = 250;
sample_rate = 1/sample_freq;
for i=2:file_len-1
    vel_H(i)=((zeroed_head_angle(i+1)-zeroed_head_angle(i-
1))/(2*sample_rate));
    vel_E(i)=((zeroed_eye_angle(i+1)-zeroed_eye_angle(i-
1))/(2*sample_rate));
    if (vel_H(i) > velT)
        direction(i) = 100;
        vel_Hr(i) = vel_H(i);
        vel_Er(i) = vel_E(i);
        Vel_HR(l) = vel_Hr(i);
        Vel_ER(l) = vel_Er(i);
        l=l+1;
    end
    if (vel_H(i) < -velT)
        direction(i) = -100;
        vel_Hl(i) = vel_H(i);
        vel_El(i) = vel_E(i);
        Vel_HL(m) = vel_Hl(i);
        Vel_EL(m) = vel_El(i);
        m=m+1;
    end
end
j=1;k=1;n=0;flag=0;
```

```
for i = 1:length(Vel_ER)
    R_gain(j)= Vel_ER(i)/Vel_HR(i);
    j=j+1;
end

for i = 1:length(Vel_EL)
    L_gain(k)= Vel_EL(i)/Vel_HL(i);
    k=k+1;
end
Gain_R = mean(R_gain);
Gain_L = mean(L_gain);
Gain = (Gain_R+Gain_L)/2;


Vel_HR10 = zeros(10,1);
Vel_HR20 = zeros(10,1);
Vel_HR30 = zeros(10,1);
Vel_HR40 = zeros(10,1);
Vel_HR50 = zeros(10,1);
Vel_HR60 = zeros(10,1);
Vel_HR70 = zeros(10,1);
Vel_HR80 = zeros(10,1);
Vel_HR90 = zeros(10,1);
Vel_HR100 = zeros(10,1);
Vel_HR110 = zeros(10,1);
Vel_HR120 = zeros(10,1);
Vel_HR130 = zeros(10,1);
Vel_HR140 = zeros(10,1);
Vel_HR150 = zeros(10,1);
Vel_HR160 = zeros(10,1);
Vel_HR170 = zeros(10,1);
Vel_HR180 = zeros(10,1);
Vel_HR190 = zeros(10,1);
Vel_HR200 = zeros(10,1);
Vel_HR_Hi = zeros(10,1);
Vel_ER10 = zeros(10,1);
Vel_ER20 = zeros(10,1);
Vel_ER30 = zeros(10,1);
Vel_ER40 = zeros(10,1);
Vel_ER50 = zeros(10,1);
Vel_ER60 = zeros(10,1);
Vel_ER70 = zeros(10,1);
Vel_ER80 = zeros(10,1);
Vel_ER90 = zeros(10,1);
Vel_ER100 = zeros(10,1);
Vel_ER110 = zeros(10,1);
Vel_ER120 = zeros(10,1);
Vel_ER130 = zeros(10,1);
Vel_ER140 = zeros(10,1);
Vel_ER150 = zeros(10,1);
Vel_ER160 = zeros(10,1);
Vel_ER170 = zeros(10,1);
Vel_ER180 = zeros(10,1);
```

```
Vel_ER190 = zeros(10,1);
Vel_ER200 = zeros(10,1);
Vel_ER_Hi = zeros(10,1);
Gain_R10 = zeros(10,1);
Gain_R20 = zeros(10,1);
Gain_R30 = zeros(10,1);
Gain_R40 = zeros(10,1);
Gain_R50 = zeros(10,1);
Gain_R60 = zeros(10,1);
Gain_R70 = zeros(10,1);
Gain_R80 = zeros(10,1);
Gain_R90 = zeros(10,1);
Gain_R100 = zeros(10,1);
Gain_R110 = zeros(10,1);
Gain_R120 = zeros(10,1);
Gain_R130 = zeros(10,1);
Gain_R140 = zeros(10,1);
Gain_R150 = zeros(10,1);
Gain_R160 = zeros(10,1);
Gain_R170 = zeros(10,1);
Gain_R180 = zeros(10,1);
Gain_R190 = zeros(10,1);
Gain_R200 = zeros(10,1);
Gain_R_Hi = zeros(10,1);
Vel_HL10 = zeros(10,1);
Vel_HL20 = zeros(10,1);
Vel_HL30 = zeros(10,1);
Vel_HL40 = zeros(10,1);
Vel_HL50 = zeros(10,1);
Vel_HL60 = zeros(10,1);
Vel_HL70 = zeros(10,1);
Vel_HL80 = zeros(10,1);
Vel_HL90 = zeros(10,1);
Vel_HL100 = zeros(10,1);
Vel_HL110 = zeros(10,1);
Vel_HL120 = zeros(10,1);
Vel_HL130 = zeros(10,1);
Vel_HL140 = zeros(10,1);
Vel_HL150 = zeros(10,1);
Vel_HL160 = zeros(10,1);
Vel_HL170 = zeros(10,1);
Vel_HL180 = zeros(10,1);
Vel_HL190 = zeros(10,1);
Vel_HL200 = zeros(10,1);
Vel_HL_Hi = zeros(10,1);
Vel_EL10 = zeros(10,1);
Vel_EL20 = zeros(10,1);
Vel_EL30 = zeros(10,1);
Vel_EL40 = zeros(10,1);
Vel_EL50 = zeros(10,1);
Vel_EL60 = zeros(10,1);
Vel_EL70 = zeros(10,1);
Vel_EL80 = zeros(10,1);
Vel_EL90 = zeros(10,1);
```

