

Fall 2003

# A LabVIEW program to obtain the initiating component of a vergence eye movement in an open-loop experiment

Anuj P. Daftari

*New Jersey Institute of Technology*

Follow this and additional works at: <https://digitalcommons.njit.edu/theses>



Part of the [Biomedical Engineering and Bioengineering Commons](#)

---

## Recommended Citation

Daftari, Anuj P, "A LabVIEW program to obtain the initiating component of a vergence eye movement in an open-loop experiment" (2003). *Theses*. 520.

<https://digitalcommons.njit.edu/theses/520>

This Thesis is brought to you for free and open access by the Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact [digitalcommons@njit.edu](mailto:digitalcommons@njit.edu).

## Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

**Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation**

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen



The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

## **ABSTRACT**

### **A LABVIEW PROGRAM TO OBTAIN THE INITIATING COMPONENT OF A VERGENCE EYE MOVEMENT IN AN OPEN-LOOP EXPERIMENT**

**by  
Anuj P. Daftari**

The vergence oculomotor system is used in viewing objects moving in depth, such as when a baseball player tracks a ball moving towards him. It is composed of two components according to the Dual Mode Theory; a fast preprogrammed initiating component and a slow feedback-controlled sustaining component. The initiating component is described with open-loop control and the sustaining component is described as a closed-loop feedback control system. Previously, several languages have been utilized to develop programs to study and isolate the open-loop portion of vergence eye movements. Presenting a subject with a stimulus, which initiates an open-loop response has been speculated to adapt the vergence system more than other stimuli. This study utilizes LabVIEW 6i in developing a program capable of real time experimentation to study the open-loop portion of vergence eye movements. LabVIEW offers many options to the programmer and operator with a user-friendly interface for program development as well as an open architecture, allowing flexibility for future studies. This research validated that LabVIEW can be used for open-loop experimentation through a timing analysis and a comparison proving that responses obtained from this code are similar to data published in the literature.

**A LABVIEW PROGRAM TO OBTAIN THE INITIATING COMPONENT OF A  
VERGENCE EYE MOVEMENT IN AN OPEN-LOOP EXPERIMENT**

**by  
Anuj P. Daftari**

**A Thesis  
Submitted to the Faculty of  
New Jersey Institute of Technology  
in Partial Fulfillment of the Requirements for the Degree of  
Master of Science in Biomedical Engineering**

**Department of Biomedical Engineering**

**January 2004**



**APPROVAL PAGE**

**A LABVIEW PROGRAM TO OBTAIN THE INITIATING COMPONENT OF A  
VERGENCE EYE MOVEMENT IN AN OPEN-LOOP EXPERIMENT**

**Anuj P. Daftari**

Dr. Tara L. Alvarez, Thesis Advisor  
Assistant Professor of Biomedical Engineering, NJIT

Date

Dr. Stanley Reisman, Committee Member  
Professor of Biomedical Engineering, NJIT

Date

~~Dr. John L. Semmlow, Committee Member  
Professor of Surgery (Bioengineering), Robert Wood Johnson Medical School – UMDNJ~~

Date

## BIOGRAPHICAL SKETCH

**Author:** Anuj P. Daftari  
**Degree:** Master of Science  
**Date:** January 2004

### Graduate and Undergraduate Education:

- Master of Science in Biomedical Engineering  
New Jersey Institute of Technology, Newark, New Jersey, 2004
- Bachelor of Science with double major in Applied Mathematics  
and Computer Science  
New Jersey Institute of Technology, Newark, New Jersey, 2002

**Major:** Biomedical Engineering

### Presentations and Publications:

- A. Daftari, T.L. Alvarez, F. Chua, R. DeMarco, K. Ciuffreda,  
“The Dynamics of Convergence Insufficiency”,  
Proceedings of the IEEE 29th Annual NorthEast Bioengineering Conference,  
Newark, NJ, Mar. 2003.
- A.P. Daftari, T.L. Alvarez, M.L. Kung, J.L. Semmlow,  
“Neuro-Control in Divergence Eye Movements”,  
Proceedings of the IEEE 29th Annual NorthEast Bioengineering Conference,  
Newark, NJ, Mar. 2003.
- L. Herrera, T.L. Alvarez, A.P. Daftari, J.L. Semmlow,  
“Motor Learning In The Binocular Themeing System”  
Proceedings of the IEEE 29th Annual NorthEast Bioengineering Conference,  
Newark, NJ, Mar. 2003.
- F.B. Chua, A.P. Daftari, T. Alvarez, R. DeMarco, M.T. Bergen, K.D. Beck, R.J.  
Servatius, “Effect of Light Flashes on Saccadic Oculomotor Control”,  
Proceedings of the IEEE 29th Annual NorthEast Bioengineering Conference,  
Newark, NJ, Mar. 2003.



**Presentations and Publications (Continued):**

T.L. Alvarez, K.D. Beck, A. Daftari, F. Chua, R. DeMarco, M.T. Bergen, R.J. Servatius,  
“The Effect of After Image on Saccadic Eye Movements”, Proceedings of  
ARVO's 2003 Annual Conference, Ft. Lauderdale, FL, 2003.





To my family and friends.

...

Without the support and guidance of everyone who I love and admire  
this thesis would not have been possible.

...

Thank You All



## ACKNOWLEDGMENT

I would like to thank Dr. Tara L. Alvarez, my advisor, for bringing me into her lab and providing me with the opportunity to extend my intellect and experience through research. Her dedication and advisement encouraged me to pursue my interests. She is a remarkable woman with amazing teaching skills who inspires everyone through her research. I would also like to thank Dr. John Semmlow for providing assistance, advisement, and guidance throughout my thesis. Special thanks goes to Dr. Stanley Reisman for advising me throughout my graduate career and teaching me the signal processing and instrumentation aspects of Biomedical Engineering.

My gratitude goes to Dr. Dawn Lott in the Mathematical Sciences department at New Jersey Institute of Technology. I have had the honor of knowing Dr. Lott for more than four years and she has helped me through many tough times. She has encouraged me to pursue my dreams and has expected nothing less than excellence from me.

I am very grateful to Florence B. Chua for the invaluable assistance she has provided throughout my thesis. Without her eyes and time, I would not have been able to complete a thorough evaluation of the open-loop program. She has been very inspirational and insightful throughout the development and has supported and encouraged me to do my best.

Thank you to the staff and students of the Vision and Neural Engineering Laboratory and the Biomedical Engineering Department. They have provided me with an excellent environment to work in and have assisted me in many tasks.

I would like to give a heartfelt thanks to my family. To my parents, Prakash and Amita Daftari, my brother, Amit Daftari, and my aunts, uncles, and cousins, all of whom

I love so dearly. They have encouraged me to pursue my dreams, take risks, and attain my goal. They have all been by my side when I needed them the most and have supported me in all my decisions.

In addition to my family, I am very fortunate to have wonderful friends who provide support and encouragement. They have helped show me the world in a way no one else could dream possible. They have been by my side through the rough times and the good.

Without all the people in my life, I would not be who I am today. I appreciate what everyone has done for me and continue to do so. Thank You Everyone.

## TABLE OF CONTENTS

<b>Chapter</b>	<b>Page</b>
1 INTRODUCTION .....	1
1.1 Objective .....	1
1.2 Background Information .....	3
1.2.1 Physiology of The Eye.....	3
1.2.2 Vergence Eye Movements .....	7
1.2.3 Dual Mode Theory .....	7
1.2.4 Ramp Stimuli Define Control Aspects of Vergence Oculomotor System .....	8
1.2.5 Independent Component Analysis .....	10
1.2.6 Goals .....	12
2 INSTRUMENTATION METHODOLOGY .....	13
2.1 Visual Equipment Utilization .....	14
2.1.1 Monitors and Mirrors .....	14
2.2 Peripheral Equipment Utilization .....	16
2.2.1 Eye Movement Tracker .....	16
2.2.2 Data Acquisition Board .....	18
2.2.3 Bayonet Neill-Concelman (BNC) Adapter .....	19
2.2.4 Trigger Button .....	20
2.2.5 Function Generator .....	21
2.3 Timing Enhancement .....	22
2.3.1 Aliasing .....	23
2.3.2 Software Timer .....	26

**TABLE OF CONTENTS**  
**(Continued)**

<b>Chapter</b>	<b>Page</b>
2.3.3 External Programs Not Under LabVIEW Control .....	26
2.4 Programming Languages .....	26
2.4.1 LabVIEW .....	27
2.4.2 Matlab® .....	27
2.5 Integration and Usage of Code Completed by Others .....	30
2.5.1 Monitor Stimulus Display .....	30
2.5.2 Eye Movement Tracker Setup .....	30
2.5.3 Vergence Degree to Pixel Calculation .....	32
2.6 Open-Loop Program 1.0 Structure .....	32
<b>3 THREE-POINT CALIBRATION METHODOLOGY .....</b>	<b>35</b>
3.1 Front Panel and Structure .....	36
3.2 Sequence One .....	37
3.2.1 Subject Notification of Three-Point Start .....	37
3.2.2 Trigger Button Control and Options .....	40
3.2.3 Presentation of Initial Target .....	40
3.2.4 Array of Targets for Analysis .....	41
3.3 Sequence Two.....	42
3.3.1 Data Acquisition (DAQ) Components .....	44
3.3.2 Array of Targets for Monitor .....	45
3.3.3 Location to Analyze Eye Movement Data for Mean and Standard Deviation .....	45
3.3.4 Large For-Loop With Blink Recognition .....	46

**TABLE OF CONTENTS**  
**(Continued)**

<b>Chapter</b>	<b>Page</b>
3.3.5 For-Loop Sequence Zero .....	49
3.3.6 Trigger Control .....	49
3.3.7 Data Acquisition Read and Start .....	51
3.3.8 Analysis for Mean and Standard Deviation .....	52
3.3.9 After Completion of For-Loop .....	54
3.4 Regression Analysis .....	56
4 OPEN-LOOP METHODOLOGY .....	58
4.1 Open-Loop Sequences .....	59
4.1.1 Stimulus Selection .....	59
4.1.2 Initial Stimulus Presentation .....	62
4.1.3 Trigger Button Initiates DAQ and Random Delay .....	64
4.1.4 Start Filling Buffer – AI Start .....	67
4.1.5 Stimulus Step and Open-Loop .....	67
4.1.6 Clear DAQ Components .....	71
4.2 Open-Loop Subvi .....	71
4.2.1 Iterations of For-Loop .....	73
4.2.2 Acquire Eye Movement Position From Eye Movement Tracker .....	73
4.2.3 Determine Eye Position In Degree and New Target Location .....	74
4.2.4 Target Presentation on Computer Monitor .....	75
4.3 Open-Loop Methodology .....	77
4.3.1 Methodology One .....	82



**TABLE OF CONTENTS**  
**(Continued)**

<b>Chapter</b>	<b>Page</b>
4.3.2 Methodology Two .....	84
4.3.3 Methodology Three .....	86
4.3.4 Optimal Methodology .....	89
4.3.5 Programmer Error .....	90
<b>5 METHODOLOGY OF ACCESSORY COMPONENTS.....</b>	<b>91</b>
5.1 Variable Initialization .....	91
5.1.1 Front Panel and Text Variables .....	91
5.1.2 Calibration Switches .....	94
5.1.3 Stimulus Presentation, Timing, and Move Target By .....	95
5.1.4 Complete Variable Initialization Button .....	95
5.2 Save to File .....	96
5.3 Determine to Calibrate Or Not .....	98
5.4 Exit Program .....	100
5.5 SubVI's Created for Modularization and Enhancement of Architecture .....	101
5.5.1 DAQ Initialization .....	101
5.5.2 Stimulus Counter Organizer .....	102
5.5.3 Random Number Generator .....	104
5.5.4 MOD Function .....	106
<b>6 EXPERIMENTAL METHODOLOGY.....</b>	<b>107</b>
6.1 Utilized Instrumentation and Software .....	108
6.1.1 Data Acquisition Instrumentation .....	108

**TABLE OF CONTENTS**  
**(Continued)**

<b>Chapter</b>	<b>Page</b>
6.1.2 Visual Instrumentation .....	108
6.1.3 Software .....	109
6.2 Experiment Setup and Layout .....	110
6.2.1 Experimentation Cube .....	110
6.2.2 Program Parameters .....	113
6.3 Subject Instructions for Open-Loop Experiment .....	117
6.3.1 Subject Arrival .....	117
6.3.2 Limbus Tracking Device Setup .....	118
6.3.3 Experimentation .....	120
6.4 Data Analysis .....	121
6.4.1 Three Point Calibration .....	121
6.4.2 Peak Velocity .....	121
6.4.3 Main Sequence .....	122
7 RESULTS .....	123
7.1 Three Point Calibration .....	123
7.2 Stimulus Target Moves 60 Percent of Eye Movement .....	126
7.2.1 Raw Data .....	126
7.2.2 Main Sequence .....	130
7.2.3 Peak Velocity .....	131
7.3 Target Receives 100 Percent Feedback of Eye Movement Response .....	132
7.3.1 Raw Data .....	132

**TABLE OF CONTENTS  
(Continued)**

<b>Chapter</b>	<b>Page</b>
7.3.2 Main Sequence .....	136
7.3.3 Peak Velocity .....	137
7.4 Timing Analysis .....	138
7.4.1 Aliasing Technique .....	138
7.4.2 Software Timer .....	139
8 DISCUSSION .....	141
9 CONCLUSION .....	147
10 FUTURE RESEARCH .....	149
APPENDIX A TRIGGER BUTTON SCHEMATIC .....	151
APPENDIX B MONITOR STIMULUS PRESENTATION .....	152
B.1 Front Panel .....	152
B.2 Clear Screen .....	153
B.3 Display Line .....	154
B.4 Oscillating Circle .....	155
APPENDIX C LIMBUS TRACKING DEVICE ADJUSTMENT AND DEGREE TO PIXEL CORRELATION .....	156
C.1 Front Panel .....	157
C.2 Code for Target Movement on Monitors .....	157
APPENDIX D DEGREE TO PIXEL CONVERSION .....	160
REFERENCES .....	161

## LIST OF TABLES

<b>Table</b>		<b>Page</b>
4.1	Position of Eyes When % <i>Target Move By</i> Variable is Set to Zero Percent and Initial Step Size is Four .....	79
4.2	Position of Eyes When % <i>Target Move By</i> Variable is Set to Sixty Percent and Initial Step Size is Four .....	80
4.3	Position of Eyes When % <i>Target Move By</i> Variable is Set to One Hundred Percent and Initial Step Size is Four .....	81

## LIST OF FIGURES

Figure	Page
1.1 The three layers of the eye. ....	4
1.2 At the back of the eye is a region with densely packed photoreceptors called the macula at which location the eyes initiate a movement to fixate the target onto for detailed viewing. At the center of the macula is the fovea, a region where the target of interest is being focused onto.....	5
1.3 Six oculomotor muscles control the movement of the eyes in horizontal, vertical, and torsional directions. The muscles responsible for these movements are the medial and lateral rectus, and the superior and inferior obliques. ....	6
1.4 Left diagram displays the two components, initiating and sustaining, together when initiating a vergence eye movement. The Right diagram shows the feedback portion of the sustaining component being removed from the vergence eye movement. ....	8
1.5 (Left) Responses and velocity traces of vergence movements when given a ramp stimulus of .7 deg/sec to 36 deg/sec. Four degree max amplitude given. (Right) The initial step amplitudes as a function of ramp velocity for the ramp responses shown in the figure to the left. ....	9
1.6 A four degree convergence response. ....	10
1.7 (a) Graph of 28 responses collected from one subject conducting 4° divergence eye movements. An independent component analysis was completed on these responses to acquire the two independent components. (b) Initiating Component (c) Sustaining Component. ....	12
2.1 Monitor and mirror setup for vergence experiments. The mirrors are placed approximately 45° from the subject's line of sight. ....	16
2.2 Left is the Limbus Tracking device placed on the subject's head to record eye movements. Right is the electronic control module used to adjust the bias and gain of the incoming signal. ....	17
2.3 Transmission and detection of light by limbus tracking device diodes (light emitters and phototransistors). ....	18
2.4 Data Acquisition Board to sample the data coming from the control module. ...	19
2.5 Bayonet Neill-Concelman (BNC) 2090 Adapter. ....	19

**LIST OF FIGURES**  
(Continued)

<b>Figure</b>	<b>Page</b>
2.6 Trigger Button. ....	21
2.7 BK Precision Model 4040A, 20 MHz Sweep Function Generator. ....	22
2.8 A sinusoidal wave with a frequency of 1.5 Hz and an amplitude of 4.0. The open-loop code samples the sinusoid three times at three positions. The sampling of the sinusoid occurs in phase with the frequency, so the target appears on the monitors at only three positions. ....	24
2.9 A sinusoidal wave with a frequency of 1.5 Hz and an amplitude of 4.0. The open-loop code samples the sinusoid at multiple positions. The sampling of the sinusoid occurs out of phase with the frequency, so the target appears on the monitors in multiple positions. ....	25
2.10 Order in which <i>Open-Loop Program 1.0</i> executes from start to finish. ....	34
3.1 Three point calibration front panel. ....	36
3.2 Top view of sequence one of the three point calibration. ....	38
3.3 (a) Popup notification code for three-point calibration. (b) Trigger button for three point calibration and determine if option one or option two should follow. ....	39
3.4 The following two sequences (a) clears the screen and (b) presents the initial target for the subject to fixate on. ....	41
3.5 Initialize array with three targets in degrees. ....	42
3.6 Top view of sequence two of the three point calibration. ....	43
3.7 DAQ initialization. ....	45
3.8 Array initialization with targets present to subject during three point. ....	45
3.9 Code determining where data should be analyzed for mean and standard deviation during each target presentation to the subject. ....	46
3.10 Blink recognition code. (a) Overall view of for-loop and structure within. (b) False statement displays nothing in the situation of no blink. (c) The operator is informed the subject has blinked and the same target will be presented to the subject again. ....	48

**LIST OF FIGURES  
(Continued)**

<b>Figure</b>	<b>Page</b>
3.11 (a) Clears the monitor of all data and figures. (b) Presentation of the targets (initial, second, and final) in sequence to the subject determined by the location in the for-loop. ....	49
3.12 Determines whether subject has control of when data acquisition begins or subject has control. (a) When set to on, true in Boolean, the subject has control. (b) When set to off, false in Boolean, the computer has control. ....	51
3.13 Controls data acquisition. (a) Starts acquiring data from the eye movement tracker and storing into a buffer. (b) Removes data from the buffer and stores displays in a waveform graph. ....	52
3.14 Analysis for mean and standard deviation for the right and left eye independently. ....	54
3.15 Clear DAQ components. ....	55
3.16 Input variables into regression analysis subvi for output of several eye movement variables and an analysis. ....	55
3.17 Input variables into regression analysis subvi for output of several eye movement variables and an analysis. ....	56
3.18 XY-Graph displaying regression analyses of three point calibration and the original three point calibration values. ....	57
3.19 Input variables into regression analysis subvi for output of several eye movement variables and an analysis. ....	57
4.1 Sequence one: Stimulus Selection. ....	59
4.2 Stimulus selection (a) Counter control and stimulus selection. (b) Selection of the stimulus type. (c) Selection of the stimulus parameters for the particular stimulus number. ....	61
4.3 Setup of stimulus to left eye and right eye. ....	62
4.4 Left Eye Subvi (a) Overall structure of of subvi and sequence zero determines pixel position for inputted degree. (b) Sequence one clears monitors. (c) Sequence two presents target to subject. ....	64
4.5 DAQ initialization and random delay. ....	66

**LIST OF FIGURES  
(Continued)**

<b>Figure</b>	<b>Page</b>
4.6 Start data acquisition. ....	67
4.7 Stage one to present new target to subject. ....	69
4.8 Calls the open-loop subvi to begin a real time analysis of each scan to present next target position to subject. ....	70
4.9 Overall structure of the open-loop subvi. ....	72
4.10 Acquire scans from DAQ card and split scans for each eye into individual array for independent analysis. ....	74
4.11 Determine position of eyes in degrees and position of next target to display. ...	75
4.12 (a) Overall structure to display next target and remove previous target along with checking for saturation of target. (b) Determine pixel position of next target to display. (c) Determine pixel position of previous target to remove. (d) Display and remove the respective targets simultaneously. ....	76
4.13 Time plot of a step movement. Convergence movement of a four degree step when zero percent feedback given to target during open-loop stimulation. Red dotted line shows the position of the stimulus with respect to time. Blue line shows the eye movement response. ....	79
4.14 Time plot of an open-loop response. An initial four degree convergence eye movement is followed by subsequent eye movements to fixate on target. Target receives sixty percent feedback during open-loop stimulation. Red dotted line shows the position of the stimulus with respect to time. Blue line displays the eye position with respect to time. The oculomotor system is able to fixate on target at 10 degrees. ....	80
4.15 Time plot of an open-loop response. An initial four degree convergence eye movement is followed by subsequent eye movements to fixate on target. Target receives one hundred percent feedback during open-loop stimulation. Red dotted line shows the position of the stimulus with respect to time. Blue line displays the eye position with respect to time. The oculomotor system is able to fixate on target at 16 degrees, at which point the target is set to saturate and cease movement. A constant four degree error is maintained between target and position of eyes. ....	81
4.16 Open-Loop methodology one. ....	83



**LIST OF FIGURES  
(Continued)**

<b>Figure</b>	<b>Page</b>
4.17 Open-Loop methodology two. ....	85
4.18 Open-Loop methodology three. ....	88
5.1 Variable initialization front panel. ....	93
5.2 Save to data file. ....	97
5.3 (a) Code to determine what should proceed, calibration or open-loop. (b) Close-up of text in code. ....	99
5.4 Close data file and clear DAQ card. ....	101
5.5 DAQ Initialization. ....	102
5.6 Stimulus counter and stimulus selection. ....	104
5.7 Random number generator. ....	105
5.8 MOD – Remainder. ....	106
6.1 Instrumentation Setup. ....	109
6.2 Experimentation cubical. ....	111
6.3 Chinrest. ....	112
6.4 Mirror Orientation. ....	112
6.5 Variable Initialization Front Panel. ....	116
6.6 Adjustment knobs on limbus tracking device. ....	118
6.7 Data acquisition for adjustment of limbus tracking device. Top is left eye and bottom is right eye. ....	119
6.8 Three point calibration XY-graph. ....	120
7.1 Subject one three-point calibration regression analysis for the correlation coefficient and regression line equation when the stimulus is given 60 percent feedback of the eye movement response. ....	124

**LIST OF FIGURES  
(Continued)**

<b>Figure</b>	<b>Page</b>
7.2 Subject two three-point calibration regression analysis for the correlation coefficient and regression line equation when the stimulus is given 60 percent feedback of the eye movement response. ....	124
7.3 Subject one three-point calibration regression analysis for the correlation coefficient and regression line equation when the stimulus is given 100 percent feedback of the eye movement response. ....	125
7.4 Subject two three-point calibration regression analysis for the correlation coefficient and regression line equation when the stimulus is given 100 percent feedback of the eye movement response. ....	125
7.5 Subject one raw data given a stimulus target with 60 percent feedback from the eye movement response. Left graph shows the phase domain (velocity plotted as a function and right graph displays the position vs time (solid blue) ... and velocity vs time (dashed green). (a, b) Subject's target saturates at 10 degrees. (c, d) Subject's target saturation set at 16 degrees, but saturation occurs at 12 degrees. ....	127
7.6 Subject two raw data for when the target is able to move 60 percent of the eye movement. Left graph shows the phase domain (velocity as a function of position) and right graph displays the position (solid blue) and velocity (dashed green) as a function of time. (a, b) Subject's target saturates at 10 degrees. (c, d) Subject's target saturates at 12 degrees. ....	129
7.7 Main sequence ratio for when the target contains 60 percent feedback from the eye movement response. Only first two peaks are shown. (a) Subject one (b) Subject two. ....	131
7.8 Peak velocity for when the target changes with 60 percent feedback from the eye movement response. Only the first two peak velocity movements are quantified. (a) Subject one (b) Subject two. ....	132
7.9 Subject one raw data for when the target is able to move 100 percent of the eye movement. Left graph shows the phase domain (velocity as a function of position) and right graph displays the position (solid blue line) and velocity (dashed green line) as a function of time. ....	133

**LIST OF FIGURES**  
**(Continued)**

<b>Figure</b>	<b>Page</b>
7.10 Subject two raw data for when the target receives 100 percent feedback from the eye movement. Left graph shows the phase domain and right graph displays the position (solid blue) and velocity (dashed green) as a function of position. (a, b) Subject's target saturates at 14 degrees. (c, d) Subject's target saturates at 20 degrees. ....	135
7.11 Main sequence ratio for when the target is able to move 100 percent of the eye movement. Only first three peaks are shown. (a) Subject one (b) Subject two. ....	137
7.12 Peak velocity for when the target receives 100 percent feedback of the eye movement. Only first three peaks are shown. (a) Subject one (b) Subject two. ....	138
8.1 Data acquired by Semmlow and colleagues from an open-loop experiment where the stimulus received 100 percent feedback from the eye movement response. Data from four subjects (a,b) Received four degree initial steps. (c,d) Received two degree initial steps. ....	143
8.2 Data acquired Semmlow and colleagues from an open-loop experiment where the stimulus received 100 percent feedback from the eye movement responses and were initially presented a four degree step. The subject who performed this movements is subject one in this thesis. ....	144
8.3 Main sequence of the two subjects run by Semmlow and colleagues under open-loop conditions. + indicates closed loop responses, open square indicates first peak in the open-loop response, and plus inside square indicates second peak in the open-loop response. (a) Subject one. (b) Subject two. Note subject two participated in this thesis research to validate this code and for this research was subject 1. ....	145
8.4 Data acquired by Alvarez and colleagues from an open-loop experiment where the stimulus received 60 percent feedback from the eye movement. ...	146
A.1 Trigger button schematic in LabVIEW. ....	151
B.1 Front Panel displays a single green line to the subject. ....	152
B.2 This case statement clears the screen when called upon. ....	153

**LIST OF FIGURES  
(Continued)**

<b>Figure</b>	<b>Page</b>
B.3 This case statement displays a line stimulus at a specified position, with the given height, width, and color. This statement can also remove a line at the specified position. ....	154
B.4 This case statement presents the subject with a circular figure at a specified position, with the given height, width, and color. ....	155
B.5 This is an example of the code written in Figure B.4. ....	155
C.1 Front panel for the eye movement tracker setup and the degree to pixel correspondence. ....	157
C.2 Entire structure of code and inside case statement (denoted as true/false) displays the final target. In the case of a vergence movement, false statement, it is 10°. ....	158
C.3 This is the other case statement, which displays the initial target. In the case of a vergence movement, it is 4°. ....	158
C.4 following code is part of both true case and false case. This code enables the target to be stopped in its location, depending on whether the true case is called upon or the false. ....	159
C.5 Data acquisition from each eye during limbus tracking device setup. ....	159
D.1 Front Panel displays a single green line to the subject. ....	161

# CHAPTER 1

## INTRODUCTION

### 1.1 Objective

The vergence oculomotor system works through a dual processing mechanism. The first mechanism conducts a fast feed forward initiating movement to bring the eyes in the vicinity of the target termed the initiating component. The second mechanism termed the sustaining component is a slow feedback process which reduces the error between the position of the eyes and the target, thus bringing the image of the target onto the fovea and in direct view. This in turn reduces disparity.