```
Vel_EL100 = zeros(10,1);
Vel_EL110 = zeros(10,1);
Vel_EL120 = zeros(10,1);
Vel_EL130 = zeros(10,1);
Vel_EL140 = zeros(10,1);
Vel_EL150 = zeros(10,1);
Vel_EL160 = zeros(10,1);
Vel_EL170 = zeros(10,1);
Vel_EL180 = zeros(10,1);
Vel_EL190 = zeros(10,1);
Vel_EL200 = zeros(10,1);
Vel_EL_Hi = zeros(10,1);
Gain_L10 = zeros(10,1);
Gain_L20 = zeros(10,1);
Gain_L30 = zeros(10,1);
Gain_L40 = zeros(10,1);
Gain_L50 = zeros(10,1);
Gain_L60 = zeros(10,1);
Gain_L70 = zeros(10,1);
Gain_L80 = zeros(10,1);
Gain_L90 = zeros(10,1);
Gain_L100 = zeros(10,1);
Gain_L110 = zeros(10,1);
Gain_L120 = zeros(10,1);
Gain_L130 = zeros(10,1);
Gain_L140 = zeros(10,1);
Gain_L150 = zeros(10,1);
Gain_L160 = zeros(10,1);
Gain_L170 = zeros(10,1);
Gain_L180 = zeros(10,1);
Gain_L190 = zeros(10,1);
Gain_L200 = zeros(10,1);
Gain_L_Hi = zeros(10,1);
j=1;k=1;l=1;m=1;n=1;o=1;p=1;q=1;r=1;s=1;t=1;u=1;v=1;w=1;x=1;y=1;z=1;
J=1;K=1;L=1;M=1;
 for i = 1:length(Vel_HR)
     if abs(Vel_HR(i)) < 40
         Vel_HR10(j)=Vel_HR(i);
         Vel_ER10(j)=Vel_ER(i);
         Gain_R10(j)=Vel_ER10(j)/Vel_HR10(j);
         j=j+1;
     end
     if abs(Vel_HR(i)) < 80 && abs(Vel_HR(i)) >40
         Vel_HR20(k)=Vel_HR(i);
         Vel_ER20(k)=Vel_ER(i);
         Gain_R20(k)=Vel_ER20(k)/Vel_HR20(k);
         k=k+1;
     end
     if abs(Vel_HR(i)) < 120 && abs(Vel_HR(i)) >80
         Vel_HR30(l)=Vel_HR(i);
         Vel_ER30(l)=Vel_ER(i);
         Gain_R30(l)=Vel_ER30(l)/Vel_HR30(l);
         l=l+1;
     end
```

```
     if abs(Vel_HR(i)) < 160 && abs(Vel_HR(i)) >120
         Vel_HR40(m)=Vel_HR(i);
         Vel_ER40(m)=Vel_ER(i);
         Gain_R40(m)=Vel_ER40(m)/Vel_HR40(m);
         m=m+1;
     end
      if 160 < abs(Vel_HR(i))
         Vel_HR_Hi(L)=Vel_HR(i);
         Vel_ER_Hi(L)=Vel_ER(i);
         Gain_R_Hi(L)=Vel_ER_Hi(L)/Vel_HR_Hi(L);
         L=L+1;
      end
 end
Avg_Vel_HR=[mean(Vel_HR10);mean(Vel_HR20);mean(Vel_HR30);mean(Vel_HR40)
;mean(Vel_HR_Hi)];
Avg_Gain_R=[mean(Gain_R10);mean(Gain_R20);mean(Gain_R30);mean(Gain_R40)
;mean(Gain_R_Hi)];

j=1;k=1;l=1;m=1;n=1;o=1;p=1;q=1;r=1;s=1;t=1;u=1;v=1;w=1;x=1;y=1;z=1;
J=1;K=1;L=1;M=1;
 for i = 1:length(Vel_HL)
     if abs(Vel_HL(i)) < 40
         Vel_HL10(j)=Vel_HL(i);
         Vel_EL10(j)=Vel_EL(i);
         Gain_L10(j)=Vel_EL10(j)/Vel_HL10(j);
         j=j+1;
     end
     if abs(Vel_HL(i)) < 120 && abs(Vel_HL(i)) >80
         Vel_HL20(k)=Vel_HL(i);
         Vel_EL20(k)=Vel_EL(i);
         Gain_L20(k)=Vel_EL20(k)/Vel_HL20(k);
         k=k+1;
     end
     if abs(Vel_HL(i)) < 160 && abs(Vel_HL(i)) >120
         Vel_HL30(l)=Vel_HL(i);
         Vel_EL30(l)=Vel_EL(i);
         Gain_L30(l)=Vel_EL30(l)/Vel_HL30(l);
         l=l+1;
     end
      if 160 < abs(Vel_HL(i))
         Vel_HL_Hi(L)=Vel_HL(i);
         Vel_EL_Hi(L)=Vel_EL(i);
         Gain_L_Hi(L)=Vel_EL_Hi(L)/Vel_HL_Hi(L);
         L=L+1;
      end
 end
Avg_Vel_HL=[mean(Vel_HL10);mean(Vel_HL20);mean(Vel_HL30);mean(Vel_HL40)
;mean(Vel_HL_Hi)];
Avg_Gain_L=[mean(Gain_L10);mean(Gain_L20);mean(Gain_L30);mean(Gain_L40)
;mean(Gain_L_Hi)];


figure(4)
```

```
hold on
title('Head Velocity(Blue), Eye Velocity(Green),Velocity Direction
(Red)')
plot (vel_H(:,1),'b')
plot (vel_E(:,1),'g')
plot (direction(:,1),'r')   % plot velocity direction array to test

figure(5)
hold on
title('Head Velocity Vs Eye Velocity')
plot (vel_H, vel_E,'k.')
xlabel('Head Velocity(degrees/sec)');
ylabel('Eye Velocity(degrees/sec)');
axis([-100,100,-300,300]);

X = xlim;
Y = ylim;

%-------------------------------------------------------------------------
----
%  Compute polynomial fit for all data
%-------------------------------------------------------------------------
----
p=polyfit(vel_H, vel_E, 1);
t2= min(vel_H):0.1:max(vel_H);
y2=polyval(p,t2);
plot (t2,y2, 'r');
[R,P] = corrcoef(vel_H, vel_E);

%-------------------------------------------------------------------------
----
%  Determine Left vs Right VOR gains
%-------------------------------------------------------------------------
----
figure(6)
hold on
title('Head Velocity Vs Eye Velocity (RIGHT(red) vs LEFT(black))')

axis([X Y])   % set axis limits to previous plot
plot (vel_Hr, vel_Er,'k.')
xlabel('Head Velocity(degrees/sec)');
ylabel('Eye Velocity(degrees/sec)');
    p=polyfit(vel_Hr, vel_Er, 1);
    t2= min(vel_Hr):0.1:max(vel_Hr);
    y2=polyval(p,t2);
    plot (t2,y2, 'r');
    [R,P] = corrcoef(vel_Hr, vel_Er);
p(1)   %gain

%figure(7)
plot (vel_Hl, vel_El,'r.')
xlabel('Head Velocity(degrees/sec)');
```

```
ylabel('Eye Velocity(degrees/sec)');
    p=polyfit(vel_Hl, vel_El, 1);
    t2= min(vel_Hl):0.1:max(vel_Hl);
    y2=polyval(p,t2);
    plot (t2,y2, 'k');
    [R,P] = corrcoef(vel_Hl, vel_El);
p(1)   %gain

%-------------------------------------------------------------------------
----
%   PoG & PoI Calculations
%-------------------------------------------------------------------------
--

pog_X=zeros(file_len,1);
pog_Y=zeros(file_len,1);
pog_Z=zeros(file_len,1);
X_error=zeros(file_len,1);
Y_error=zeros(file_len,1);
Z_error=zeros(file_len,1);