Attempting to isolate the two mechanisms is a difficult task. This thesis presents a computer program written in LabVIEW 6i to isolate the initiating component in a vergence eye movement. This program has been defined as an open-loop protocol because the code removes the feedback portion of the eye movement by maintaining a constant error during tracking of the object. However the initiating component does not rely on feedback to initiate the eye movement and will thus be the only component present during the experiment. Essential to the open-loop protocol is the three-point calibration used to equate the present digitized eye movement signal to its corresponding degree value and calculate the precise location of the eyes when determining where the next position the target should appear.

LabVIEW was utilized because it offers a diverse array of programming options needed for research. Some of LabVIEW's key attributions are data acquisition and

analysis. The central question of this thesis is: can LabVIEW be used for open-loop condition eye movement experiments? The purpose is to develop and evaluate a system for generating an open-loop response.

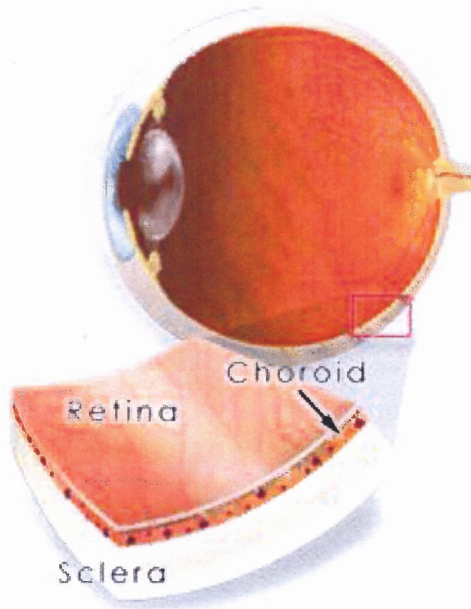
Several languages have been used in the past to collect eye movement data in an open-loop condition. Programs have been designed in many languages such as FORTRAN, C, and MATLAB to elicit vergence responses. As programming languages improve, creating more powerful and versatile platforms, the software must advance as well into these new platforms [13]. This gives the researcher greater control over the software, keeps him or her current with advancing technology, and allows the software to be shared among other scientists.

LabVIEW is a higher level language, which offers a wide variety of options graphically creating a more appealing user interface, which FORTRAN, C, and other languages have not incorporated effectively. In addition the LabVIEW interface is easier to learn how to use for other students and researchers who did not program the platform making it a more robust program to be used for many years of experimentation in a given laboratory. Data acquisition becomes more efficient for the researcher because they can easily change stimulus types and view data as it is being collected from the subject. Furthermore, this thesis has the computer architecture to allow the expansion and modification of the program for future studies.

## **1.2 Background Information**

### **1.2.1 Physiology of The Eye**

The eye is composed of three layers shown in Figure 1.1. The inner most layer is called the retina. The retina contains millions of photoreceptors to convert the light entering through the pupil into an electrical signal for processing by the brain. The back of the eye on the retina is a region called the macula. This portion of the eye is highly sensitive to light and is densely packed with photoreceptors for detailed viewing of the image [18, 31]. The center of the macula is called the fovea which is responsible for sharpness. An image of the macula and fovea can be seen in Figure 1.2. The target being looked at is focused in on this region and allows reading, driving, and other activities to take place with high accuracy and precision. The eyes move to maintain the detail of the object projected onto the fovea. The eyes also see the surrounding environment of the object, but with less detailed resolution. This is because the information is presented on the periphery, outside the fovea, where there are much fewer photoreceptors [18, 31]. For instance, take two pencils and put one directly in front and the other to the side. When looking at the pencil in front, the image of it is very clear, but the image of the pencil to the side is faint.

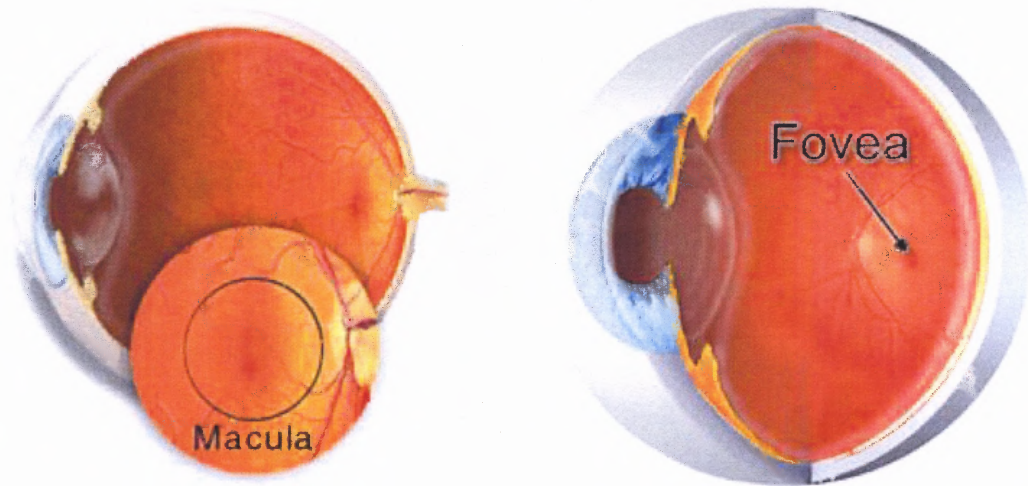


**Figure 1.1** The three layers of the eye [30].

There are two types of photoreceptors in the eye, rods and cones. Cones are primarily centered in the macula and are very sensitive to light and allow images to be seen in color [18, 31]. On the other hand, rods appear in the periphery, outside the macula, and are highly receptive in dim light to allow night vision [18, 31].

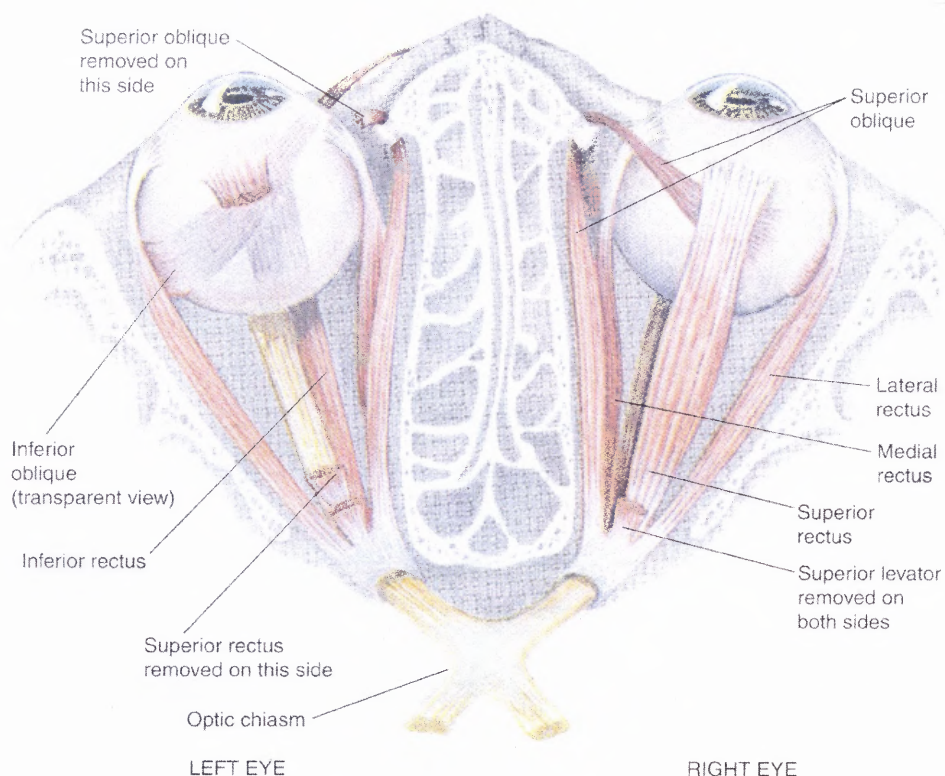
The next layer is called the choroid. The choroid is responsible for providing blood flow with enriched oxygen and nutrients to the retina and other regions of the eye [18, 31]. It is also attached to the edges of the nerve cord in the back of the eye and connects to the ciliary body in the front of the eye [18, 31]. The nerve cord or optic nerve is responsible for taking the electrical signal from the photoreceptors to the brain for processing. The ciliary body on the other hand is responsible for changing the shape of the lens for focusing of the target image onto the fovea [18, 31].





**Figure 1.2** At the back of the eye is a region with densely packed photoreceptors called the macula at which location the eyes initiate a movement to fixate the target onto for detailed viewing. At the center of the macula is the fovea, a region where the target of interest is being focused onto [30].

The final layer of the eye is the sclera. This portion of the eye provides protection and movement in the horizontal, vertical, and torsional directions fovea [18, 31]. Movement is conducted through the manipulation of six muscles, the lateral and medial rectus, superior and inferior rectus, and superior and inferior obliques, as shown in Figure 1.3. Muscles always come in pairs, one that acts as the agonist and the other as the antagonist. The medial and lateral rectus muscles are responsible for horizontal movements of the eyes and the superior and inferior rectus muscles initiate vertical movements fovea [18, 31]. The last pair of muscles is the superior and inferior obliques. They are responsible for the rotation of the eyes and act through a pulley system fovea [18, 31] as seen in Figure 1.3.



**Figure 1.3** Six oculomotor muscles control the movement of the eyes in horizontal, vertical, and torsional directions. The muscles responsible for these movements are the medial and lateral rectus, and the superior and inferior obliques [31].

With all the components of the eyes working together, five forms of eye movements can take shape: saccades, smooth pursuit, vergence, vestibulo-ocular, and optokinetic. The goal of these movements is to keep the target image situated on the fovea for detailed and accurate viewing. If these mechanisms did not coexist to mediate the movement of the eyes in the proper location, diplopia would result. With diplopia, there is only one real world target image, but the eyes perceive the object as two images. This thesis will primarily focus on the vergence oculomotor system.

### **1.2.2 Vergence Eye Movements**

Vergence is the oculomotor system that mediates eye movements in the opposite direction to reduce disparity and track images in depth. They are known as disjunctive movements. Disparity is the error generated when the target image is not on the fovea and is perceived as two images. Disparity is defined as the difference between where the object is projected on the back of the retina and fovea. To reduce the disparity, two forms of movement can take place: convergence, the movement of the eyes inward, and divergence, the movement of the eyes outward.

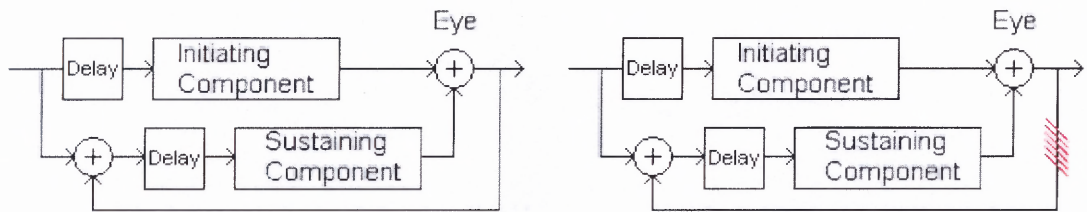
To mediate the movement of the eyes in opposite direction allows for depth perception in order to track objects. Depth perception is controlled by the inward and outward movement of the eyes by the lateral and medial rectus muscles which are closely guided by centers in the midbrain [18]. Both the eyes receive the signal to conduct a disjunctive eye movement with approximately equal amplitudes [20].

In the past, the vergence system was thought to be under continuous feedback control, a single mechanism controlling eye movements [16, 24]. However, behavioral and neurophysiological results have shown that the vergence system is not under only feedback control, but another mechanism accompanies it, a feed forward portion [15, 21, 33]. It follows a Dual-Mode Theory.

### **1.2.3 Dual Mode Theory**

The Dual Mode Theory for disparity vergence states there are two control components used to mediate a vergence eye movement [3, 12, 14, 26, 27, 28, 29]. There is a fast, preprogrammed, transient response giving rise to the initiating feed forward component and a sustaining feedback component to move the eyes to fuse a single perceived image

with high accuracy [28, 29]. The initiating component brings the eyes to a position close to the target; it is the sustaining component which further reduces the disparity so that the eyes perceive a single image and the target appears on the fovea. The sustaining component creates the closed loop system and the initiating component creates the open loop portion as Figure 1.4 depicts.



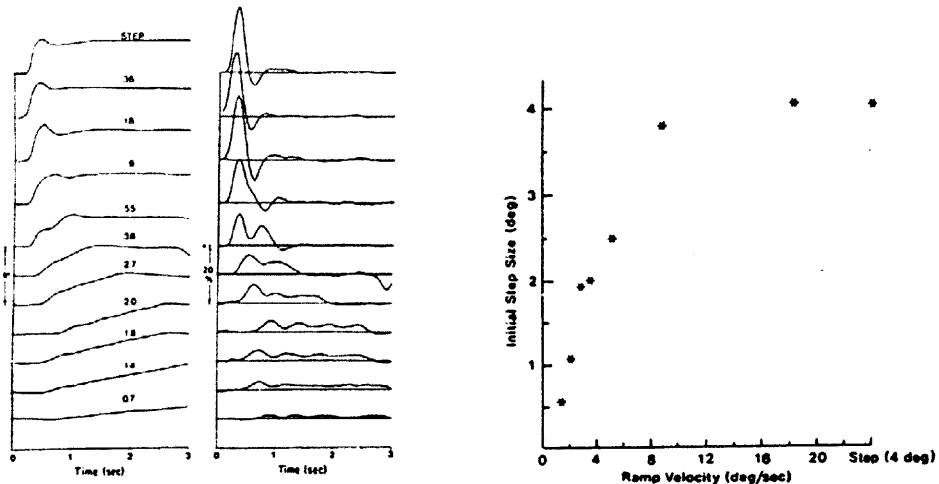
**Figure 1.4** Left diagram displays the two components, initiating and sustaining, together when initiating a vergence eye movement. The Right diagram shows the feedback portion of the sustaining component being removed from the vergence eye movement.

Many experiments have been completed to ascertain the validity of the Dual Mode Theory. The dual component nature was first seen in an experiment conducted by Westheimer and Mitchell in 1969 [33]. They presented short stimuli lasting between 50 and 100 msec and noticed significant vergence eye movement in this short time period before a feedback system could become activated [33]. This behavior could easily be described if the vergence system, in addition to feedback, contained a feedforward open loop paradigm.

#### 1.2.4 Ramp Stimuli Define Control Aspects of Vergence Oculomotor System

One of the methods to ascertain whether the vergence system is only a feedback driven system or a combined feed forward and feedback driven system is to perform an experiment where the subject is presented with a series of different velocity ramp stimuli. As the subject tracks the ramp stimulus, the eyes move at a designated velocity to track

the moving target. Semmlow and colleagues (Semmlow, 1986) utilized ramps at various velocities to determine the mechanistic nature of the vergence system.

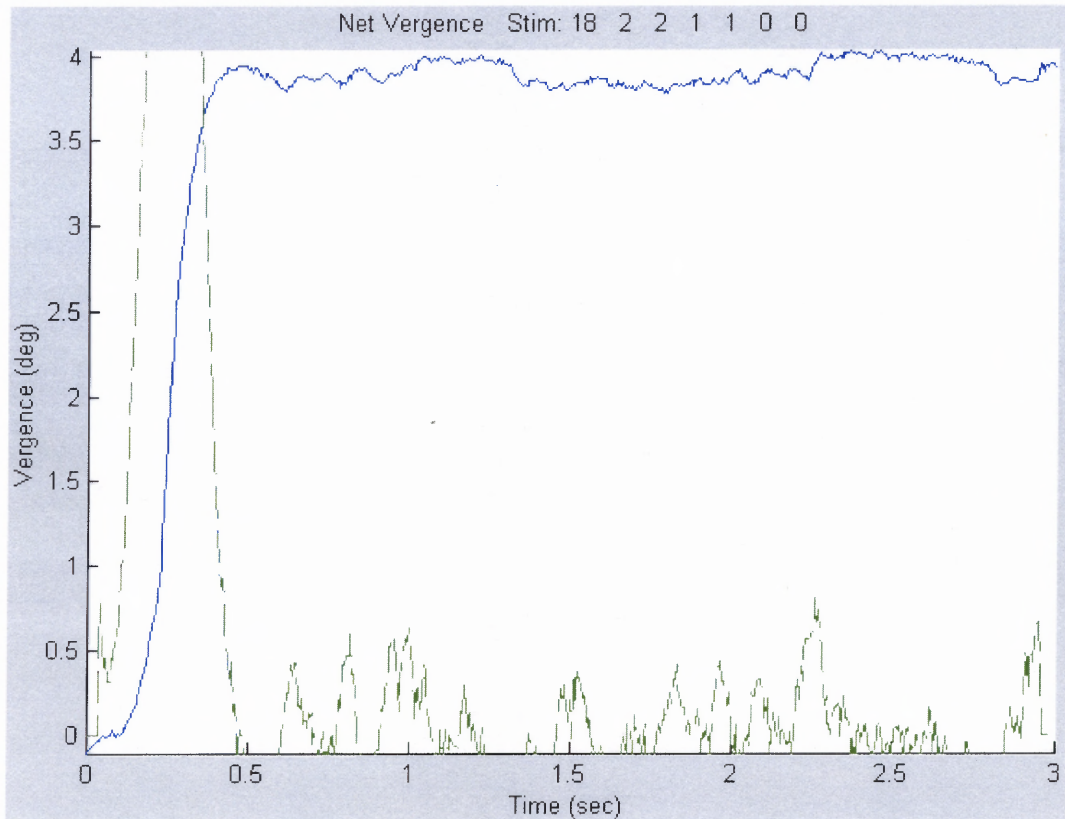


**Figure 1.5** (Left) Responses and velocity traces of vergence movements when given a ramp stimulus of .7 deg/sec to 36 deg/sec. Four degree max amplitude given. (Right) The initial step amplitudes as a function of ramp velocity for the ramp responses shown in the figure to the left [26].

Ramp stimuli ranging from .7 deg/sec to 36 deg/sec were presented to the subjects. They found the subject was able to smoothly track the moving target up to approximately two deg/sec. After two deg/sec and up to nine deg/sec, the subject showed a step/ramp pattern of movement, while everything beyond nine deg/sec showed up as steps [26]. The results from their experiment can be seen in Figure 1.5. This data suggested that a preprogrammed initiating component exists in which the amplitude of the initial step and the target velocity are proportional, thus getting the eyes oriented to look in the vicinity of the target [26].

### 1.2.5 Independent Component Analysis

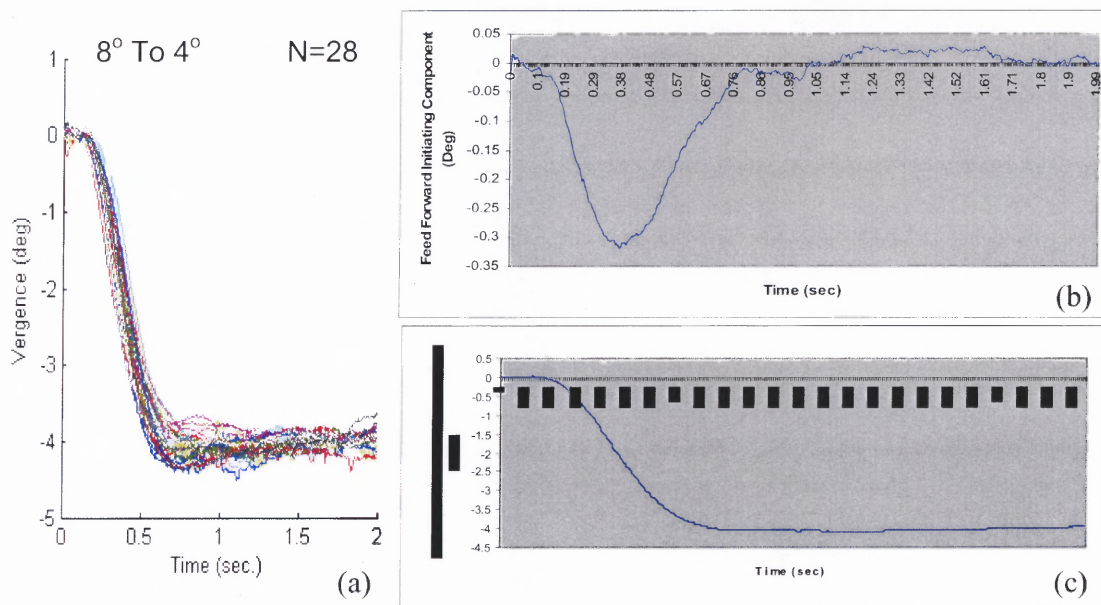
When recording an eye movement noninvasively, the transient and steady state components are recorded as a single signal as seen in Figure 1.6. When studying a system, it is important to study each component independently. A method needed to be established to isolate each component from the combined response. One technique used in isolating the two components in eye movement data is Independent Component Analysis (ICA).



**Figure 1.6** A four degree convergence response.

The ICA algorithm requires several different signals which represent a combination of sources [28, 29]. The measurements acquired from each signal arise from different physical locations [28, 29], but with eye movement data this is not possible. To accommodate for this requirement, multiple eye movement responses are collected and passed into ICA's mathematical algorithm. ICA is an effective algorithm for this application because it is assumed the initiating and sustaining component arise from different neural processes [28, 29]. The advantage is that the responses gathered from the eye movements will have some variability which ICA will be able to exploit when looking for the independent components [28, 29].

The algorithm proceeds to blindly separate the responses into its independent components. The assumption ICA makes is that in the combined response the variables are statistically independent and have a nongaussian distribution [28, 29]. The algorithm optimizes the nongaussian nature of the data to attain the independent components [28, 29]. In the case of a vergence movement, the response is decomposed into the initiating and sustaining components. Figure 1.7 displays one example of a four degree divergence eye movement in which the combined responses were decomposed into their independent components. Figure 1.7(a) shows the combined response, 1.7(b) shows the initiating component, and 1.7(c) shows the sustaining component.



**Figure 1.7** (a) Graph of 28 responses collected from one subject conducting 4° divergence eye movements. An independent component analysis was completed on these responses to acquire the two independent components. (b) Initiating Component (c) Sustaining Component.

### 1.2.6 Goals

Open-loop stimulation can be a valuable tool in the study of vergence eye movements. The LabVIEW program created presents the subject with a stimulus that allows the transient, open-loop, portion to be isolated and data collected from the vergence eye movement. Future work will include obtaining several responses stimulated by this open-loop code and performing an ICA decomposition on the data. It is hypothesized that a single source will be present.