XYZ_error=zeros(file_len,1);
LED_X = 0.364;
LED_Y = 1;
LED_Z = -0.409;
%compute eye point-of-gaze within stationary eye-position segments
for j = 1:file_len
    ml = Lg_X(j)/Lg_Y(j);% Calculating m1 in excel file
    mr = Rg_X(j)/Rg_Y(j);% Calculating m2 in excel file
    bl = abs(LE_Y(j))* ml + LE_X(j);% Calculating b1 in excel file
    br = abs(RE_Y(j))* mr + RE_X(j);% Calculating b2 in excel file
    pog_Y(j) = (br - bl)/(ml - mr);% Calculating Y in excel file
    pog_X(j) = ml*pog_Y(j) + bl;% Calculating X in excel file

    lgQ = atan(Lg_Z(j)/Lg_Y(j))*180/pi;% Calculating Vertical(left)
gaze angle in excel file
    rgQ = atan(Rg_Z(j)/Rg_Y(j))*180/pi;% Calculating Vertical(right)
gaze angle in excel file
    lvo = (pog_Y(j)- LE_Y(j))*tan(lgQ*pi/180);% Calculating
Vertical(left) offset in excel file
    rvo = (pog_Y(j)- RE_Y(j))*tan(rgQ*pi/180);% Calculating
Vertical(right) offset in excel file
    lvi = lvo+ LE_Z(j);% Calculating Vertical(left) intercept in excel
file
    rvi = rvo+ RE_Z(j);% Calculating Vertical(right) intercept in excel
file
    pog_Z(j) = (lvi+ rvi)/2;% Calculating Z in excel file

end
```

```
%-------------------------------------------------------------------------
----
%  Compute intercept error
%-------------------------------------------------------------------------
----
X_cyc=zeros(file_len,1);
Y_cyc=zeros(file_len,1);
Z_cyc=zeros(file_len,1);
X_cal=zeros(file_len,1);
Z_cal=zeros(file_len,1);
Int_err_X=zeros(100,1);
X_left=zeros(file_len,1);
Z_left=zeros(file_len,1);
X_right=zeros(file_len,1);
Z_right=zeros(file_len,1);
L_Int_err_X=zeros(100,1);
R_Int_err_X=zeros(100,1);
X_fov=zeros(file_len,1);
Z_fov=zeros(file_len,1);
nLaserX=zeros(file_len,1);
nzeroed_head_angle=zeros(100,1);
nzeroed_eye_angle=zeros(100,1);
check = zeros(10,1);

for j = 1:file_len
    % Calculating Cyclopian Eye;
    X_cyc(j)= (LE_X(j)+ RE_X(j))/2;
    Z_cyc(j)= (LE_Z(j)+ RE_Z(j))/2;
    Y_cyc(j)= (LE_Y(j)+ RE_Y(j))/2;

    % Calculating Point of Interception (based on target depth);
    r = (LED_Y - Y_cyc(j))/(pog_Y(j) - Y_cyc(j));
    X_cal(j)= X_cyc(j) + r*(pog_X(j)-X_cyc(j));% - offsetx
    Z_cal(j)= Z_cyc(j) + r*(pog_Z(j)-Z_cyc(j));% - offsetz
    X_left(j)= LE_X(j)+ r*(pog_X(j)- LE_X(j)) ;
    X_right(j)= RE_X(j)+ r*(pog_X(j)- RE_X(j));
    Z_left(j) = LE_Z(j) + r*(pog_Z(j)-LE_Z(j));
    Z_right(j) = RE_Z(j) + r*(pog_Z(j)-RE_Z(j));

end
%-------------------------------------------------------------------------
----
%Laser Motion display
%-------------------------------------------------------------------------
----
nocheck=zeros(10,1);
LaserX = zeros(file_len,1);
X_Cal=zeros(10,1);
for j=1:length(X_cal)
    nocheck(j) = isfinite(X_cal(j));
end
i=1;
for j=1:length(nocheck)
```

```
    if nocheck(j) == 1
        X_Cal(i) = X_cal(j);
        i=i+1;
    end
end
Lx = mean(X_Cal(50:100));
stdxcal=std(X_Cal(50:100));
i=1;
for j = 100:file_len
   LaserX(j)= Lx - (1-LE_Y(j))*tan(zeroed_head_angle(j)*pi/180);
   lx = mean (LaserX);
   if (lx+0.1524) < lx < (lx-0.1524) %Choosing an area of interest (the
center foot of the screen)
       nLaserX(i) = LaserX(j);
       nzeroed_head_angle(i) = zeroed_head_angle(j);
       nzeroed_eye_angle(i) = zeroed_eye_angle(j);
       i=i+1;
   end
end

i=1;
for j=100:length(X_cal)
    Int_err_X(i) = X_cal(j)-LaserX(j);
    L_Int_err_X(i) = X_left(j)-LaserX(j);
    R_Int_err_X(i) = X_right(j)-LaserX(j);
    i=i+1;
end
%-------------------------------------------------------------------------
----
%Eliminating NaNs
%-------------------------------------------------------------------------
----
Int_err=zeros(10,1);
for j=1:length(Int_err_X)
    check(j) = isfinite(Int_err_X(j));
end
i=1;
for j=1:length(check)
    if check(j) == 1
        Int_err(i) = Int_err_X(j);
        i=i+1;
    end
end

 std_Int_err_X = std(Int_err);
 avg_Int_err_X = mean(Int_err);
std_zeroed_head_angle = std(zeroed_head_angle);



figure(7)
hold on
title('Head Angle Vs Eye Angle (blue=area of interest)')
```

```
plot (zeroed_head_angle, zeroed_eye_angle,'.g')
plot (nzeroed_head_angle, nzeroed_eye_angle,'.b')
xlabel('Head Angle(degrees)');
ylabel('Eye Angle(degrees)');

figure (8)
hold on
title('Scatter plot (Laser Target(red) vs. point-of-gaze(blue))(green =
area of interest)')
xlabel('X co-ordinate');
ylabel('-Z co-ordinate');
plot(X_cal,-Z_cal,'.b')
plot(LaserX,-Z_cal,'.r')
plot(nLaserX,-Z_cal,'.g')
axis ([-0.2,1.2,-0.15,0.65]);

%-------------------------------------------------------------------------
----
%  Target Foveation
%-------------------------------------------------------------------------
----
for j =1:file_len
   if atan(sqrt((X_cal(j)-LaserX(j))^2))*180/pi < 3.26   % in degrees =
1(fovea)+0.03(target)+0.5(eyelink)+0.1(motion monitor)...double it 2
count for errors during head movement.
       X_fov(j) = X_cal(j);
       Z_fov(j) = Z_cal(j);
   end
end
f = 0;
for j = 1:file_len
   if X_fov(j)~=0
       f = f +1;
   end
c = length(X_cal);

end
FF = f/c*100;
%-------------------------------------------------------------------------
----
%  Final Temporal Plot of both Velocities
%-------------------------------------------------------------------------
----
a =600; %1)600;2)550;3)400;4)650;5)300;6)525;7)600
z =1700; %1)2650;2)2350;3)1850;4)1900;5)1550;6)2000;7)1700
flag=-1;
slope = diff(head_angle)./diff(time);
for i =a:z
    if slope(i)<0 && flag==-1
        if max(slope(i:i+55))<0
            left_start1=i;
            left_end1 = i+55;
            flag=1;
```

```
    end
end
if slope(i)>0 && flag==1
    if min(slope(i:i+55))>0
        right_start1=i;
        right_end1 = i+55;
        flag=-2;
    end
end
if slope(i)<0 && flag==-2
    if max(slope(i:i+55))<0
        left_start2=i;
        left_end2 = i+55;
        flag=2;
    end
end
if slope(i)>0 && flag==2
    if min(slope(i:i+55))>0
        right_start2=i;
        right_end2 = i+55;
        flag=-3;
    end
end
if slope(i)<0 && flag==-3
    if max(slope(i:i+55))<0
        left_start3=i;
        left_end3 = i+55;
        flag=3;
    end
end
if slope(i)>0 && flag==3
    if min(slope(i:i+55))>0
        right_start3=i;
        right_end3 = i+55;
        flag=-4;
    end
end
if slope(i)<0 && flag==-4
    if max(slope(i:i+55))<0
        left_start4=i;
        left_end4 = i+55;
        flag=4;
    end
end
if slope(i)>0 && flag==4
    if min(slope(i:i+55))>0
        right_start4=i;
        right_end4 = i+55;
        flag=-5;
    end
end
if slope(i)<0 && flag==-5
    if max(slope(i:i+55))<0
        left_start5=i;
```