## CHAPTER 2

### INSTRUMENTATION METHODOLOGY

The purpose of this thesis was to develop software code to allow for the presentation of a stimulus to conduct an experiment which would be able to obtain the transient, open-loop, component of a vergence eye movement and record the eye movement to understand the underlying neuronal behavior. The goal was accomplished by developing the open-loop and three-point calibration code in LabVIEW 6i and integrating work done by previous students from the *Vision Research Program 3.0* [2] into *Open-Loop Program 1.0*.

The *Vision Research Program 3.0* [2] is able to control light-emitting diodes (LEDs) which induce vergence and saccade eye movements under real world target conditions, oscilloscopes for disparity vergence experiments or smooth pursuit, photic stimuli to produce an after image through flashes of light, and monitors for smooth pursuit, saccade and disparity vergence experiments. In addition, eye movement data are collected and stored into a file for offline analysis.

The equipment currently available in the Vision and Neural Engineering Laboratory is an infrared Skalar IRIS Limbus Tracking device model 6500, an eye movement monitor, which contains a total of nine light emitting diodes (LEDs) per eye sensor for eye movement experiments, a Bayonet Neill-Concelman (BNC) adapter, two analog oscilloscopes, a function generator model 4040A, two partially reflective mirrors, three computer monitors, and a Dell Optiplex GX240 Pentium IV computer with 1.70 GigaHertz (GHz) processor, 256 MB of RAM, Microsoft Windows XP Professional

2002, a three-output Matrox Parhelia™ 128 MB videocard, and a data acquisition (DAQ) board model 6024e series from National Instruments.

## 2.1 Visual Equipment Utilization

To run an open-loop experiment requires carefully controlled stimulus to each eye and for stimulus presentation either monitors or oscilloscopes must be utilized to present the target to the subject. The target must be able to move as the eyes move toward the target, keeping the error constant. Employing real world targets such as LEDs is not sufficient since it is not able to illuminate the entire space between the initial target position and the final target position necessary for an open-loop experiment.

Oscilloscopes, on the other hand, are a better choice than LEDs, but do not offer a full range in the amount of degrees the eyes may cover while tracking the target. When placed at a reasonable distance, they typically allow the eyes to move a maximum of ten degrees through the presentation of a target from the center of the scope to the outer margin of the scope. In addition, analog oscilloscopes have a tendency to drift and experience timing delays [8], while digital oscilloscopes experience flicker which would make them incapable for real time studies [8].

The best option to display the target in open-loop experiments is to use computer monitors. Monitors offer greater freedom in terms of the range of movement the eyes may make. They are a digital solution at a tenth of the cost of digital oscilloscopes [8].

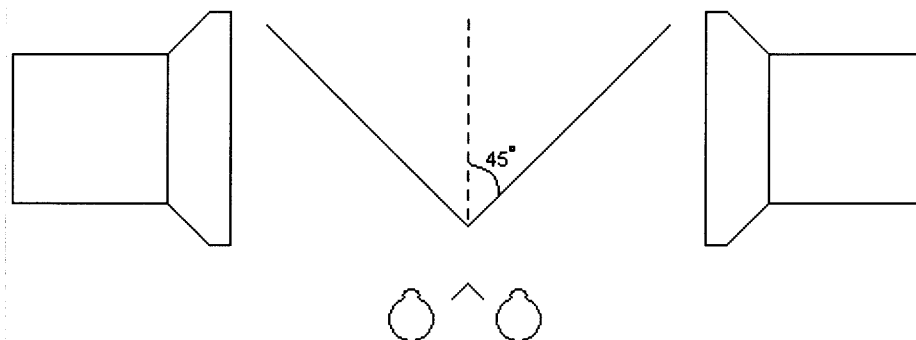
### 2.1.1 Monitors and Mirrors

The monitors used in the lab were fairly inexpensive and allowed for the stimulus to be presented on the visual display when placed in a haploscope configuration. A haploscope configuration consists of two monitors on either side of two mirrors. The monitors each project a target onto the mirrors to be seen on by the subject. The monitors of choice were the ViewSonic model E70fb 17 inch (16 inch viewable) display. It is a flat screen color cathode ray tube (CRT) monitor with a .21 millimeter horizontal dot pitch and a .25 millimeter diagonal dot pitch [32]. The optimal resolution for the monitor is 1024 x 768 at 87 Hertz (Hz) flicker free refresh rate. Incorporated within the monitor is PerfectFlat<sup>®</sup> Technology, an absolutely flat screen that reduces reflection, image distortion, and eye fatigue to provide realistically proportioned images, and SuperClear<sup>®</sup> technology, which delivers vivid colors and enhanced image quality. The monitors are MPR-II and ENERGY STAR<sup>®</sup> compliant, thus emitting safe radiation levels [32].

Presentation of the stimulus using computer monitors was essential to the open-loop experiment. They allowed the eyes to conduct greater degrees of movement compared to oscilloscopes, thus allowing an experiment to be run for an extended period of time. The monitors were placed on either side of the subject as Figure 2.1 shows. Between the monitors were a set of mirrors placed approximately 45° from the subject's line of sight. The monitors presented the stimulus on the mirrors, which was reflected to the subject.

The mirrors were adjusted into proper position using two alternating lines on the monitor between four degree and ten degree fusional fixation from the subject's midline. The alternating lines were then matched up to a board indicating where the four degree

and ten degree lines from the monitor should show. The target was aligned through the manipulation of the position of the mirrors and width and position of the lines on the monitors. The width and position values were saved and used by *Open-Loop Program 1.0*, since they indicated the location of four degree and ten degree real world targets.

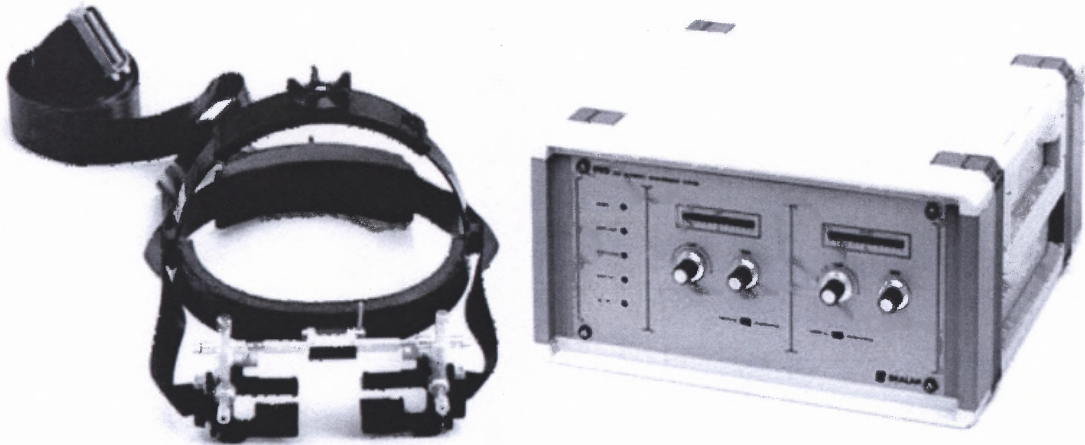


**Figure 2.1** Monitor and mirror setup for vergence experiments. The mirrors are placed approximately  $45^\circ$  from the subject's line of sight.

## 2.2 Peripheral Equipment Utilization

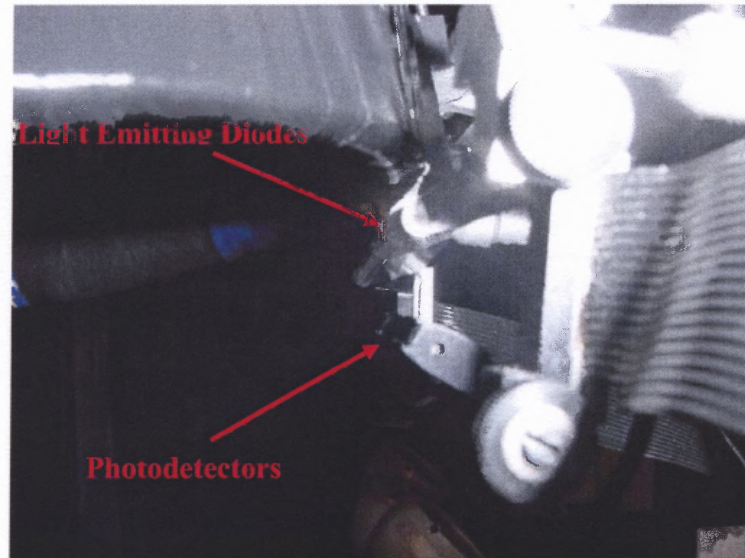
### 2.2.1 Eye Movement Tracker

Tracking eye movements with accuracy and precision is an important task when studying the oculomotor system. The limbus tracking device (left) and the control module (right) shown in Figure 2.2 are a non-invasive method utilized to record eye movement.



**Figure 2.2** Left is the Limbus Tracking device placed on the subject's head to record eye movements. Right is the electronic control module used to adjust the bias and gain of the incoming signal [10].

The limbus tracking device is able to record both horizontal and vertical eye movements. Although not simultaneously, this allows for experiments to be conducted on the saccades, smooth pursuit, vergence, vestibular, and optokinetic eye movements. The limbus tracking device emits an infrared (IR) light at a wavelength of 950 nm from its nine light-emitting diodes located above the eye which is reflected back by the eye and converted into an analog voltage by the nine photodetectors below the eye, as shown in Figure 2.3. The physiology of the eye is exploited by this method. The light reflected off the sclera is greater than that of the pupil and iris such that a difference can be detected in the position of the eye. The boundary of the iris and sclera on the nasal and temporal sides of the eye determine the reflection of the IR light through the use of a differential infrared system. The optimal resolution of the limbus tracking device is a 2 min. arc and has a linearity of  $\pm 25$  degrees [10]. Each eye has its own sensor to record the signal. The center of the bottom sensor of the eye piece contains a single white line which is aligned with the center of the pupil.

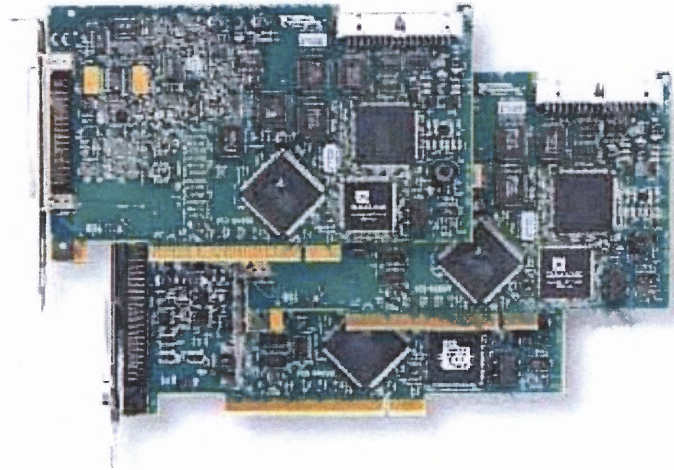


**Figure 2.3** Transmission and detection of light by limbus tracking device diodes (light emitters and phototransistors).

The voltage reading from the limbus tracking device is then transmitted into the control module where the gain and offset can be adjusted to control the amplification of the signal and ensure the signal does not saturate the system.

### **2.2.2 Data Acquisition Board**

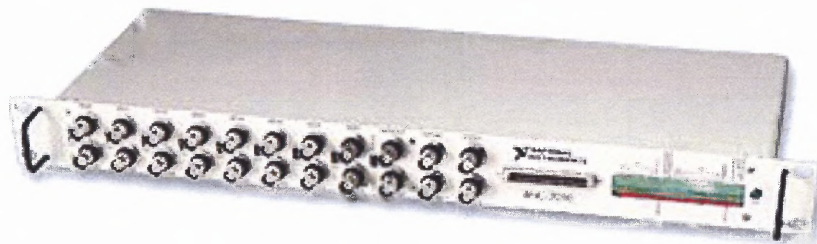
The Vision and Neural Engineering Laboratory operating computer has installed a Data Acquisition (DAQ) board, model 6024e, from National Instruments shown in Figure 2.4. The DAQ card is responsible for sampling analog signals as voltage values and transmitting analog and digital signals to external devices. The 12 bit DAQ card has a total of eight digital inputs or eight digital outputs that can read or write  $\pm 5$  volts along with two 12-bit analog outputs and sixteen channels of analog inputs.



**Figure 2.4** Data Acquisition Board to sample the data coming from the control module [22].

### 2.2.3 Bayonet Neill-Concelman (BNC) Adapter

The BNC adapter available in the Vision and Neural Engineering Laboratory is model BNC-2090. The BNC-2090 is a magnetically shielded, rack mountable adapter. It consists of 22 connectors and 28 spring terminals for connection with analog, digital, trigger, and counter/timer signals. For additional flexibility, any E-series DAQ device can be connected to this adapter from the front through a 68-pin connector. An image of the adapter is represented in Figure 2.5



**Figure 2.5** Bayonet Neill-Concelman (BNC) 2090 Adapter [23].

The eye movement tracker utilizes analog channel zero for left eye movement and analog channel one for right eye movement during data collection. The incoming data from the BNC-2090 is sampled by the DAQ card at a specified sampling rate determined by the operator in LabVIEW. The *Open-Loop Program 1.0* samples the data at a rate of 200 Hz, which is much greater than twice the Nyquist frequency for vergence eye movements.

#### **2.2.4 Trigger Button**

The trigger button is a device developed in the Vision and Neural Engineering Laboratory for use with eye movement experiments during data collection. A picture of the trigger button can be seen in Figure 2.6. The trigger button allows the subject to have control over the experiment when the stimulus is presented. Once the trigger button is pressed by the subject, the subject is asked not to blink for an amount of time ranging between 3 and 7 seconds. Within this experimental time, there is a 0.5 to 2 second random delay incorporated to avoid prediction by the subject when the target moves. During the remaining experimental time, the presentation of the stimulus and recording of the eye movement is completed and the loss of signal from the eyes is avoided since the subject knows not to blink or conduct any other form of eye or head movement than what is asked for by the presenting stimulus. After the particular run is complete, the subject is able to blink and begin the next run at his or her discretion.





**Figure 2.6** Trigger Button.

The trigger button was originally designed to occupy one of the digital bits in the BNC-2090. As experimentation advanced and the need for additional digital bit space became greater, the trigger button was moved to the analog TRIG1, channel 2, on the BNC-2090 by the graduate student Florence B. Chua. The code for this move was written in LabVIEW and incorporated into *Vision Research Program 3.0* (See Appendix A for a schematic of the trigger button) [2].

### **2.2.5 Function Generator**

The BK Precision, model 4040A, is a 20 MHz sweep function generator. It is able to generate waveforms, pulses, and a frequency sweep. For added convenience, there is a built in frequency counter to determine the frequency output. There is a five digit frequency counter which is able to display frequencies between 5 and 30 MHz. The core of the function generator is the voltage-controlled generator (VCG), which is able to

produce sine, square, and triangle waves with precision between .2 Hz and 20 MHz. A picture of the device can be seen in Figure 2.7.



**Figure 2.7** BK Precision Model 4040A, 20 MHz Sweep Function Generator [5].

The function generator is a very useful device when testing code developed for real time systems. The *Open-Loop Program 1.0* uses the function generator to calculate how fast it can analyze eye movements and present the next position of the target to the monitor. This will be further discussed in section 2.3.

### 2.3 Timing Enhancement

For real time systems, timing is a very important issue. The collection and analysis of eye movement data are not a speedy task and involves many procedures. The oculomotor system has a delay of ninety millisecond in the preprogrammed, feedforward, component. An algorithm had to be developed with optimization in mind, which would complete the

analysis of the eye movement and present the next target within milliseconds. The algorithm must work faster than the rate at which the brain analyzes and sends a signal to conduct an eye movement, if it is to keep the target ahead of the eyes with a fixed error. A timing analysis using the aliasing technique and the software timer was completed to ascertain the speed at which the open-loop code was running and presenting the position of the next stimulus target to the subject after a thorough analysis. Along with this, additional modifications to the program environment were completed to improve the speed as discussed later.

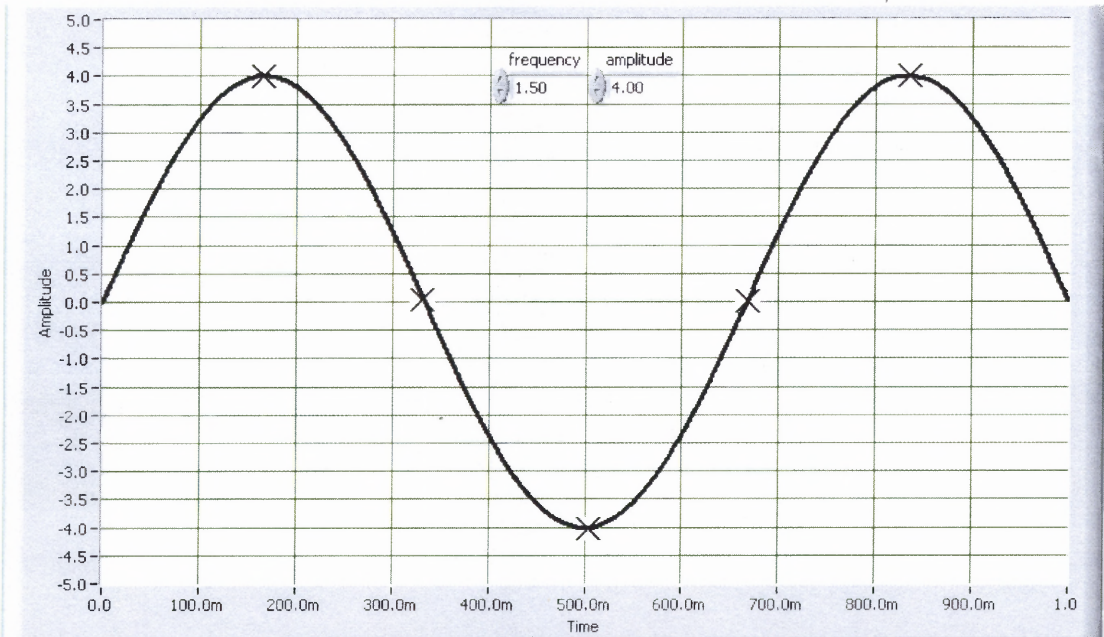
### **2.3.1 Aliasing**

Aliasing is a problem associated with under sampling, but can also be used to measure how fast a specific algorithm is running in real time. In this technique, a function generator is attached to one of the analog input channels in the BNC-2090. The function generator creates a sine wave at a specified frequency which is read in by the open-loop code, analyzed, and the next target presented on the monitors.

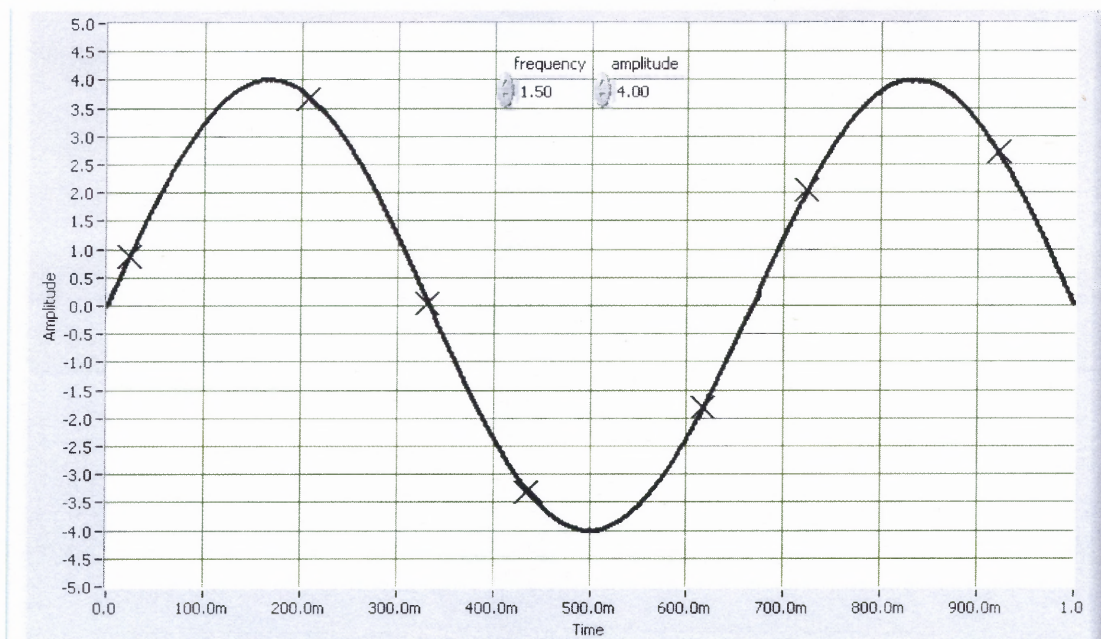
The method in which aliasing occurs depends on how fast the input data from the function generator is sampled. When the sine wave is sampled in the same location each time, this indicates aliasing taking place, Figure 2.8. During this condition, the sampling rate is in phase with the frequency of the sine wave. When this occurs in the open-loop mode, the target on the monitors oscillates between the corresponding positions. In case of figure 2.8, the target oscillates between three positions on the monitors and remains steady at each of these positions. However, when the sine wave is not sampled at the same positions each time, the rate at which the data is sampled from the sine wave is not

in phase with the frequency, Figure 2.9. This led to the target moving between multiple positions on the monitors.

To find how fast the code is running, the lowest frequency at which the target oscillates between two positions must be found. Taking the inverse of the frequency and multiplying it by 0.5 gives a value of how long the target is shown on the monitors twice. The value is for two positions, because the frequency oscillates the target between the two positions.



**Figure 2.8** A sinusoidal wave with a frequency of 1.5 Hz and an amplitude of 4.0. The open-loop code samples the sinusoid three times at three positions. The sampling of the sinusoid occurs in phase with the frequency, so the target appears on the monitors at only three positions.



**Figure 2.9** A sinusoidal wave with a frequency of 1.5 Hz and an amplitude of 4.0. The open-loop code samples the sinusoid at multiple positions. The sampling of the sinusoid occurs out of phase with the frequency, so the target appears on the monitors in multiple positions.

The inverse of the frequency represented in hertz (Hz) is taken to obtain a time in seconds. This value is multiplied by .05 because the code samples the data from the two troughs in the sine wave. The distance between the troughs is half a period, thus the inverted frequency is multiplied by 0.5. For instance, if the algorithm is found to display two targets at a rate of 10 Hz provided by the function generator for sampling, then the two targets are displayed in  $(1/10) \times (1/2) = .05$  seconds. Each target is being displayed in  $(.05) \times (1/2) = .025$  seconds or 25 milliseconds (msec). If the code had not sampled the data at exactly two peaks, then the target would oscillate between more than two locations.

### **2.3.2 Software Timer**

A software timer provided by LabVIEW was utilized to measure how fast on average each iteration of the for-loop takes in the open-loop code. This conducts a timing analysis on how fast the data is acquired, analyzed, and how long it takes to present the next target to the subject. This code is explained in greater detail in section 4.2.

### **2.3.3 External Programs Not Under LabVIEW Control**

To increase computer processor usage by LabVIEW, all nonessential programs running in the windows environment were shutdown. This enables LabVIEW to have priority of the processor and insures the real-time analysis does not have to wait in a queue for access to the processor.

## **2.4 Programming Languages**

The Vision and Neural Engineering Laboratory has employed the use of both LabVIEW and Matlab<sup>®</sup> for a real time presentation of the stimulus, the acquisition of data, and analysis. The *Vision Research Program 3.0* was written in LabVIEW and controlled the stimulus presented to the subjects such as LEDs, photic stimulators, monitors, and oscilloscopes. In addition, this program was responsible for data acquisition and saving the data with corresponding header information to a file so that the data could be analyzed offline. The analysis of the data was completed by a program previously written in Matlab.

### 2.4.1 LabVIEW

LabVIEW Version 6i is a visual programming language, which offers a diverse array of options to the programmer needed to conduct research. LabVIEW is an industry standard which allows advanced data acquisition and analysis. The *Vision Research Program 3.0* was written in LabVIEW and to keep consistency within the Vision and Neural Engineering Laboratory, this thesis utilized LabVIEW for *Open-Loop Program 1.0*. LabVIEW performed an integral part in collecting eye movement data and completing a real time analysis. The development and incorporation of the three-point calibration in LabVIEW was essential for the real time analysis. It was used to determine the precise location of the eyes so that LabVIEW could determine the next position the stimulus should be displayed within milliseconds.

### 2.4.2 Matlab<sup>®</sup>

Matlab<sup>®</sup> 6.1 was utilized for data analysis. Data analysis needed to be completed in a timely manner with accuracy in mind. Code written previously in Matlab and currently used by *Vision Research Program 3.0* was modified to work with data acquired with *Open-Loop Program 1.0*. The following code was added into the calibration section to assist in the acquisition of the three-point data, to draw the regression analysis, and convert the responses from voltages to degrees movement.

```

nu_cal_pts=3;

left_eye=fliplr(l_cal);
y=[left_eye(:,1,:) left_eye(:,2,:) left_eye(:,3,:)];
left_eye=fliplr(y);

right_eye=fliplr(r_cal);
y=[right_eye(:,1,:) right_eye(:,2,:) right_eye(:,3,:)];
right_eye=fliplr(y);

```

The current Matlab code uses a two point and five point calibration for data analysis when converting the voltage values into vergence position denoted in degree movement. The *Open-Loop Program 1.0* on the other hand utilizes a three-point calibration to accomplish this same task. A three-point calibration is more effective because calibration is completed before every run of the open-loop stimulus. Whereas a five point calibration fatigues the subject at a faster rate and becomes very tedious for data analysis within LabVIEW, a three-point calibration is sufficient and gives similar results. A two point calibration could not be used because this method does not collect the mean voltage value at two known degree positions during the execution of the LabVIEW code. The *Open-Loop Program 1.0* relies on the calibration data for a real time analysis, which is why the two point calibration was not used.

The *Vision Research Program 3.0* utilizes the five point code in Matlab. The five point code assumes it is receiving five voltage values and five degree positions. Two of the voltage and position values are saved as dummy values in the open-loop code to keep consistency between the data acquisition programs. The altered code removes the dummy values and passes only the three voltage and position points to the five point data analysis code, which then converts the response acquired from the left and right eye into degrees. Each eye is analyzed independently.

Within the code above, the function `fliplr` takes the array and flips it. For instance the array `[1 2 3 4]` would become `[4 3 2 1]` after passing through `fliplr`. The three voltage and position values are taken and passed into a dummy variable. This dummy variable is then equated to the original variable read in by the five point calibration for analysis.



Another aspect that needed to be taken into account was the noise in the incoming signal. This noise could be from instrumentation, physiological variation in the eye movement, or another unknown source. To remove the noise, a filtering algorithm provided by Matlab<sup>®</sup> was already incorporated into the Vision and Neural Engineering Matlab<sup>®</sup> code, but was disabled during analysis. The filtering was re-enabled for *Open-Loop Program 1.0*. The Matlab code made use of the Butterworth digital filter. The Butterworth filter offers a variety of options from allowing lowpass, bandpass, highpass, and bandstop filtering. Its algorithm provides enhanced smoothness in the curve. For the open-loop code, a lowpass filter is administered to remove noise from the signal before moving through the three-point calibration. The code is as follows:

```
order = 8;
CutOfFreq=.2;
[posb posa] = butter(order, CutOfFreq);
response(:,1) = filtfilt(posb, posa, response(:,1));
response(:,2) = filtfilt(posb, posa, response(:,2));
```

```
posa has the values: [1 -4.7845 10.4450 -13.4577 11.1293 -6.0253 2.0793 -0.4172 0.0377]
posb has the values: [0 0.0002 0.0007 0.0013 0.0017 0.0013 0.0007 0.0002 0]
```

The cutoff frequency is the frequency at which the magnitude of the response is 0.707. This cutoff frequency is a value which ranges between 0 and 1, where 1 corresponds to  $\pi$  radians per sample, the Nyquist frequency [19]. After determining *posa* and *posb* from the order and cut off frequency, the values of *posa* and *posb* are passed into the filter which acts upon the response from the left eye and right eye independently. The function *filtfilt* is provided by Matlab and performs filtering by processing the data bidirectionally, forward and back. This gives the resulting data a zero-phase distortion and the filter order is doubled [19].

## **2.5 Integration and Usage of Code Completed by Others**

Many people have had an integral part in the development of the *Vision Research Program 3.0* [2]. Several components essential for eye movement stimulus presentation and initial setup of the limbus tracking device were incorporated into the *Open-Loop Program 1.0*. The initial setup portion allows the operator to adjust the limbus tracking device for enhanced and accurate signal acquisition [2]. The stimulus presentation displays the target to the subject on computer monitors [8].

### **2.5.1 Monitor Stimulus Display**

The monitor display stimulus code [8] was written to display a target to the subject on computer monitors available in the Vision and Neural Engineering Laboratory. This system is able to stimulate vergence, saccade, and smooth pursuit eye movements on a digital panel, depending on where the monitors are placed and which condition of the monitor display stimulus code is selected. A schematic of the code can be seen in Appendix B. The stimulus being displayed to the subject could be a single line or a variety of shapes, depending on what the researcher requires for the experiment. In the case of an open-loop experiment, a single vertical line was utilized. The dimensions of the target, such as height and width, were also under operator control.

### **2.5.2 Eye Movement Tracker Setup**

The eye movement tracker setup is completed through the code embedded in the initialization portion of the *Vision Research Program 3.0*. Proper placement, orientation, and signal acquisition of the limbus tracking device is completed at this point and is essential to acquiring coherent eye movement data for analysis. This code was written by

several people in the Vision and Neural Engineering Laboratory [2]. The initialization code from the *Vision Research Program 3.0* was added to *Open-Loop Program 1.0* and modified to suit its needs. The initialization portion of the *Open-Loop Program 1.0* provides continuous data acquisition of eye movements for adjustment of the limbus tracking device to acquire a coherent signal from both eyes. The target oscillates between the two known positions stimulating eye movements, which are detected by the limbus tracking device and transmitted to the computer. The operator is then able to determine the necessary adjustments, which need to be made to attain a stronger signal. The position of the targets can also be stopped in their location for further control over what the subject is viewing.

During the eye movement tracker setup stage, the pixel positions of where the four degree position resides and where the ten degree position resides were determined. This is essential when the operator enters the stimulus in degrees, but the computer needs to translate this information into pixels. A conversion code was written which is further explained in section 2.5.3. The code for assigning a pixel position to the four degree position and the ten degree position was written by Florence B. Chua [8]. This was accomplished by having two real world targets position at a four degree and ten degree vergence position and allowing a line to oscillate between the known targets on the monitors. The position of the lines, known as the offset, and the width between the oscillating lines, called the gain, were adjusted to match the real world targets. Along with this, the two partially reflective mirrors between the monitors could be adjusted for greater accuracy in the location of the target. (See Appendix C for the front panel display of the initialization section and the code.)

### 2.5.3 Vergence Degree to Pixel Calculation

When presenting the stimulus to the subject, the operator needs to understand the stimulus in terms of degrees, but the monitors do not understand at what location the specified degree refers to on a monitor screen. This led to the creation of a conversion factor from degrees to pixel position for the monitors. This method was originally used in the *Vision Research Program 3.0* and now is being utilized by *Open-Loop Program 1.0*.

With the assistance of the initialization code, the vergence degree to pixel code knew at what degree the corresponding pixel value on the monitor was. Code written by Florence B. Chua [8] allowed for a linear line of pixels vs. degrees to be determined in order to calculate the rest of the degree to pixel positions. (See Appendix D for the conversion code) The pixels for the two known degrees and the corresponding two degree values were passed throughout the *Open-Loop Program 1.0* and the conversion program called upon each time a pixel position was needed, thus simplifying operator work by letting the computer determine the pixel value for the given degree.

## 2.6 Open-Loop Program 1.0 Structure

The *Open-Loop Program 1.0* flow chart is shown in Figure 2.10. The program begins first with variable initialization in which all the parameters are set for the open-loop code to proceed with the necessary information. Next, the eye movement monitor is properly placed on the subject during setup and target values are calibrated with known real world targets where the corresponding pixel values are calculated, which was discussed in subsections 2.5.2 and 2.5.3.

The third segment is the three-point calibration essential for real time data analysis to determine the exact position of the eyes during execution of the open-loop code. The fourth item in the list is the stimulus selection. One stimulus will be selected by the program from a list of multiple stimuli, which the operator specifies during variable initialization.

The next segment is the crucial portion of the code, designated as open-loop. This segment presents the initial target to the subject, completes a real time data analysis while moving the target with the movement of the eyes, and collects the eye movements for storage into a file for offline analysis. The seventh segment determines whether to go back to three-point calibration, stimulus selection, or to exit the program. The selection criteria are controlled by the operator and not by the subject or the open-loop code. The last stage is to complete all the remaining tasks and finally exit the program successfully.

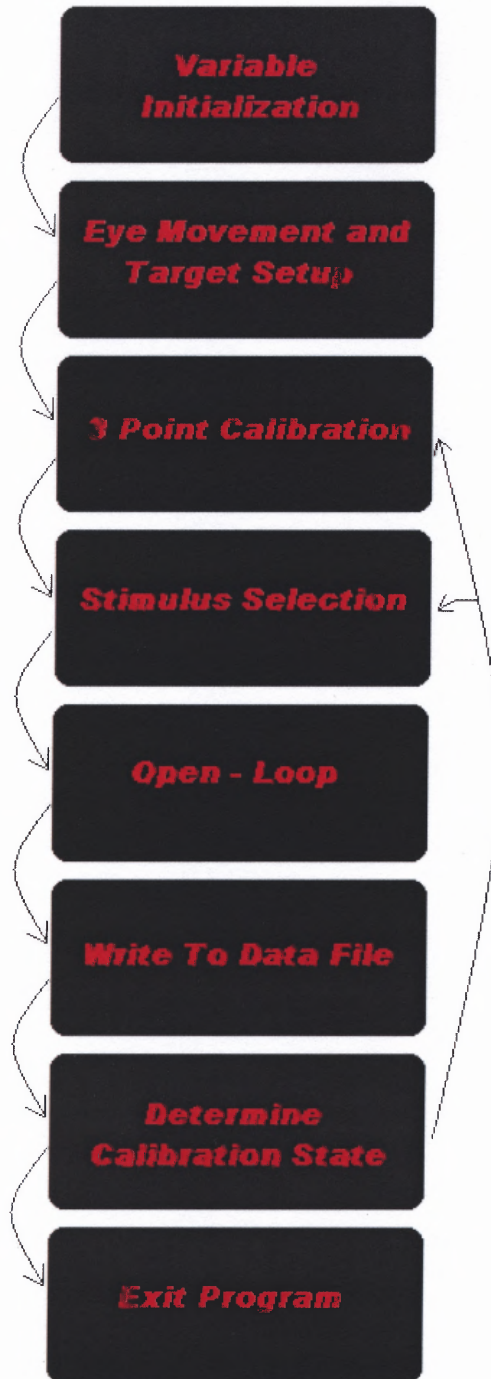


Figure 2.10 Order in which *Open-Loop Program 1.0* executes from start to finish.

## CHAPTER 3

### THREE-POINT CALIBRATION METHODOLOGY

The three-point calibration is essential during the open-loop procedure and for analysis in Matlab and was created for this study by this thesis. The calibration is responsible for converting the voltage values from the eye movement tracker into degrees. During the open-loop procedure, this task is important because the data from the calibration is used for determining the current position of each eye. Matlab utilizes the calibration data to convert the entire response from both eyes into degrees of movement. This conversion is important because stimuli are typically given in degrees not voltage values.

Calibration is performed by presenting the subject with the three known degree positions, one at a time and recording the incoming signal from the eye movement tracker. The incoming data from each eye are collected and analyzed independently for a designated amount of time and sampling rate determined by the operator. In an open-loop experiment, the three-point calibration presents each target to the subject for two seconds and samples at a rate of 200 Hz. The calibration analysis averages the steady state portion of the response which is the final 1.5 seconds of the eye movement response to determine the mean voltage which correlates to a known target that is a known degree change in fixation. A graph is then drawn of the voltages vs. the known target in degrees. The last step is to complete a linear regression analysis, which obtains the y-intercept and slope for the calibration curve to correlation input response in volts to eye movement response in degrees. The equation of the regression line is used to convert the incoming data from the eye movement tracker into degrees of movement.

### 3.1 Front Panel and Structure

The front panel, shown in Figure 3.1, displays two graphs and a table. The large graph at the top, called a waveform graph, displays the data acquired during the two second data acquisition period. The graph displays the data collected for each target presented to the subject. Once data acquisition has completed for all three targets and the mean and standard deviation are calculated, the means are plotted on the second graph, called an XY-Plot. Along with this, the mean and standard deviation are shown in the table for each eye with the corresponding degree position.

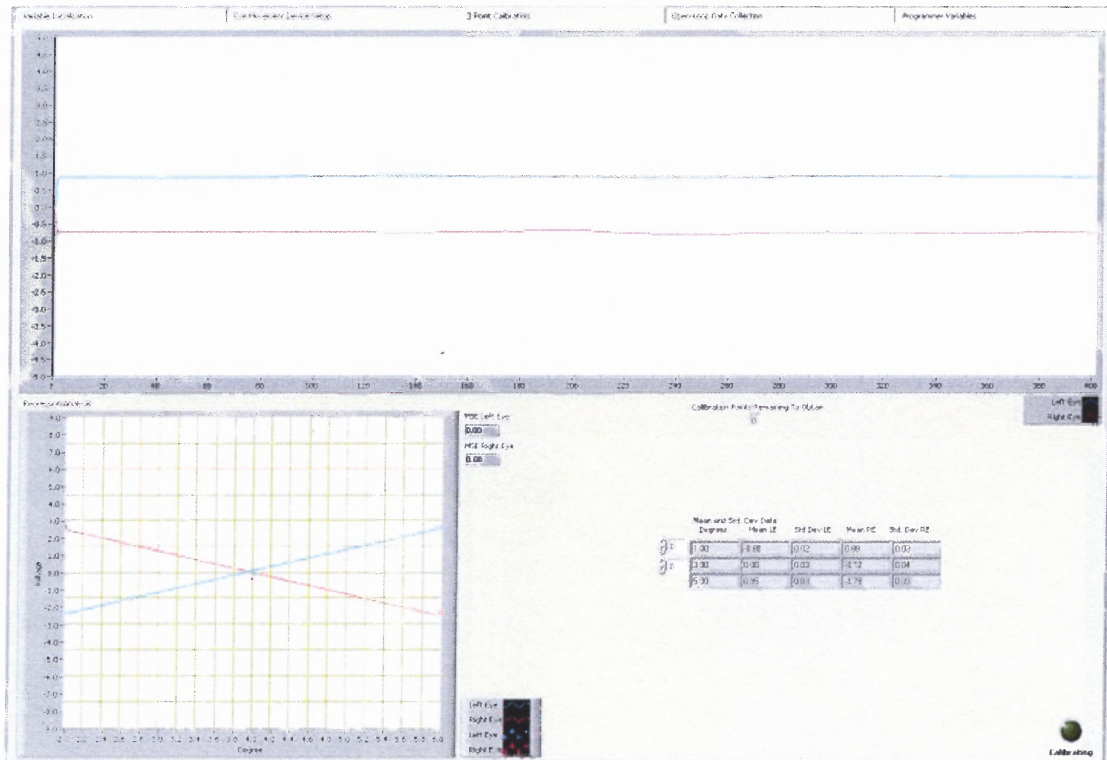


Figure 3.1 Three-point calibration front panel.



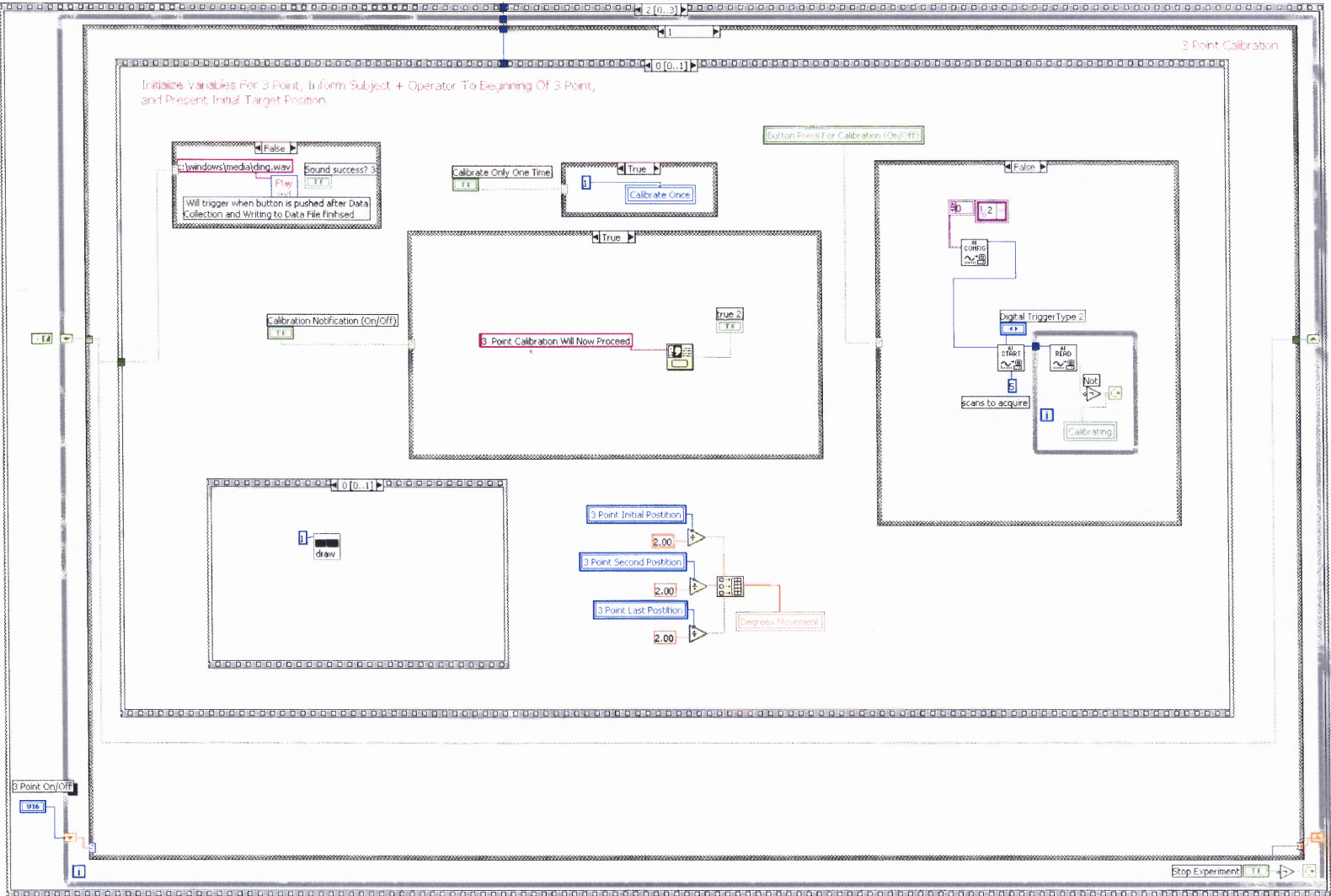
The code for the three-point calibration follows a specific sequence. Each sequence is given the responsibility of a specific task. There are many aspects important to the three-point. One of the most important is the detection of when the subject blinks and if a blink occurs then data for the target presented to the subject must be reacquired. As the design and structure of the code is explained in greater detail, the importance of each section will be understood. The design of the three-point calibration program is in a two sequence structure developed in LabVIEW 6i. Sequence one is responsible for variable and parameter initialization to allow the three-point code to work flawlessly and sequence two is responsible for data acquisition and analysis.

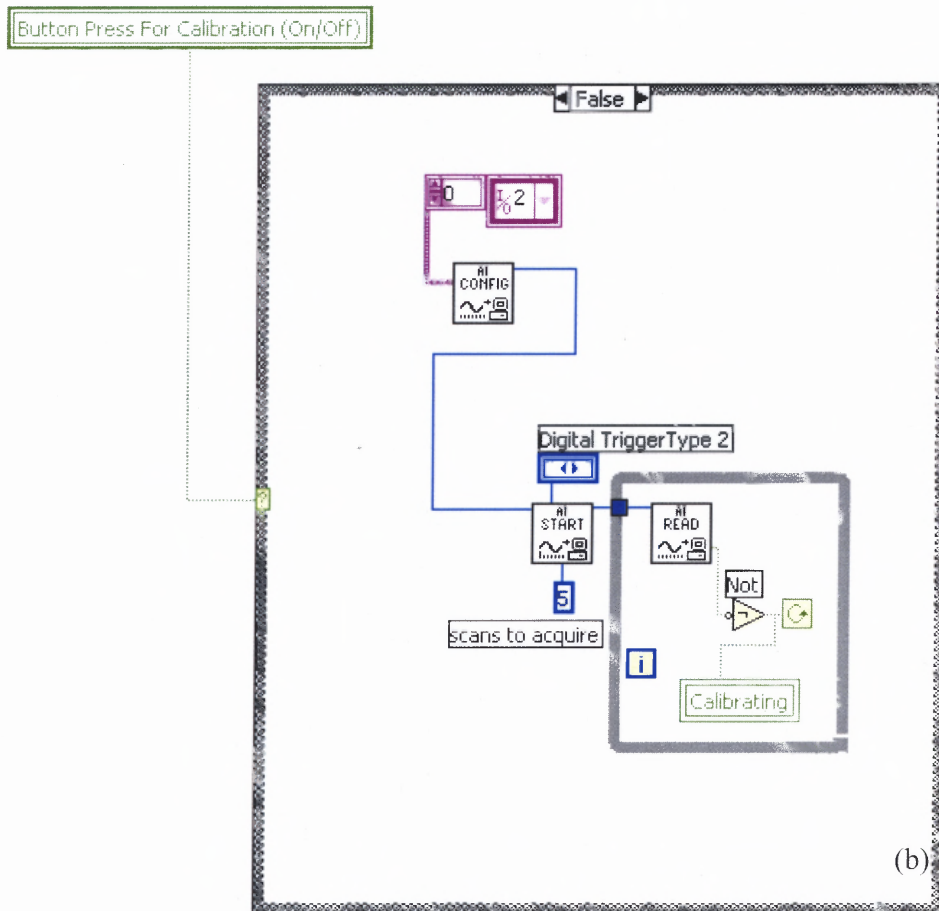
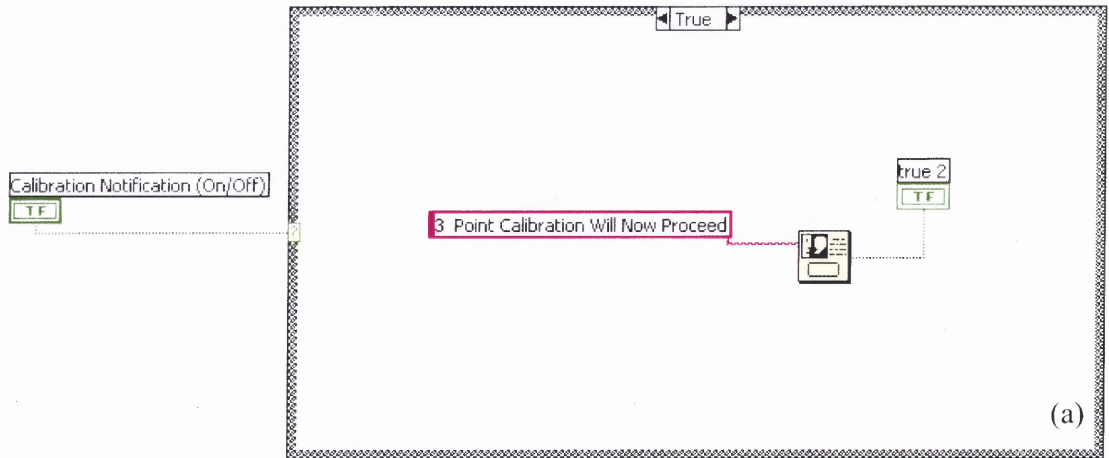
## **3.2 Sequence One**

### **3.2.1 Subject Notification of Three-Point Start**

In the first sequence shown in Figure 3.2, the primary objective is to inform the subject the three-point calibration will proceed. This is accomplished through two methods, a tone is heard by the subject or the operator can inform the subject verbally. The operator also has the option of having a popup window appear to inform both the subject and him or her of the beginning of a three-point calibration. The code for the popup can be seen in Figure 3.3(a). The popup window is controlled through a switch placed in the initialization portion of *Open-Loop Program 1.0* and is explained in greater detail in section 5.1. When the button is set to on, the operator receives the popup dialogue box.

Figure 3.2 Top view of sequence one of the three-point calibration.





**Figure 3.3** (a) Popup notification code for three-point calibration. (b) Trigger button for three-point calibration and determine if option one or option two should follow.