```
            left_end5 = i+55;
            flag=5;
        end
    end
    if slope(i)>0 && flag==5
        if min(slope(i:i+55))>0
            right_start5=i;
            right_end5 = i+55;
            flag=-6;
        end
    end
    if slope(i)<0 && flag==-6
        if max(slope(i:i+55))<0
            left_start6=i;
            left_end6 = i+55;
        end
    end
end
A=vel_H(left_start1:left_end1);
B=vel_H(left_start2:left_end2);
C=vel_H(left_start3:left_end3);
D=vel_H(left_start4:left_end4);
E=vel_H(left_start5:left_end5);
Hl=[A';B';C';D';E'];
lh_vel=mean(Hl);
lh_vel=lh_vel';
F=vel_H(right_start1:right_end1);
G=vel_H(right_start2:right_end2);
H=vel_H(right_start3:right_end3);
I=vel_H(right_start4:right_end4);
J=vel_H(right_start5:right_end5);
Hr=[F';G';H';I';J'];
rh_vel=mean(Hr);
rh_vel=rh_vel';

K=vel_E(left_start1:left_end1);
L=vel_E(left_start2:left_end2);
M=vel_E(left_start3:left_end3);
N=vel_E(left_start4:left_end4);
O=vel_E(left_start5:left_end5);
El=[K';L';M';N';O'];
le_vel=mean(El);
le_vel=le_vel';
P=vel_E(right_start1:right_end1);
Q=vel_E(right_start2:right_end2);
R=vel_E(right_start3:right_end3);
S=vel_E(right_start4:right_end4);
T=vel_E(right_start5:right_end5);
Er=[P';Q';R';S';T'];
re_vel=mean(Er);
re_vel=re_vel';

figure (9)
```

```
hold on
title('Leftwards Velocity plot (Head(blue) and eye(green))')
xlabel('Time');
ylabel('Velocities');
plot(time(1:56),lh_vel,'b')
plot(time(1:56),le_vel,'g')
axis([0,0.28,-200,200]);

figure (10)
hold on
title('Rightwards Velocity plot (Head(blue) and eye(green))')
xlabel('Time');
ylabel('Velocities');
plot(time(1:56),rh_vel,'b')
plot(time(1:56),re_vel,'g')
axis([0,0.28,-200,200]);
%-------------------------------------------------------------------------
----
%  10 deg/s range head velocity sorting
%-------------------------------------------------------------------------
----
peak_A=min(vel_H(left_start1:left_end1));
mean_A=mean(vel_H(left_end1-25:left_end1));
peak_F=max(vel_H(right_start1:right_end1));
mean_F=mean(vel_H(right_end1-25:right_end1));

peak_B=min(vel_H(left_start2:left_end2));
mean_B=mean(vel_H(left_end2-25:left_end2));
% mode_B=mode(vel_H(left_start2:left_end2));
peak_G=max(vel_H(right_start2:right_end2));
mean_G=mean(vel_H(right_end2-25:right_end2));

peak_C=min(vel_H(left_start3:left_end3));
mean_C=mean(vel_H(left_end3-25:left_end3));
% mode_C=mode(vel_H(left_start3:left_end3));
peak_H=max(vel_H(right_start3:right_end3));
mean_H=mean(vel_H(right_end3-25:right_end3));
peak_D=min(vel_H(left_start4:left_end4));
mean_D=mean(vel_H(left_end4-25:left_end4));
% mode_D=mode(vel_H(left_start4:left_end4));
peak_I=max(vel_H(right_start4:right_end4));
mean_I=mean(vel_H(right_end4-25:right_end4));

peak_E=min(vel_H(left_start5:left_end5));
mean_E=mean(vel_H(left_end5-25:left_end5));
peak_J=max(vel_H(right_start5:right_end5));
mean_J=mean(vel_H(right_end5-25:right_end5));
Mean_Left_Velocities = [mean_A;mean_B;mean_C;mean_D;mean_E];
Mean_Right_Velocities = [mean_F;mean_G;mean_H;mean_I;mean_J];
Wave_Data_H = [A,B,C,D,E,F,G,H,I,J];
Wave_Data_E = [K,L,M,N,O,P,Q,R,S,T];
```

VITA

Harshad B. Hegde was born on March 1, 1984 in Mumbai, India and is a citizen of India. He graduated from Don Bosco High School in 1999. He completed his Bachelor of Science in Biomedical Engineering from Mumbai University in Mumbai, India in 2006. He later earned a Master's degree in Biomedical Engineering from Virginia Commonwealth University in Richmond, Virginia, U.S.A.