### **3.2.2 Trigger Button Control and Options**

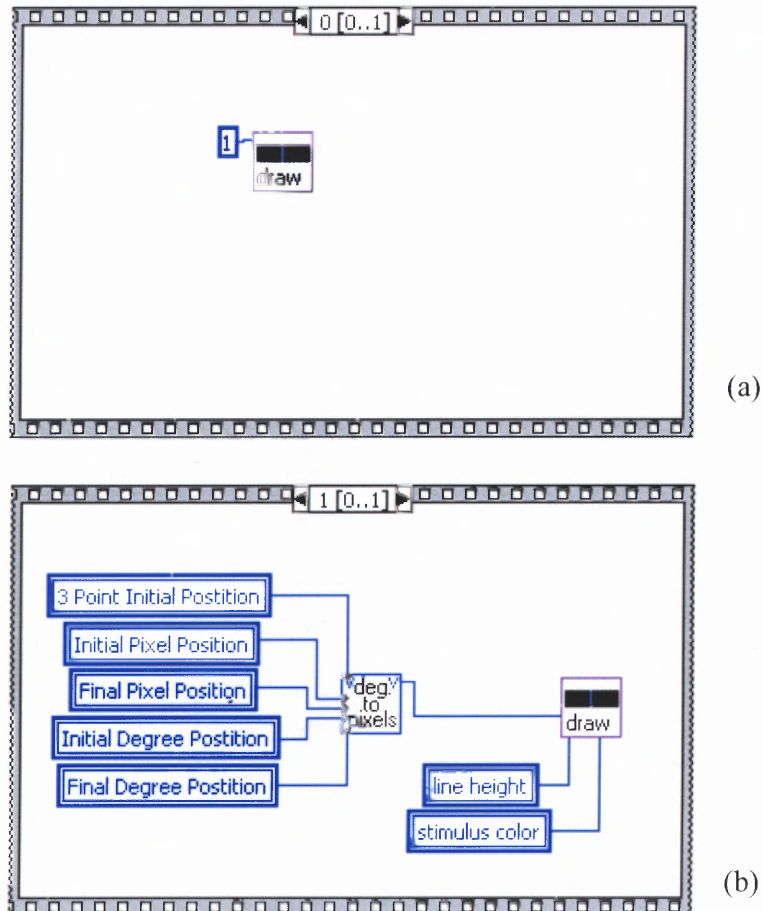
The trigger button is essential for allowing the subject to have control when three-point data acquisition will begin. This allows the subject to rest if needed and relax his / her eyes. The subject is given two options with the three-point trigger button. Option one is the subject can press the trigger button once, and all three targets will be presented one by one in sequence. This allows for calibration to proceed in rapid succession and the subject experiences less fatigue. Option two allows the subject to press the trigger button before each presentation of the target giving more control to the subject. Both options are controlled through a switch connected to the initialization panel. When it is set to no, option one is activated (no trigger pauses between calibration stimuli); otherwise it defaults to option two throughout the three-point calibration.

When option one is activated, sequence one contains the code for the single push button required to initiate the presentation of the targets, Figure 3.3(b). When option two is activated, sequence two disables the trigger button code in sequence one and enables the trigger button code in sequence two, the data acquisition and analysis sequence thus requiring the subject to push the trigger to initiate data acquisition of the next calibration target.

### **3.2.3 Presentation of Initial Target**

The next portion of this sequence is responsible for clearing the screen and presenting the initial target position to the subject, Figure 3.4. This portion of the three-point calibration follows a two sequence structure. The first sequence, Figure 3.4(a), sends a signal to the monitor to clear all data and graphics from the screen. The second sequence, Figure 3.4 (b), uses the data acquired from the eye movement monitor tracker stage to draw the first

target, which the subject is asked to fuse. In the three-point calibration, this initial position is at two degrees and is controlled by the three-point *Initial Position* variable located on the initial setup page.

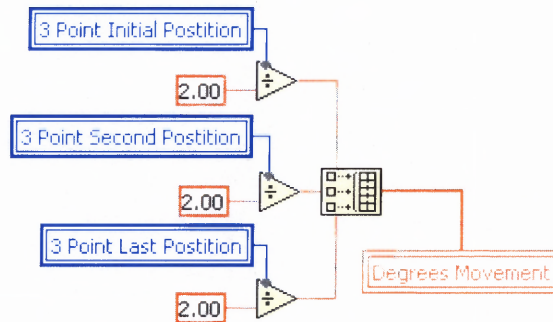


**Figure 3.4** The following two sequences (a) clears the screen and (b) presents the initial target for the subject to fixate on.

### 3.2.4 Array of Targets for Analysis

The final part of the first sequence is to create an array, which consists of the three targets in degrees. This is completed as shown in Figure 3.5. The *3 point initial position*, *3 point second position*, and the *3 point final position* variables contain degree information for both eyes together combined. The reason for dividing the three degree values by two

is to obtain the degree positions for each eye (not the combined response) and save this information into the array for use with the regression analysis completed in sequence two of the three-point calibration.

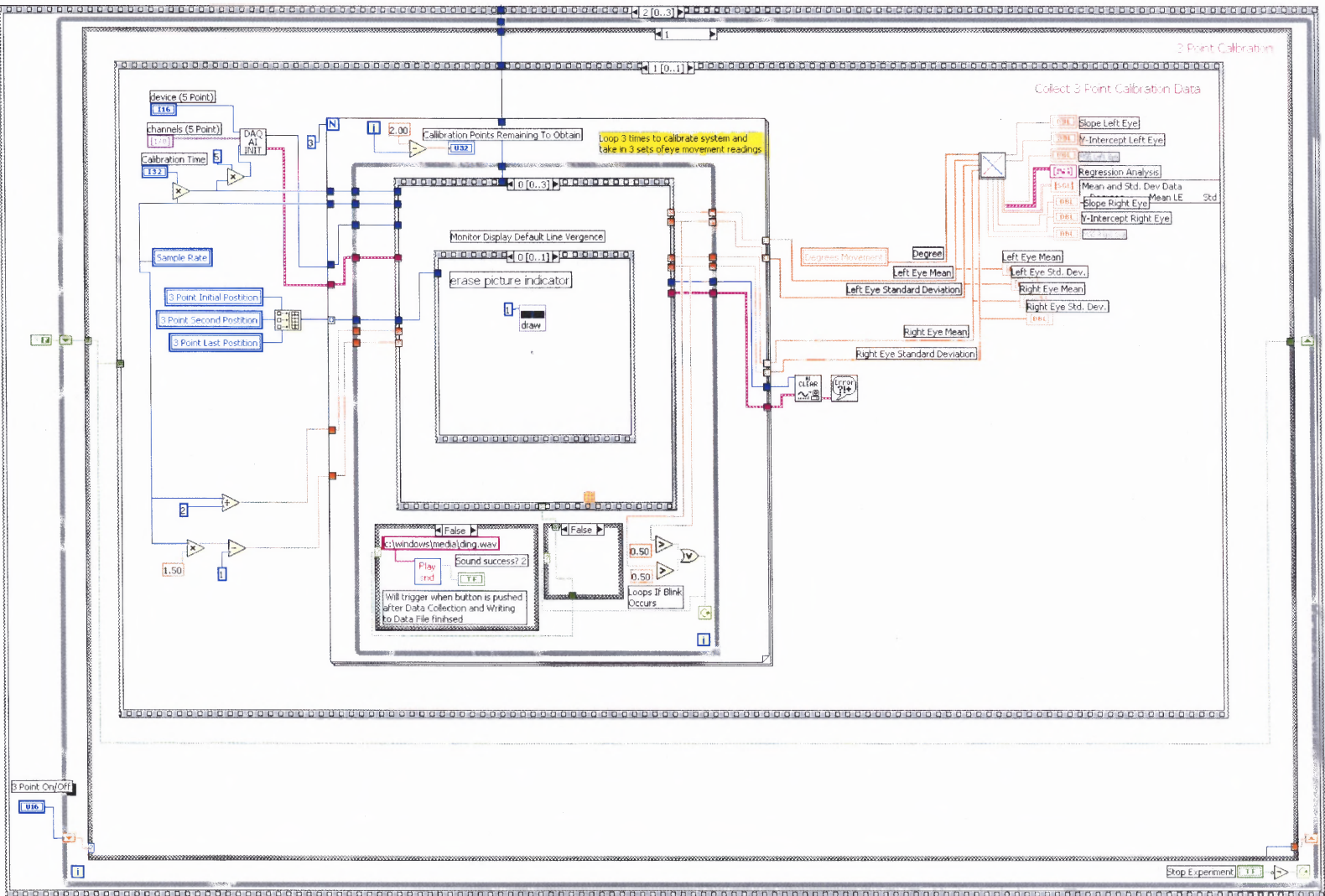


**Figure 3.5** Initialize array with three targets in degrees.

### 3.3 Sequence Two

Sequence two is the main portion of the three-point calibration responsible for the presentation of the target to the subject, data acquisition, and a linear regression analysis. A general view and structure of the code can be seen Figure 3.6. There is a large nested for-while loop, which contains another sequence structure each responsible for its own task which will be explained further in subsections 3.3.5 – 3.3.8. The code appears to be very complex, but is very simple and structural.

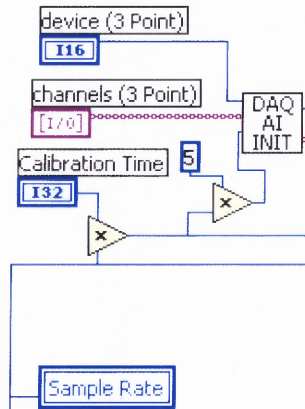
Figure 3.6 Top view of sequence two of the three-point calibration.



### 3.3.1 Data Acquisition (DAQ) Components

The code first begins by initializing the DAQ components for data acquisition. This is an important step because data are to be digitized from the eye movement monitoring device. The DAQ initialization subvi is discussed in subsection 5.4.1. For data acquisition to execute properly, it must receive three parameters to the DAQ AI INIT subvi in Figure 3.7. The top parameter is the device number informing the acquisition components from which device to acquire data. The second parameter informs the DAQ components from which channel the information will be acquired. The open-loop code uses channels zero and one simultaneously to acquire the data from the eye movement tracker. Each channel acquires data from each eye independently and stores it into a two dimensional array. Channel zero collects left eye movement signal and channel one collects right eye movement signal. The last parameter creates the buffer size where the data will be temporarily stored until later retrieved by the open loop program. The buffer is created using the calibration time, set to two seconds for open-loop, multiplied by the sampling rate, which is 200 Hz and multiplied by five. The last multiplication of five creates additional space in the buffer to allow greater speed and accuracy in placing data into the buffer. This is also the value LabVIEW technicians suggest the buffer be multiplied by for optimal use of the buffer size [17].





**Figure 3.7** DAQ initialization.

### 3.3.2 Array of Targets For Monitor

The next stage initializes the array with degree positions to be shown to the subject. This portion is redundant since it was completed earlier in sequence one, but the values in sequence one are used for the analysis of each eye independently. The values inserted into the following array, Figure 3.8, are used to present the target to the subject, which is seen by both eyes together.

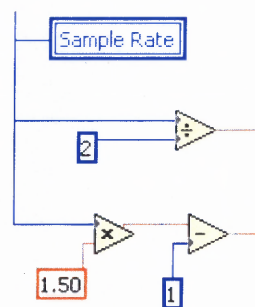


**Figure 3.8** Array initialization with targets present to subject during three-point.

### 3.3.3 Location to Analyze Eye Movement Data for Mean and Standard Deviation

When collecting the eye movement data at each degree position, the three-point calibration is given the range of points the analysis uses for the mean and standard deviation calculations. The code in Figure 3.9 accomplished this task. The sample rate of 200 Hz was divided by two stating where the data analysis should start from in terms

of seconds. Thus if 200 samples are collected in one second the data to be used during the analysis should be at 0.5 seconds or begin at sampled data point 100. The sampling rate was then multiplied by 1.5 and subtracted by one stating that a total of 300 points should be collected. All put together, this meant the data is analyzed from 0.5 seconds to two seconds for each target presented to the subject. These values are then passed into the three-point calibration loop, sequence three.

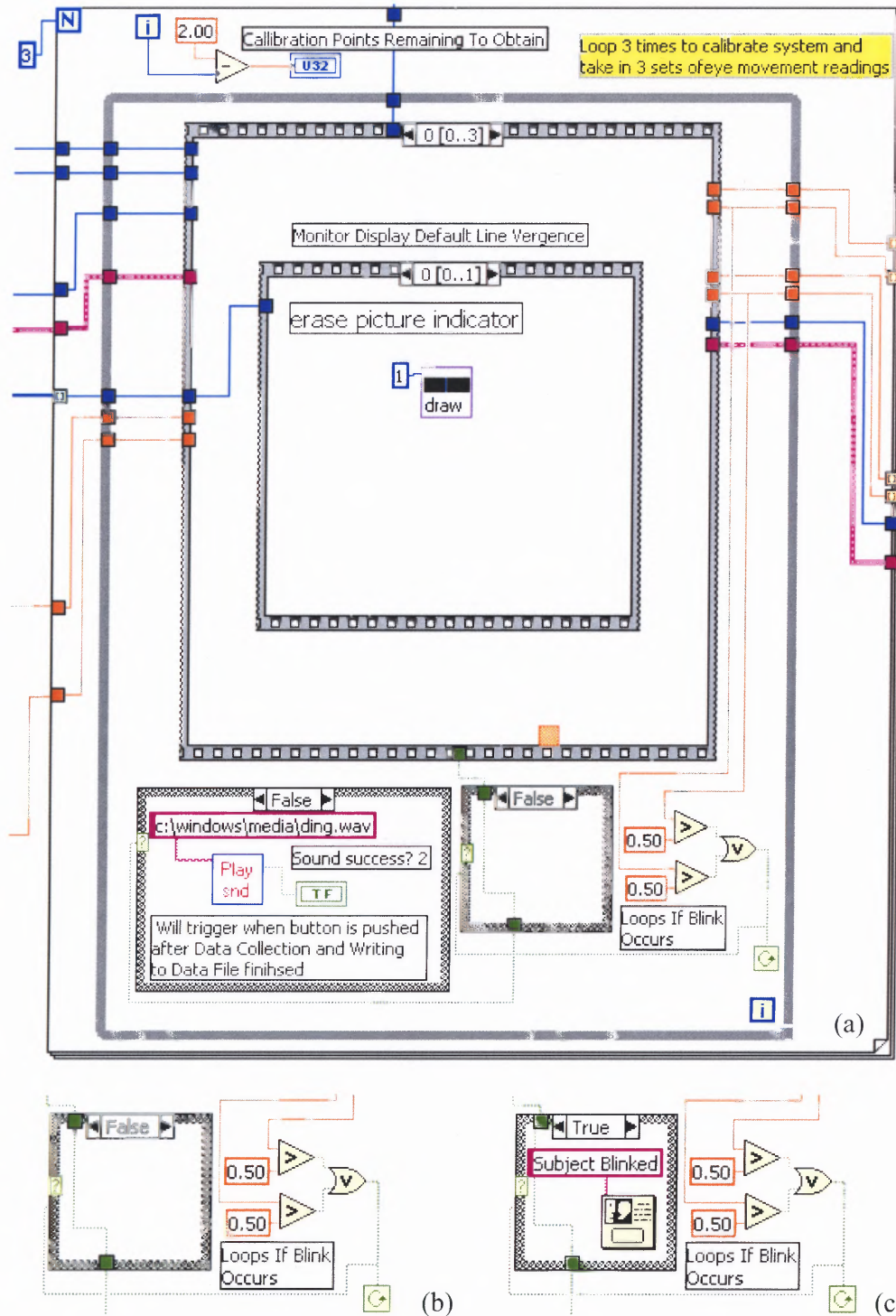


**Figure 3.9** Code determining where data should be analyzed for mean and standard deviation during each target presentation to the subject.

### 3.3.4 Large For-Loop With Blink Recognition

The next section of the code is the large for-loop, Figure 3.10(a). This for-loop is responsible for presenting the target to the subject, collecting eye movement data, calculating the mean and standard deviation, and determining if the subject blinked and if so to repeat the same target data collection. The for-loop loops for a total of three runs, once for each target. Within the for-loop is a while-loop responsible for redoing the target run if a blink occurred. A blink is recognized by the standard deviation calculated from the incoming data from the DAQ card. When the standard deviation is greater than 0.50, the program interprets it as excess variability in the signal most likely due to a blink. The standard deviation from each eye is checked to see if the value is greater than

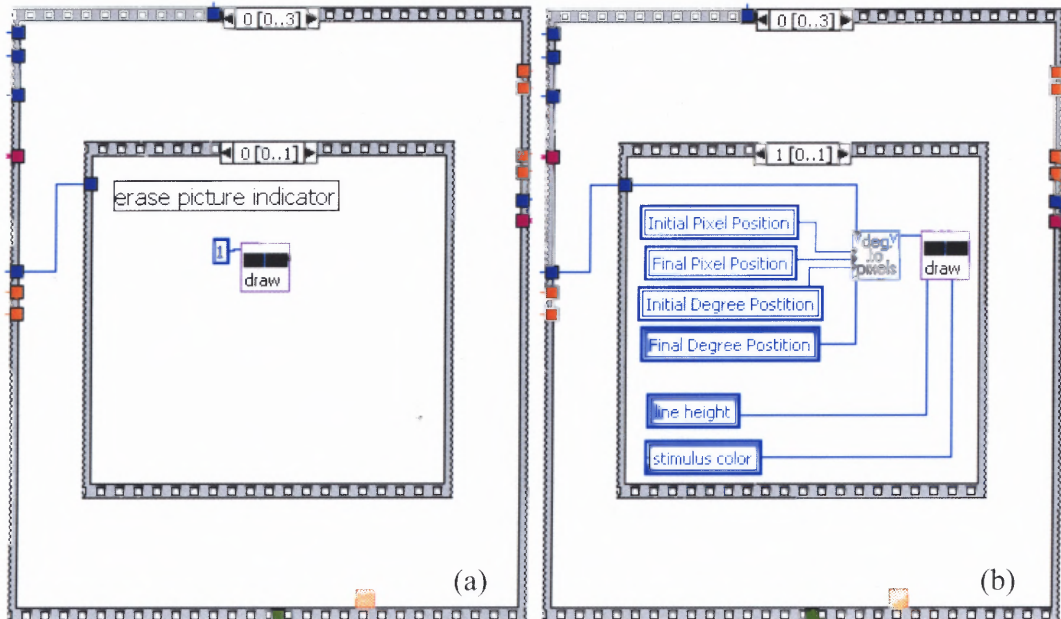
0.50 by the comparator. If either one of the eyes is greater, a true value is sent to the case statement, Figure 3.10(b) informing the operator the subject has blinked and the run must be redone. The true value is also sent to the while loop to present the same target for data acquisition of the eye movement to the subject. If a false value is sent from the comparator, Figure 3.10(c), the while loop does not present the same target and the subject is able to move on to the next target for data acquisition. The method of acquiring the mean and standard deviation will be discussed further in subsection 3.3.8.



**Figure 3.10** Blink recognition code. (a) Overall view of for-loop and structure within. (b) False statement displays nothing in the situation of no blink. (c) The operator is informed the subject has blinked and the same target will be presented to the subject again.

### 3.3.5 For-Loop Sequence Zero

When the for-loop begins, it must proceed through four sequences each assigned a specific task. Within sequence zero, the graphics displayed to the monitors are cleared as shown in Figure 3.11(a). The next part of that sequence, Figure 3.11(b) then displays the next stimulus to be shown to the subject. The stimulus is selected from the array of degrees and that particular one is chosen by the for-loop counter. This takes place a total of three times, once for each of the degree positions used by the three-point calibration.

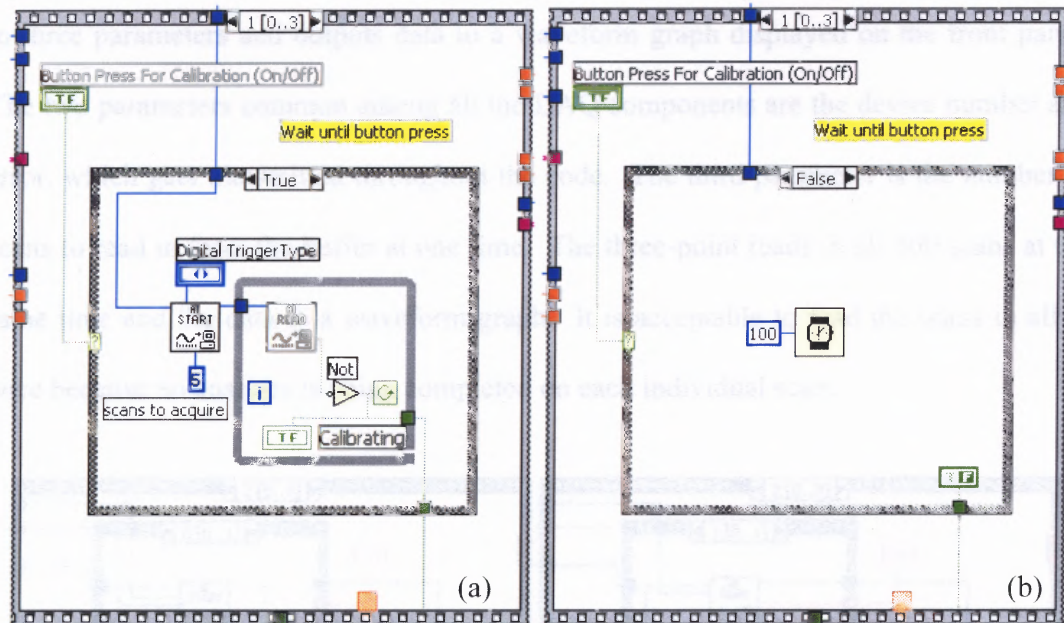


**Figure 3.11** (a) Clears the monitor of all data and figures. (b) Presentation of the targets (initial, second, and final) in sequence to the subject determined by the location in the for-loop.

### 3.3.6 Trigger Control

The next sequence in the for-loop, sequence one, is responsible for how much subject control there is when data acquisition begins. The operator can give the subject two options. The first option, the one that is preferred because it decreases the amount of

time spent in calibration, is to have the computer control when data acquisition takes place between the three stimuli. The second option is to allow the subject to use the trigger button to begin data acquisition. Option one is the best because it flows through calibration very rapidly and fatigues the subject less to allow for longer experimentation and increased data collection rather than time spent on calibration. Both options are controlled through a switch on the front panel in the initial setup stage called *Button Press For Calibration (On/Off)*. When the switch is set to off (false to the computer), option one is selected, otherwise option two is. Figure 3.12 displays the case statement in which option, one or two, can be selected depending on the button press variable. When option two, Figure 3.12(a), is selected, the button press variable is activated, which waits in a loop until the subject presses the trigger button. Alternately, if option one is selected, Figure 3.12 (b), a delay is held to allow the subject to fixate on the new target before data acquisition proceeds.



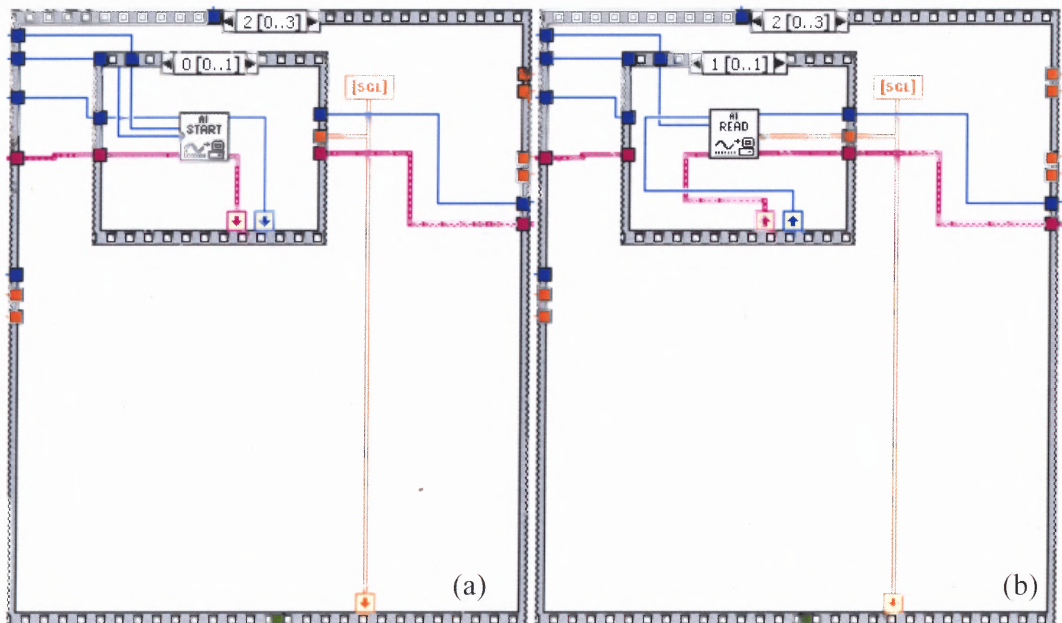
**Figure 3.12** Determines whether subject has control of when data acquisition begins or subject has control. (a) When set to on, true in Boolean, the subject has control. (b) When set to off, false in Boolean, the computer has control.

### 3.3.7 Data Acquisition Read and Start

Sequence two in the for-loop is dedicated to data acquisition. The first step, Figure 3.13(a) is responsible for starting the data acquisition and storing the incoming data into the buffer. It takes in four inputs. The first is the device number to tell which DAQ card to utilize during acquisition. The second is the total number of scans to take in, which is 400 scans for the three-point calibration in the open-loop program. The third is the sampling rate. The current program has a sampling rate of 200 Hz, which means data acquisition for each position displayed in the three-point occurs for 400 divided by 200, or 2 seconds. The last input is the error, which comes in from the AI Config.vi. This informs the AI Start.vi of all running errors and if it should proceed.

Once data acquisition has begun, data must be removed from the buffer at a rapid rate. This task is accomplished by the AI Read.vi, Figure 3.13(b). The AI Read.vi takes

in three parameters and outputs data to a waveform graph displayed on the front panel. The two parameters common among all the DAQ components are the device number and error, which gets transmitted throughout the code. The third parameter is the number of scans to read in from the buffer at one time. The three-point reads in all 400 scans at the same time and the data to a waveform graph. It is acceptable to read the scans in all at once because no analysis is being completed on each individual scan.



**Figure 3.13** Controls data acquisition. (a) Starts acquiring data from the eye movement tracker and storing into a buffer. (b) Removes data from the buffer and stores displays in a waveform graph.

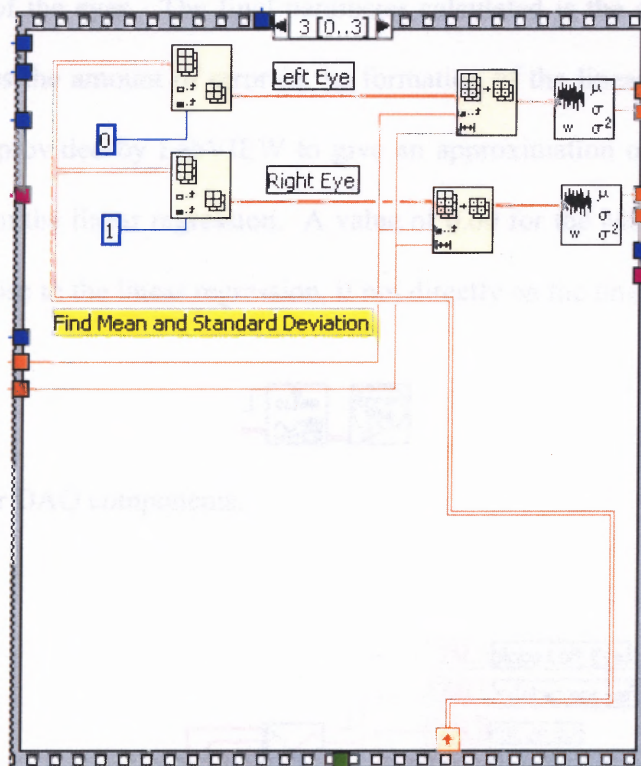
### 3.3.8 Analysis for Mean and Standard Deviation

Once data acquisition has completed digitizing the 400 scans or sampling points, sequence three proceeds. The data must be analyzed for the mean and standard deviation from sample points 100 to 399. Scans 1 – 100 are not analyzed because during this time, the eye moves toward the target thus the signal is in a transient state. In a vergence



response, there is a latency period of approximately 200 msec before the eyes begin to move to the target. Due to this, the first 0.5 seconds is not analyzed because the eyes are moving and to fuse the target. The analysis for a mean and standard deviation is completed in the last sequence of the for-loop, sequence three, Figure 3.14.

The data collected by the DAQ components from channel zero, the left eye movement signal, and channel one, right eye movement signal, is acquired as a single two-dimensional array represented by the double lines which must be split up to analyze each eye independently. After the channels are split up by the index array component, the two dimensional array has now become a single dimensional array for each eye which is then passed into the array subset component. The array subset component takes the input array and gives back only the portion of the array asked for, which for the three-point is the points between 99 and 399. This array is then passed into the standard deviation and variance subvi, which calculates the mean and standard deviation of each eye movement response. The mean and standard deviation are then stored into a new array for each of the degree positions presented to the subject.



**Figure 3.14** Analysis for mean and standard deviation for the right and left eye independently.

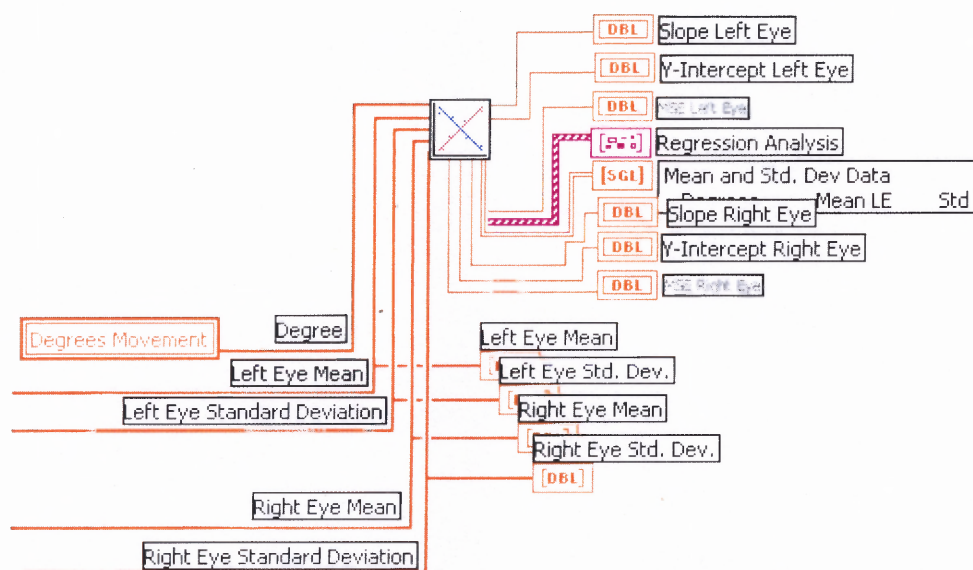
### 3.3.9 After Completion of For-Loop

The for-loop completes the exact same process for all three degree positions. Once the information has been gathered on the mean and standard deviation, the DAQ components are cleared, Figure 3.15, and a linear regression analysis is completed, Figure 3.16. The degree movement per eye and the means and standard deviations per eye are sent to the regression analysis subvi. Once the analysis is complete, a XY-graph is plotted displaying the mean voltage vs. the corresponding degree position and a regression line, for each eye. Along with the graph, the slope and y-intercept of the linear regression line are saved for each eye for use during the open-loop experimental trial section of the program. These two new parameters assist during the real time analysis to determine the

current position of the eyes. The final parameter calculated is the mean squared error, MSE, which gives the amount of error in the formation of the linear regression. It is a built in function provided by LabVIEW to give an approximation of how far away the points appear from the linear regression. A value of 0.00 for the MSE indicates that the points are very close to the linear regression, if not directly on the line.



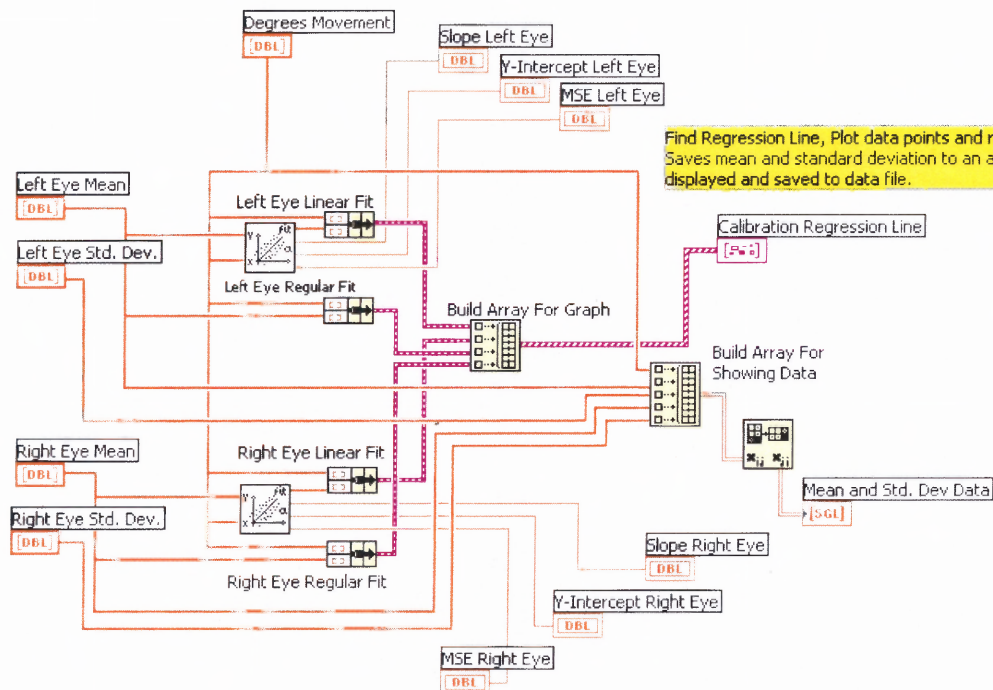
**Figure 3.15** Clear DAQ components.



**Figure 3.16** Input variables into regression analysis subvi for output of several eye movement variables and an analysis.

### 3.4 Regression Analysis

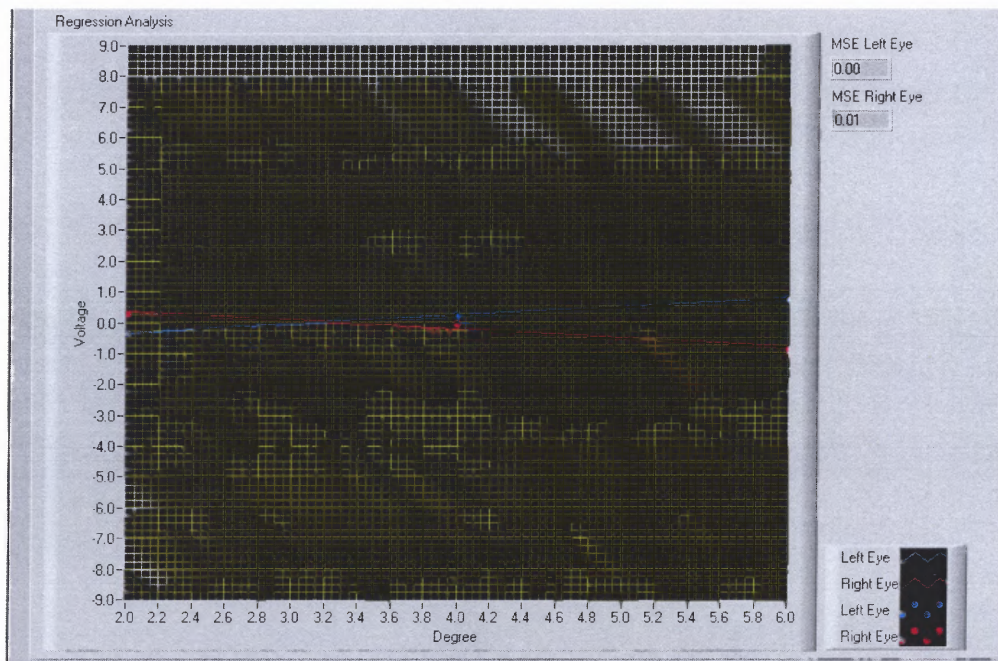
The regression analysis subvi, Figure 3.17, takes the array of means from the left eye response signals and the right eye response signal along with the corresponding known degrees position and enters them into the Linear Fit subvi provided by LabVIEW. This linear fit subvi analyzes the input points to generate the best fit linear line which goes through the points. Each eye is analyzed independently. The output of the linear fit subvi is the best linear fit data, slope, y-intercept, and MSE.



**Figure 3.17** Input variables into regression analysis subvi for output of several eye movement variables and an analysis.

The linear fit data are then matched with the corresponding degree position values and clustered for each eye. In addition, each eye's original voltage values and the degree positions are clustered. Clustering is a method of grouping values together to form an association between two values for graphing on an XY-plane. In addition, clustering

enables changes to occur when the values are wired to the cluster function. There are four sets of clusters at the end of this stage. Each of them is then inserted into the build array and then passed to the XY graph. An example of the output XY-graph can be seen in Figure 3.18. The mean and standard deviation are displayed to the operator through building an array with the information. The transform of this array is then taken to display the information in a logical sense as displayed in Figure 3.19.



**Figure 3.18** XY-Graph displaying regression analyses of three-point calibration and the original three-point calibration values.

Mean and Std. Dev Data		Mean LE	Std Dev LE	Mean RE	Std. Dev RE
Degrees					
0	1.00	-0.35	0.10	0.29	0.04
0	3.00	0.24	0.06	-0.06	0.09
	5.00	0.75	0.04	-0.90	0.18

**Figure 3.19** Input variables into regression analysis subvi for output of several eye movement variables and an analysis.

## **CHAPTER 4**

### **OPEN-LOOP METHODOLOGY**

The open-loop algorithm is a robust algorithm, which underwent three revisions each following alternate methodologies to find the optimal working error free code. All methodologies conduct a real time analysis of the position of the eyes to determine the next position where the target would appear on the visual displays which for this experimentation set-up are a pair of computer monitors. The first methodology was simple and kept a constant error between the eyes and the target. It attempted to reduce the movement of the target as the eyes progressed toward the target, but was not very successful. The second methodology was a more robust method, which required storing the previous eye position in a variable to compare with the current position of the eyes to determine the next position of the target. The last methodology is more mathematically oriented which does not rely on the previous position of the eye, but only the current position to determine the location of the next target. Before discussing each methodology, the structure of the open-loop code section will first be presented to understand the flow. Each methodology is explained in greater detail in section 4.3.

## 4.1 Open-Loop Sequences

The open-loop code flows through seven sequences. Each of the sequences is assigned a task similarly to the three-point calibration.

### 4.1.1 Stimulus Selection

Sequence zero is responsible for stimulus selection, shown in Figure 4.1. In the initial setup stage, the operator is able to enter a total of ten types of stimuli to be presented to the subject. This portion of the code selects one of the stimuli for open-loop presentation.

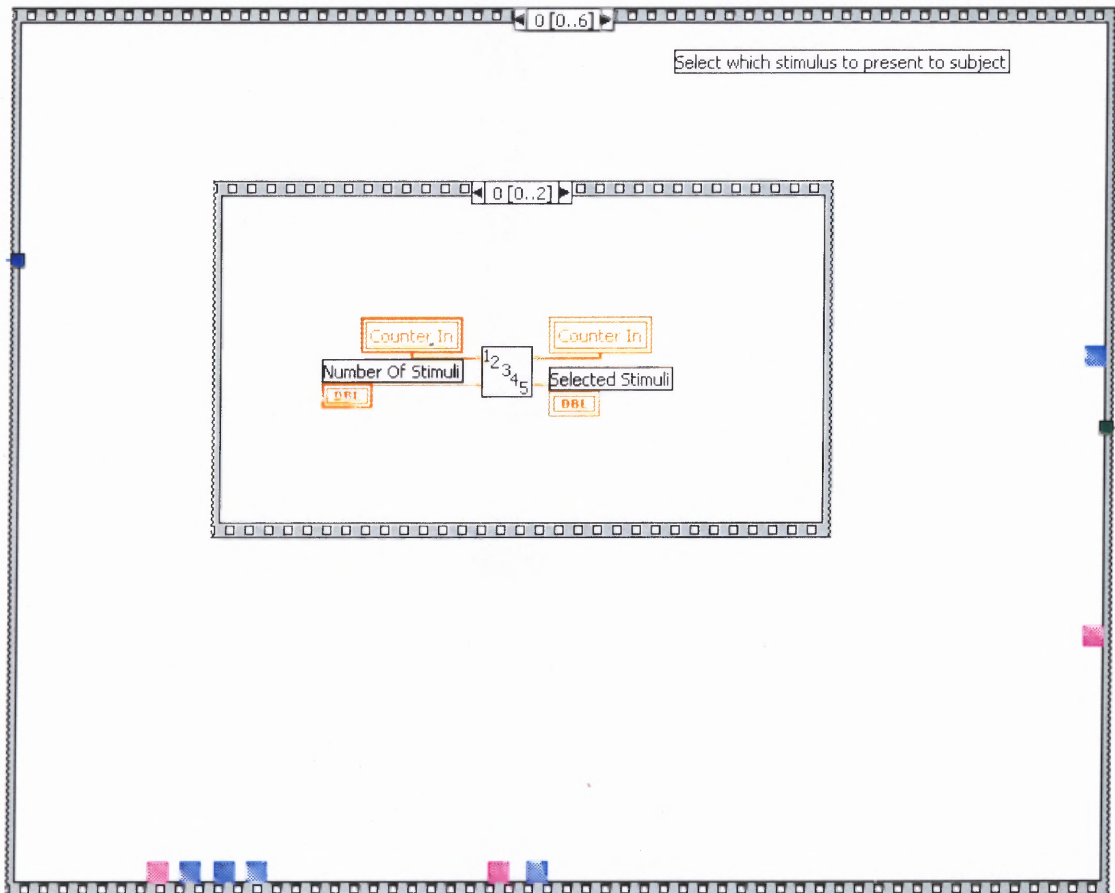


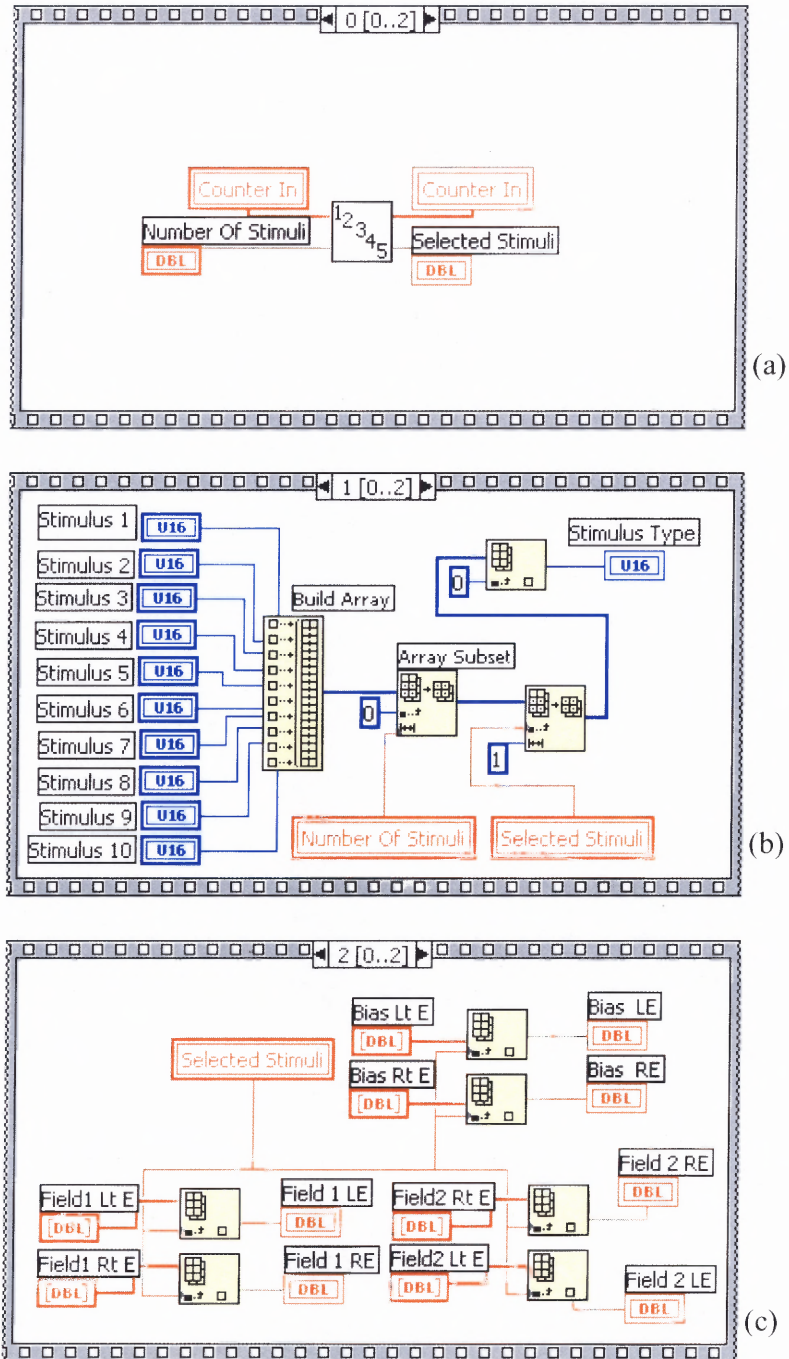
Figure 4.1 Sequence one: Stimulus Selection.

The first part of stimulus selection occurs through the selection of the particular stimuli. The subvi *Counter Organization* is invoked to keep track of how many times the particular stimulus has been called upon and to select the stimulus. This subvi is further explained in subsection 5.4.2. The total number of stimuli is passed into the counter organization subvi. This subvi also logs the number of times each stimulus has been called upon. The output to the subvi is the selected stimulus number and the counter array with the counter of the particular stimulus selected incremented by one, as shown in Figure 4.2(a).

The next part of stimulus selection is to select the type of stimulus to present to the subject, Figure 4.2(b). This is accomplished by first building an array of all the types of stimuli presented by the operator. The array is then reduced to the number of stimuli the operator requests by calling the array subset function. This array is then further decreased to the particular stimulus selected by the counter organization subvi. Lastly, the index array function is called to return the number corresponding to the specific stimulus type.

The final portion of the stimulus selection sequence is to select the parameters of the particular stimulus type, Figure 4.2(c). This is completed by taking each of the six parameter arrays and applying the index array function to select the parameters of the particular stimulus type.

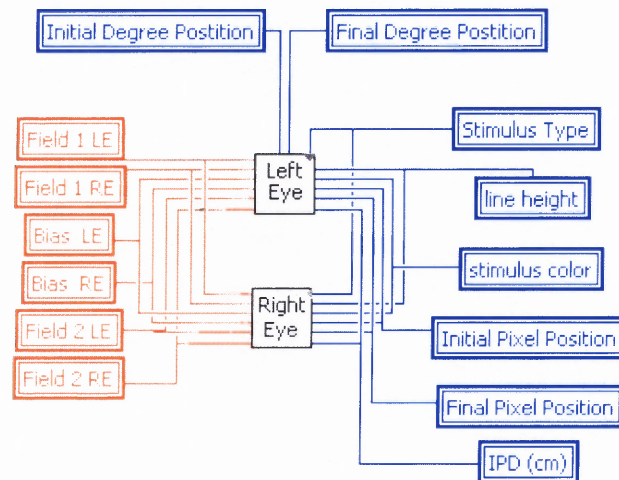




**Figure 4.2** Stimulus selection (a) Counter control and stimulus selection. (b) Selection of the stimulus type. (c) Selection of the stimulus parameters for the particular stimulus number.

### 4.1.2 Initial Stimulus Presentation

The next sequence is responsible for initializing the stimulus presented to the left eye and right in their respective subvi, Figure 4.3. This portion of the code was developed keeping in mind the stimulus presented to the subject in the future may be different to each eye, which is why the right eye and left eye have a different subvi. To each of these subvi are passed the parameters shown in Figure 4.3 and have no output variables. Currently, the right eye subvi is not utilized by the open-loop program. However, the left eye subvi is responsible for presenting the initial target on the monitors before the trigger button has been pressed by the subject. This gives the subject's eyes an initial position to fixate upon.

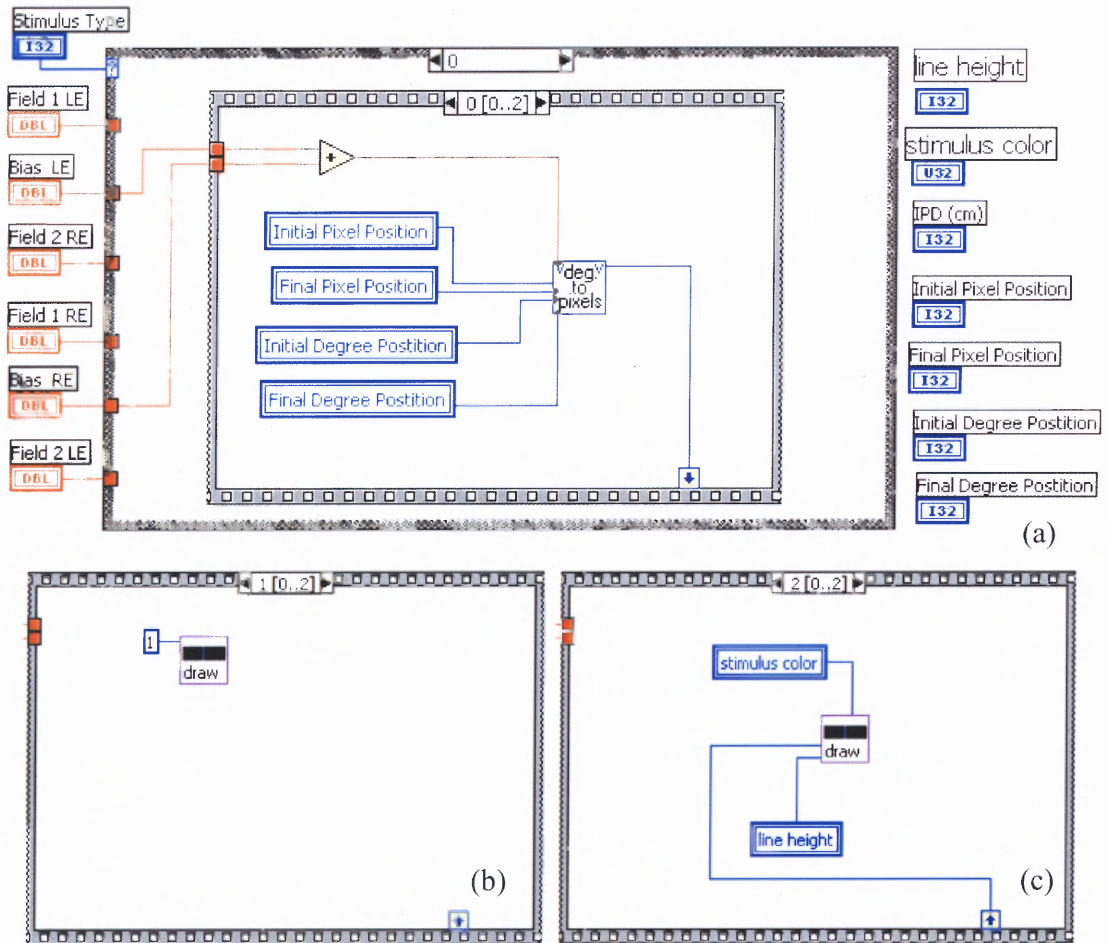


**Figure 4.3** Setup of stimulus to left eye and right eye.

The structure and design of the left eye subvi, Figure 4.4, and right eye subvi are identical, although the stimulus contained within the left eye subvi is not present in the right eye subvi. This would be redundant and could cause problems in the case where both subvis execute simultaneously. Within both the subvis, the flow through the code

begins as follows: the stimulus type being presented to the subject is first selected in order to execute the proper code to present what the operator has asked for. Next, the variables are passed in and the particular code for the stimulus type executes.

In the left eye of the open-loop procedure, an initial target had to be presented to the subject on the computer monitors before the trigger button was pressed. This portion of the code follows three sequences. The first sequence, Figure 4.4(a), determines the pixel position of where the target should appear on the monitor depending on the degree. The second sequence, Figure 4.4(b), clears the monitor of all graphics and the third, Figure 4.4(c), displays the new target at the correct position on the monitors.



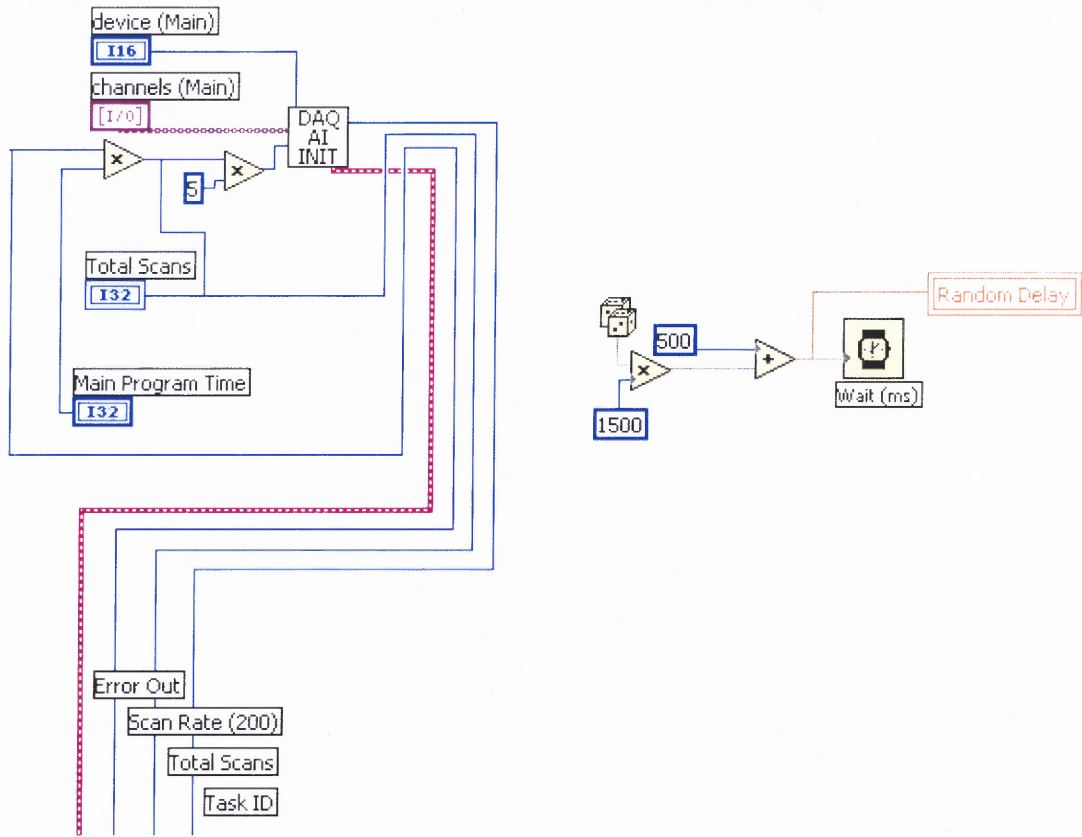
**Figure 4.4** Left Eye Subvi (a) Overall structure of of subvi and sequence zero determines pixel position for inputted degree. (b) Sequence one clears monitors. (c) Sequence two presents target to subject.

#### 4.1.3 Trigger Button Initiates DAQ and Random Delay

Sequence two controls the trigger button. The subject can pause at any time if he or she feels fatigue. Thus the subject is given control over when the open-loop experiment data acquisition and stimulus presentation begin. Pressing the trigger button initiates the execution of sequence three, Figure 4.5. This sequence initializes the DAQ card for data acquisition and invokes a random delay to suppress subject prediction of when the target will move.

The DAQ AI INIT subvi receives three parameters during initialization. Parameter one informs the DAQ components from which device to read the scans. Parameter two tells from what channels the scans will be taken and parameter three creates the buffer in which the scans will be stored. The buffer is created by multiplying the main program time variable, the scan rate of 200 Hz used for vergence eye movement acquisition for open-loop experiment, and the digit five. The multiplication by five gives the buffer the optimal buffer size as specified by LabVIEW technicians. (LabVIEW Technician)

One of the primary reasons why the DAQ component is initialized during each run is to allow the operator to have additional control of the buffer size and the experimental run time during the open-loop experiment. If the DAQ component were initialized only once before beginning the six sequence open-loop code, the run time of the program could not be changed during execution without creating buffer underrun errors. This method also enables the buffer to be cleared in order to prevent any information remaining in the buffer from the previous run from entering the current run.



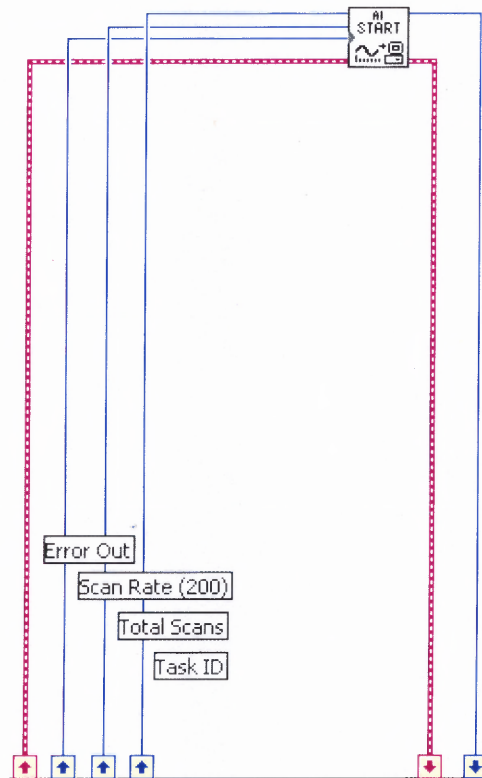
**Figure 4.5** DAQ initialization and random delay.

During this stage of program development, a problem was encountered where the DAQ component for data acquisition would not work. The original setup of the code initialized the DAQ component first and then called the trigger press algorithm. The conflict arose in the AI Config.vi. The DAQ component's AI Config was being initialized for data acquisition, then the trigger button's AI Config was being initialized which overrode the DAQ AI Config. This resulted in buffer errors, since no buffer was present to insert scans information. This problem was resolved by reversing the order of presentation of the AI Config.

This sequence is also responsible for the random delay. A delay of 0.5 to 2 seconds is given before the start of data acquisition, Figure 4.5.

#### 4.1.4 Start Filling Buffer – AI Start

Sequence four is responsible for starting data acquisitions and storing the scans in the buffer. It receives four parameters from sequence three: the error and device numbers are passed from the DAQ AI INIT subvi, the scan rate of 200 Hz for vergence movements, and the total number of scans to gather. The total number of scans is the main program time multiplied by the scan rate. These parameters are passed into the AI Start.vi.



**Figure 4.6** Start data acquisition.

#### 4.1.5 Stimulus Step and Open-Loop

Sequence five presents the stimulus to the subject and begins the real time analysis. This sequence occurs in two stages. Stage one presents the initial stimulus to the subject, Figure 4.7. It works in four subsequences in which sequence zero, Figure 4.7(a), builds the pixel value representing the target degree position to present to the subject. Sequence

one, Figure 4.7(b), builds the pixel value of the current degree position the subject is looking at to be removed from the monitor. Sequence two, Figure 4.7(c), clears the monitors of all graphics. Lastly, sequence three, Figure 4.7(d), presents the new target to the subject.



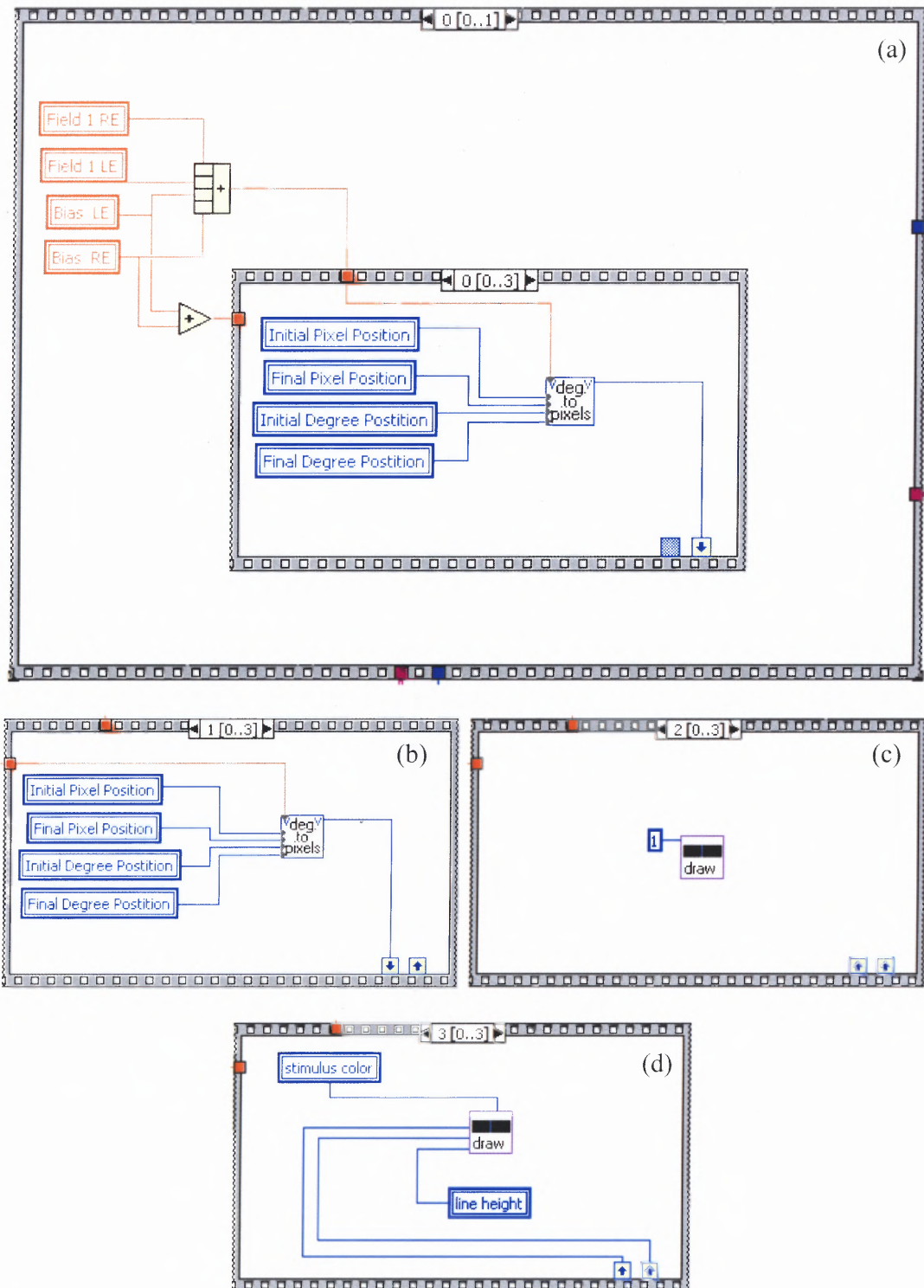
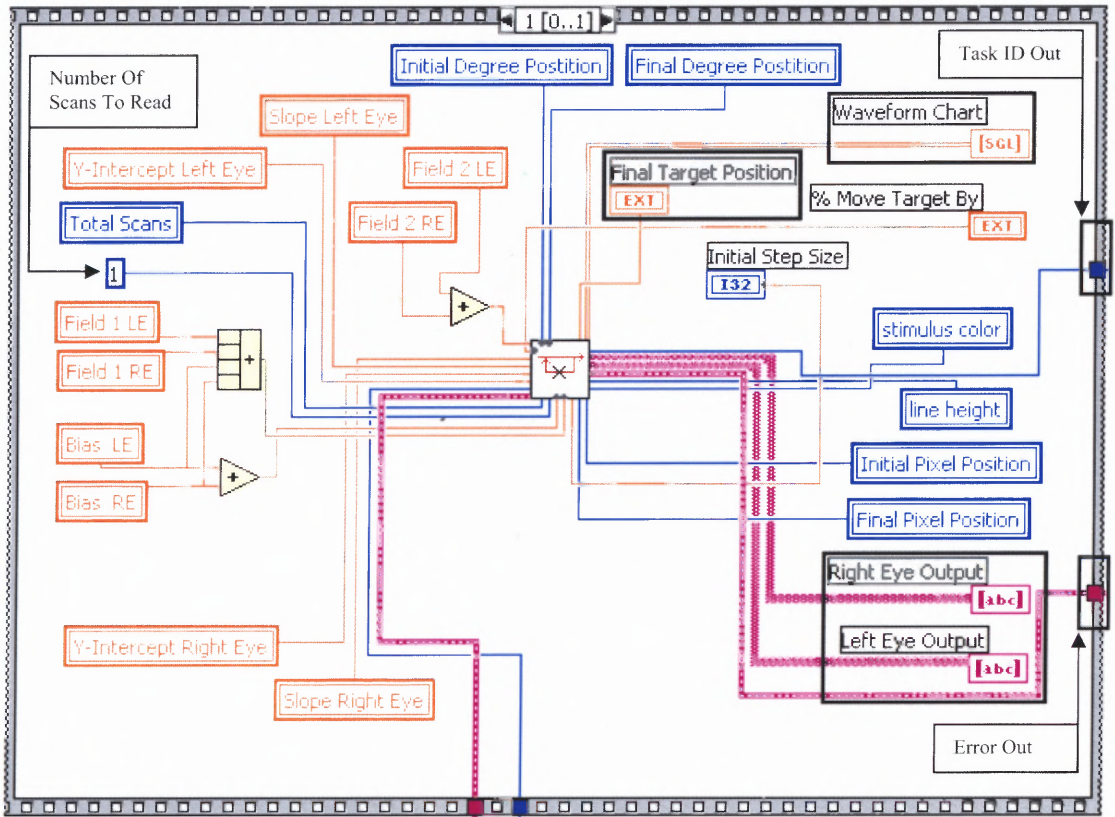


Figure 4.7 Stage one to present new target to subject.

The second stage is the calling of the open-loop subvi to begin a real time analysis of each scan. There are several inputs to the subvi shown in Figure 4.8, but only six outputs. The six outputs are in a black box: the waveform chart, the final target position presented to the subject on the monitor, the task ID and error code from the AI read, and the data collected from the right and left eye. All the input variables are labeled indicating what they stand for. The open-loop subvi is explained in greater detail in section 4.2.



**Figure 4.8** Calls the open-loop subvi to begin a real time analysis of each scan to present next target position to subject.

#### **4.1.6 Clear DAQ Components**

The final sequence in the open-loop code clears the DAQ card. This procedure clears the buffer before the next initialization occurs and maintains the stability and operation of both LabVIEW and the computer.

### **4.2 Open-Loop Subvi**

The open-loop subvi is responsible for collecting the eye movement data in voltage values and sending the information through the three-point regression analysis to calculate with the current position of the eyes in degrees. This must be completed in real time in order to maintain tracking of the eye movement and present the next target to the subject on the monitors. The subvi shown in Figure 4.9 processes through three sequences within the for-loop.

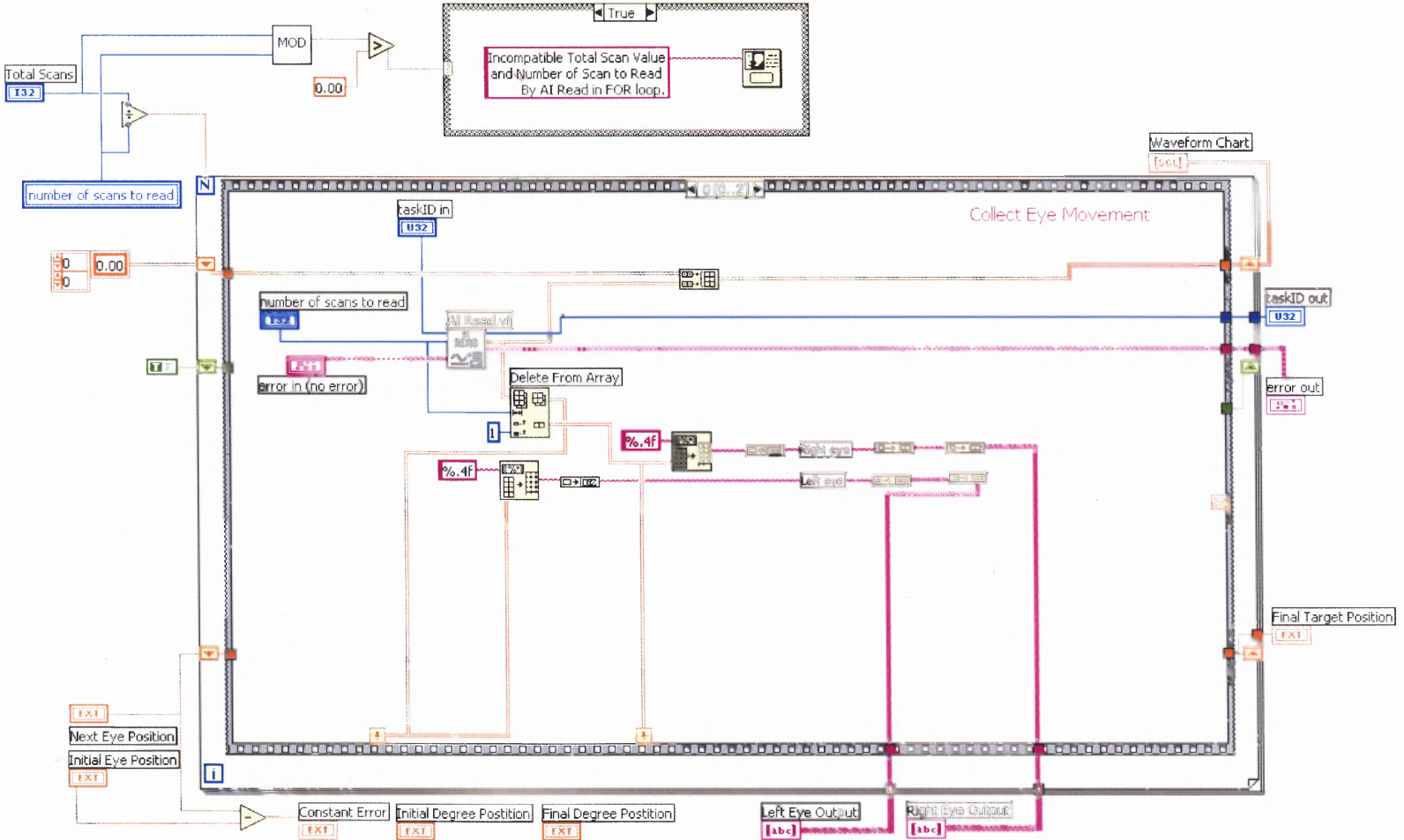


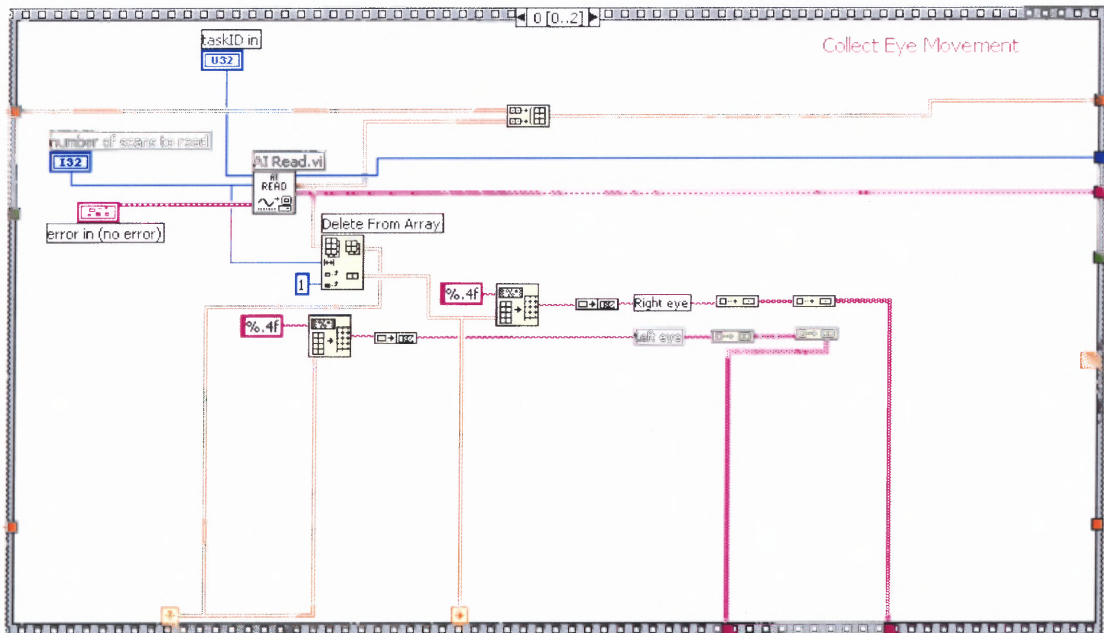
Figure 4.9 Overall structure of the open-loop subVI.

#### **4.2.1 Iterations of For-Loop**

The total number of iterations for the for-loop is controlled by the total number of scans to acquire divided by the number of scans to read by the AI Read.vi at a time. For the open-loop, one scan is read at a time and analyzed in real time. The number of scans read in by the AI Read is under operator control and to insure the number which is taken allows the for-loop to acquire all the scans, a validation is completed. For instance, if the total scans is 1000 and the operator requested reading in three scans at a time by the AI READ, then 1000 divided by three is not a whole number and the total number of scans requested will not be acquired. To solve this problem, the MOD subvi was created to locate a remainder during the division process. This is a very simple subvi which the current LabVIEW edition does not provide and the coding behind it is discussed in subsection 5.5.4. If a remainder was found, then the operator was notified that an incompatible number of scans to be read by the AI Read was entered.

#### **4.2.2 Acquire Eye Movement Position From Eye Movement Tracker**

Within the for-loop, sequence zero is given the task of obtaining the scans from each channel and splitting the scans into their independent arrays designated as left eye and right eye. Along with this, the scans are converted to a string value and then placed into an array in the wall of the for-loop to save the data acquired by the DAQ card, Figure 4.10. Once the for-loop has completed all tasks, the arrays with the saved left eye and right eye data are passed to the original calling program, *Open-Loop Program 1.0*, to save into a file for future analysis.



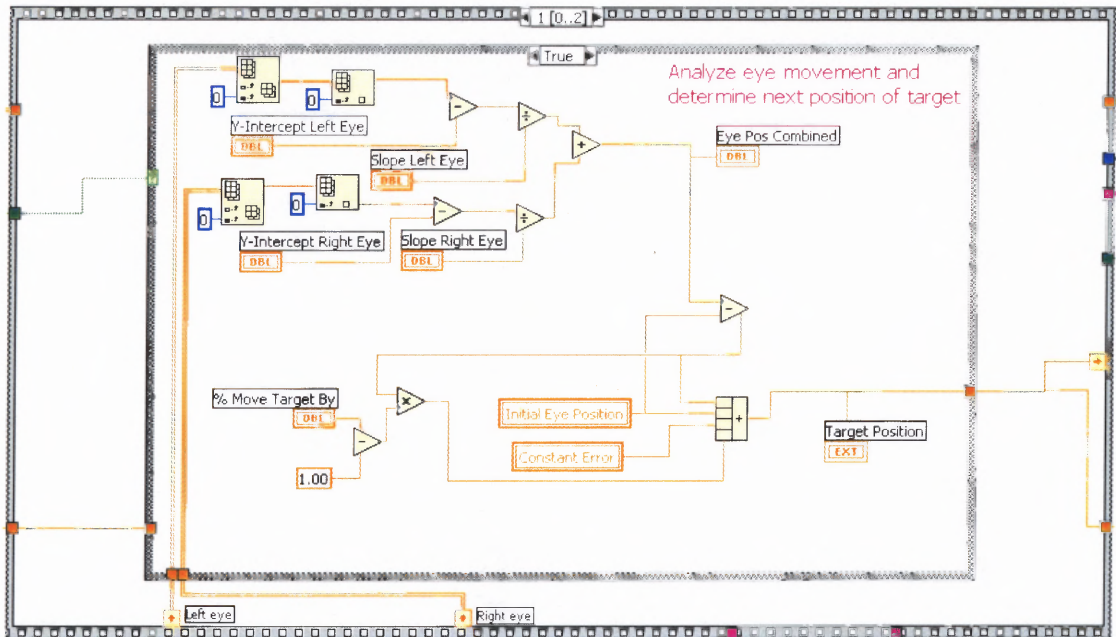
**Figure 4.10** Acquire scans from DAQ card and split scans for each eye into individual array for independent analysis.

#### 4.2.3 Determine Eye Position In Degree and New Target Location

Sequence one is responsible for analyzing the eye movement and determining the next position of the target, Figure 4.11. The eye movements are analyzed in real time by employing the regression analysis from the three-point calibration to convert the voltage value to the current degree position. This is accomplished by using the slope and y-intercept for each eye. Since it is a linear regression analysis, the equation  $y = mx + b$  can be applied to solve for the unknown degree. In this equation,  $y$  is equal to the voltage value and  $x$  is equal to the corresponding degree value. The equation now turns into  $\text{degrees} = (\text{voltage value} - b) / m$  which is what appears in Figure 4.11.

Once the conversion is complete, the position of the eyes are summed. This is the methodology used in the vision research community to determine the current position of the eyes during vergence movements. To keep consistent with current methodology, this

procedure has been applied throughout the code when presenting the next target to the subject. The three methodologies used to determine target position are explained in greater detail in section 4.3.

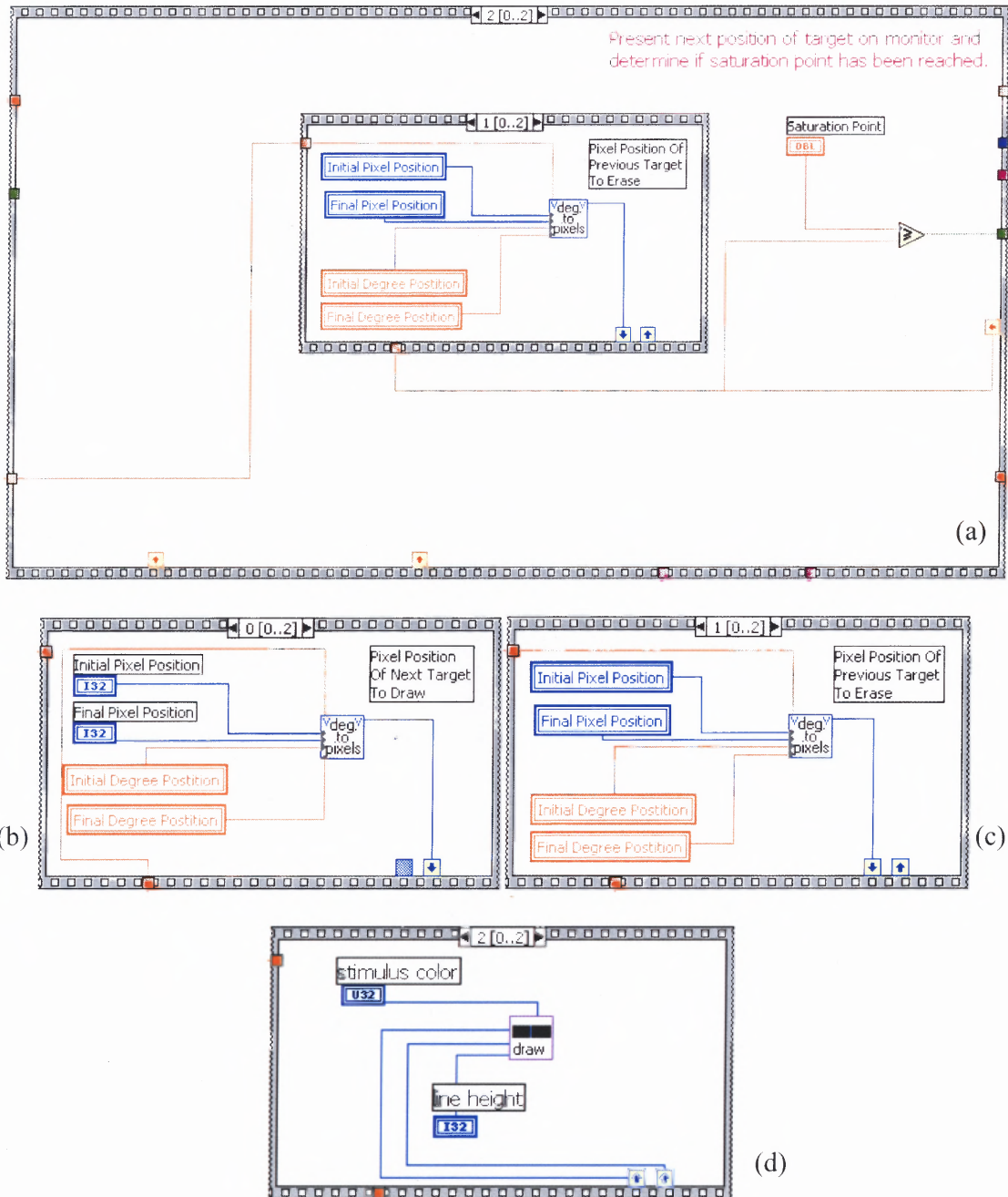


**Figure 4.11** Determine position of eyes in degrees and position of next target to display.

#### 4.2.4 Target Presentation on Computer Monitor

The final sequence, sequence two displays the new target to the subject and removes the previous target, Figure 4.12. The method of accomplishing this task is the identical to the one used in displaying the initial step to the subject described earlier. A calculation is completed to determine if the saturation point has been reached. This is completed by comparing the current position of the target and the position at which the operator wants the target to stop. Once the saturation position has been reached by the target, a flag is set to no longer analyze the eye movements in real time during sequence one and to

one and to display the same target at the position it currently resides while simultaneously collecting the scans from the DAQ card.



**Figure 4.12** (a) Overall structure to display next target and remove previous target along with checking for saturation of target. (b) Determine pixel position of next target to display. (c) Determine pixel position of previous target to remove. (d) Display and remove the respective targets simultaneously.



### 4.3 Open-Loop Methodology

The real time data analysis and presentation of the next target on the monitor went through three methodologies, each working differently. The method of acquiring the eye movements and analyzing them in real time to determine the current position in degrees remained constant throughout all three methods. The portion, which differed was the method in which the next target was displayed to the subject. Three items had to be accounted for in each methodology. The first was the initial step size. The second was by what the target should move which was dependent on the displacement of the eyes from the previous position, the current position, the variable *% Target Moves By*. The third was at what position on the monitors should the target saturate. Both these items were under the control of one variable, *% Move Target By*. The value in this variable ranged between zero and one.

When the value was zero, this indicated open-loop should be shut off, since the target will not move based on current position of the eye, table 4.1 and Figure 4.13. When the value was any number between zero and less than one, the eyes should be able to catch up to the target at some point in time, table 4.2 and Figure 4.14. This meant the rate at which the target is moving is slowing down at a constant velocity as the eyes approach. The last situation is when the value is equal to one, table 4.3 and Figure 4.15. During this condition, a full open-loop procedure takes place, in which the moving target keeps a constant error with the position of the eyes.

The difference between table 4.1 and table 4.2 is the amount the target is able to move depending on the how much the eyes have moved from their previous position. In table 4.1, the target is not allowed to move at all which is why it remains stationary at six

degrees. In table 4.2, the target is allowed to move 60 percent (0.60) of the eye movement. This is why when the eyes move one degree, the target is able to move .6 degrees further beyond its previous position. The method in how this works is to take the difference between the current position of the eyes and the previous position and then multiply this value by the percent the target is able to move. In the case of the table, the eyes move one degree each time, so the target moves .6 degrees each time ( $1 \times 0.60 = 0.60$ ). In this situation, the eyes are able to catch up to the target at 12 degrees, at which time in an ideal situation the target should cease to move since the eyes will be focused in on the target.

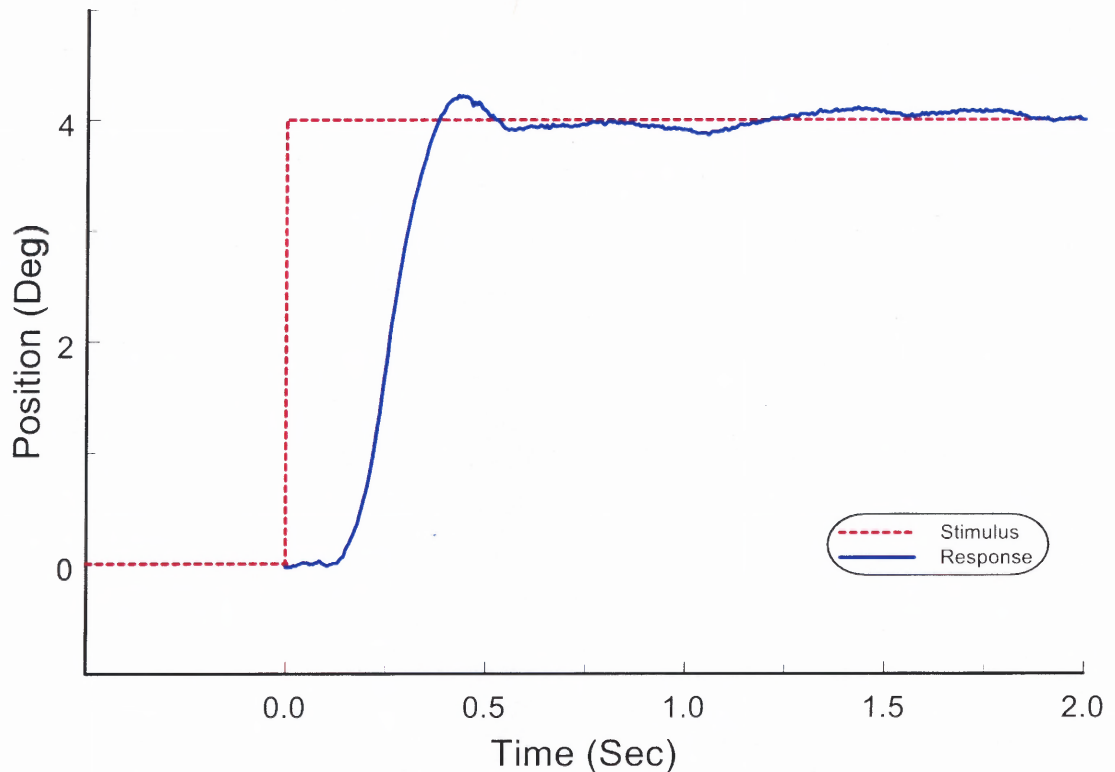
Tables 4.3 also exhibits a difference between tables 4.1 and 4.2. Table 4.3 keeps a constant error between the target and the position of the eyes. Thus, when the eyes move one degree, the target moves one degree further. This is calculated in the same fashion as the values in table 4.2, but the percent which the eye movement displacement is multiplied by is now 100 percent (1.00). Thus, if the eyes move one degree then  $1 \times 1.00 = 1.00$ . This value is added to the previous position of the target to attain the new target position. When maintaining a constant error, the eyes will not be able to catch up to the target unless the target has reached its saturation point designated by the operator.

The three methodologies determined how the next stimulus is presented, depending on the value in the variable. The tables were created using methodology two and information on how open-loop should be presented to the subject.

**Table 4.1** Position of Eyes When % *Target Move By Variable* Is Set To Zero Percent and Initial Step Size Is Four

Position Of Eyes (deg)	2	3	4	5	6	7	8	9	10	11	12
Target Position (deg)	6	6	6	6	6	6	6	6	6	6	6

### 0% Open Loop Stimulus and Response

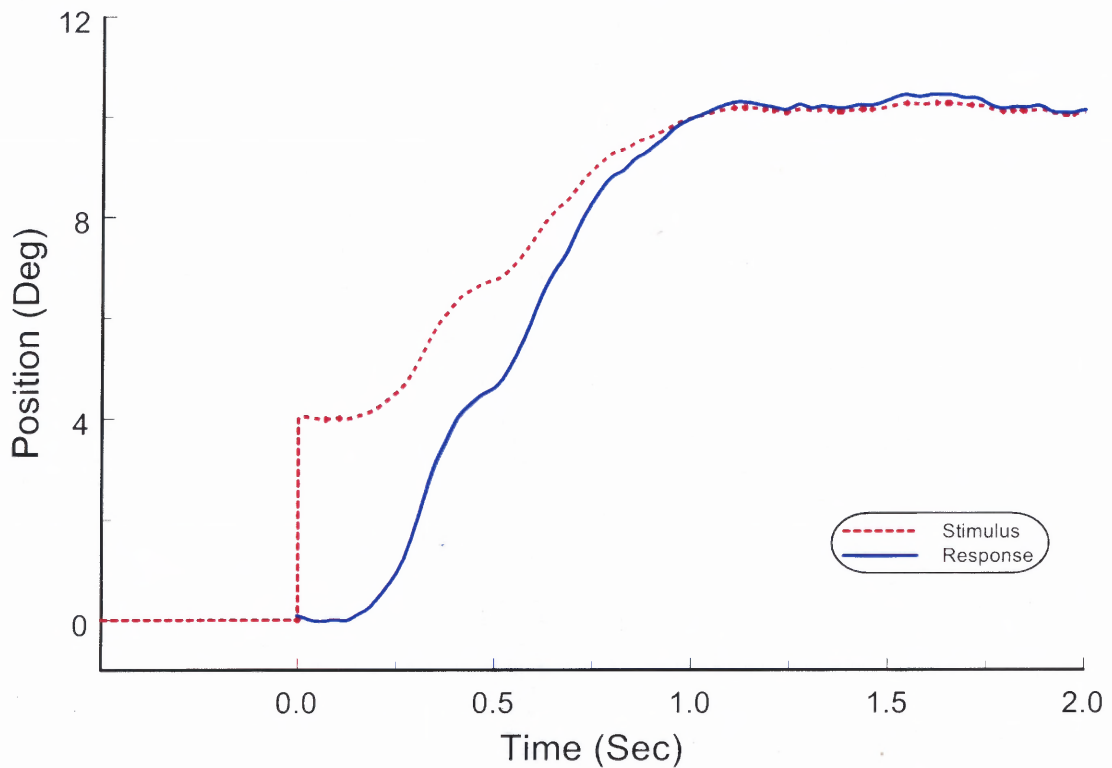


**Figure 4.13** Time plot of a step movement. Convergence movement of a four degree step when zero percent feedback given to target during open-loop stimulation. Red dotted line shows the position of the stimulus with respect to time. Blue line shows the eye movement response.

**Table 4.2** Position of Eyes When % *Target Move By* Variable Is Set To Sixty Percent and Initial Step Size Is Four

Position Of Eyes (deg)	2	3	4	5	6	7	8	9	10	11	12
Target Position (deg)	6	6.6	7.2	7.8	8.4	9	9.6	10.2	10.8	11.4	12

### 60% Open Loop Stimulus and Response

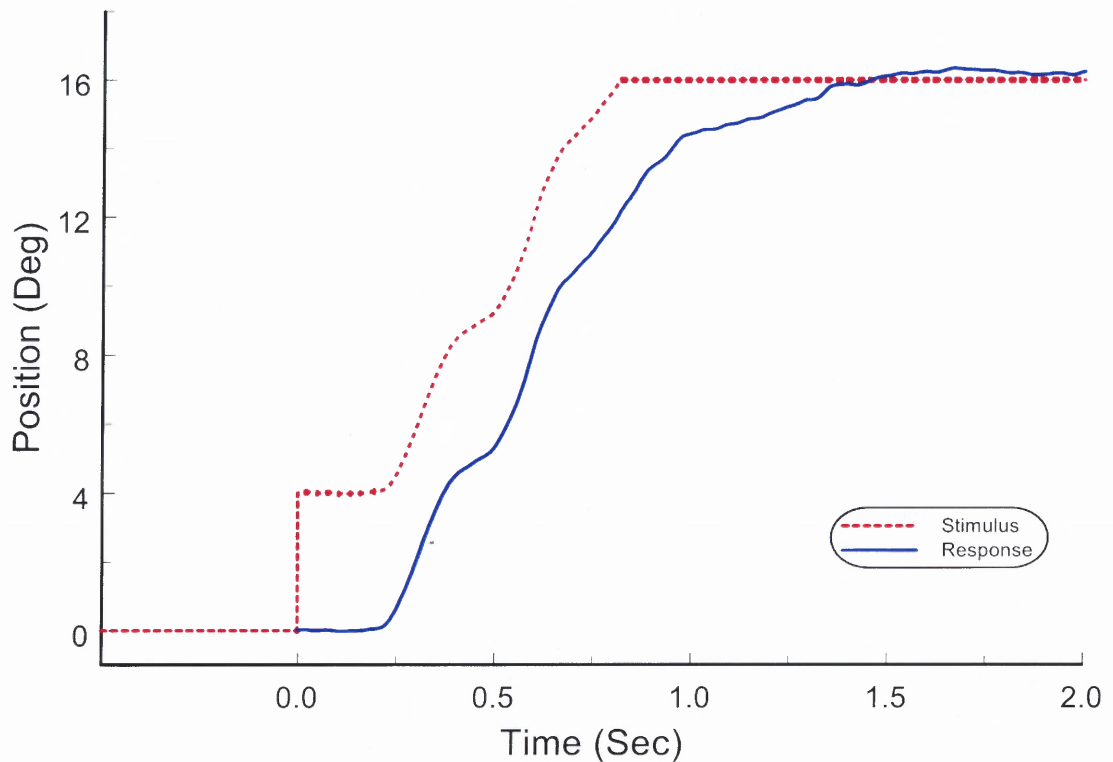


**Figure 4.14** Time plot of an open-loop response. An initial four degree convergence eye movement is followed by subsequent eye movements to fixate on target. Target receives sixty percent feedback during open-loop stimulation. Red dotted line shows the position of the stimulus with respect to time. Blue line displays the eye position with respect to time. The oculomotor system is able to fixate on target at 10 degrees.

**Table 4.3** Position of Eyes When % *Target Move By Variable* Is Set To One Hundred Percent and Initial Step Size Is Four

Position Of Eyes (deg)	2	3	4	5	6	7	8	9	10	11	12
Target Position (deg)	6	7	8	9	10	11	12	13	14	15	16

### 100% Open Loop Stimulus and Response



**Figure 4.15** Time plot of an open-loop response. An initial four degree convergence eye movement is followed by subsequent eye movements to fixate on target. Target receives one hundred percent feedback during open-loop stimulation. Red dotted line shows the position of the stimulus with respect to time. Blue line displays the eye position with respect to time. The oculomotor system is able to fixate on target at 16 degrees, at which point the target is set to saturate and cease eyes movement. A constant four degree error is maintained between target and position of eyes.

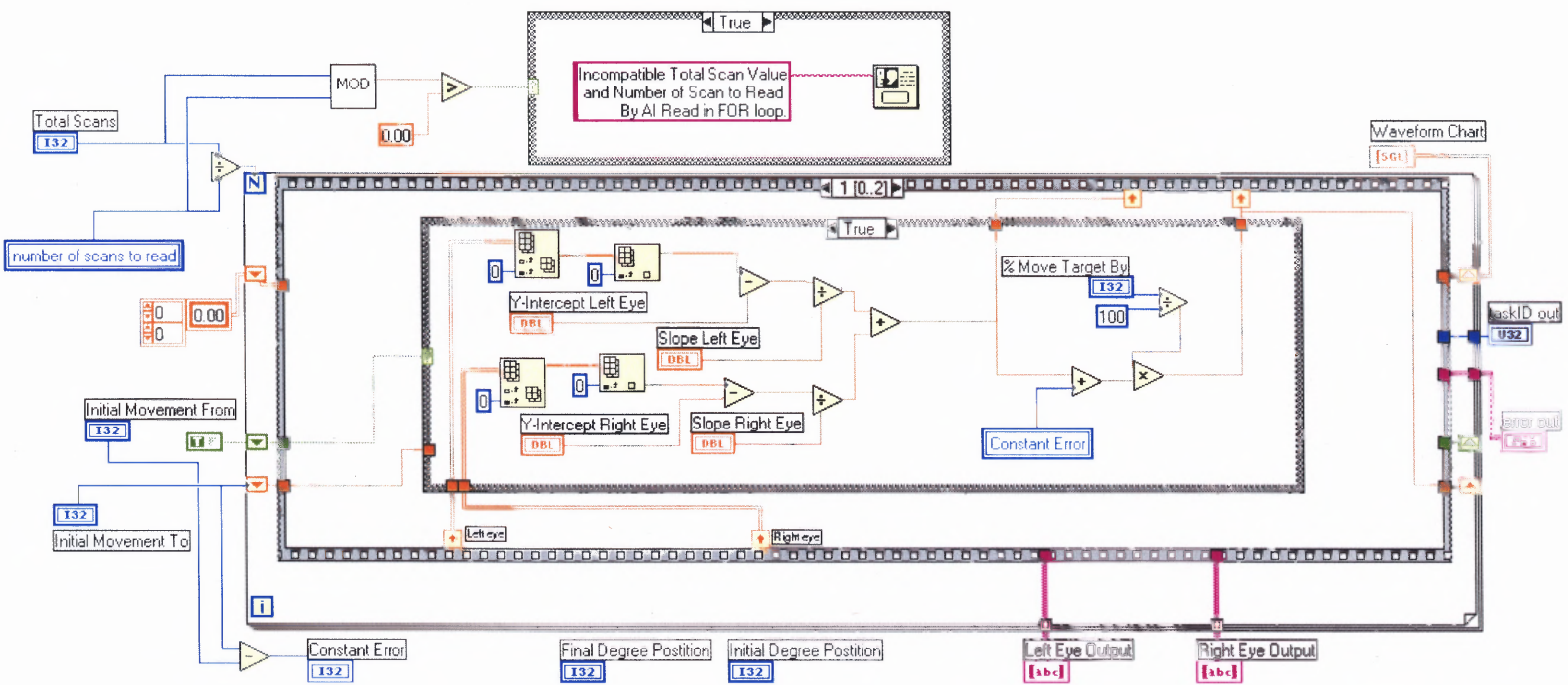
### 4.3.1 Methodology One

This first method, Figure 4.16, was created to keep a constant error between the position of the eyes and the stimulus position. The code began by taking the difference between the position of the target after the trigger button had been pressed and the position of the target before the trigger button had been pressed to obtain the constant error. This procedure was completed outside the for-loop.

The next step was to display the target to the subject in the correct location as the eyes moved. This was accomplished by acquiring the current position of the eyes and adding on the constant error. To take into account by how much the target should move by using the *% Move Target By* variable to control the feedback allowed to the subject, the current position of the eyes plus the error were multiplied by the value in the variable.

Although this method allowed the target to move as the eyes moved, it was not performing an open-loop experiment when the variable was any value other than one. This was because the constant error remained constant, yet smaller due to the value in the variable. When the value in the variable was between zero and one, the target did not slow down, but kept a constant error smaller in magnitude. This method was improved upon with the creation of methodology two.

Figure 4.16 Open-Loop methodology one.



### 4.3.2 Methodology Two

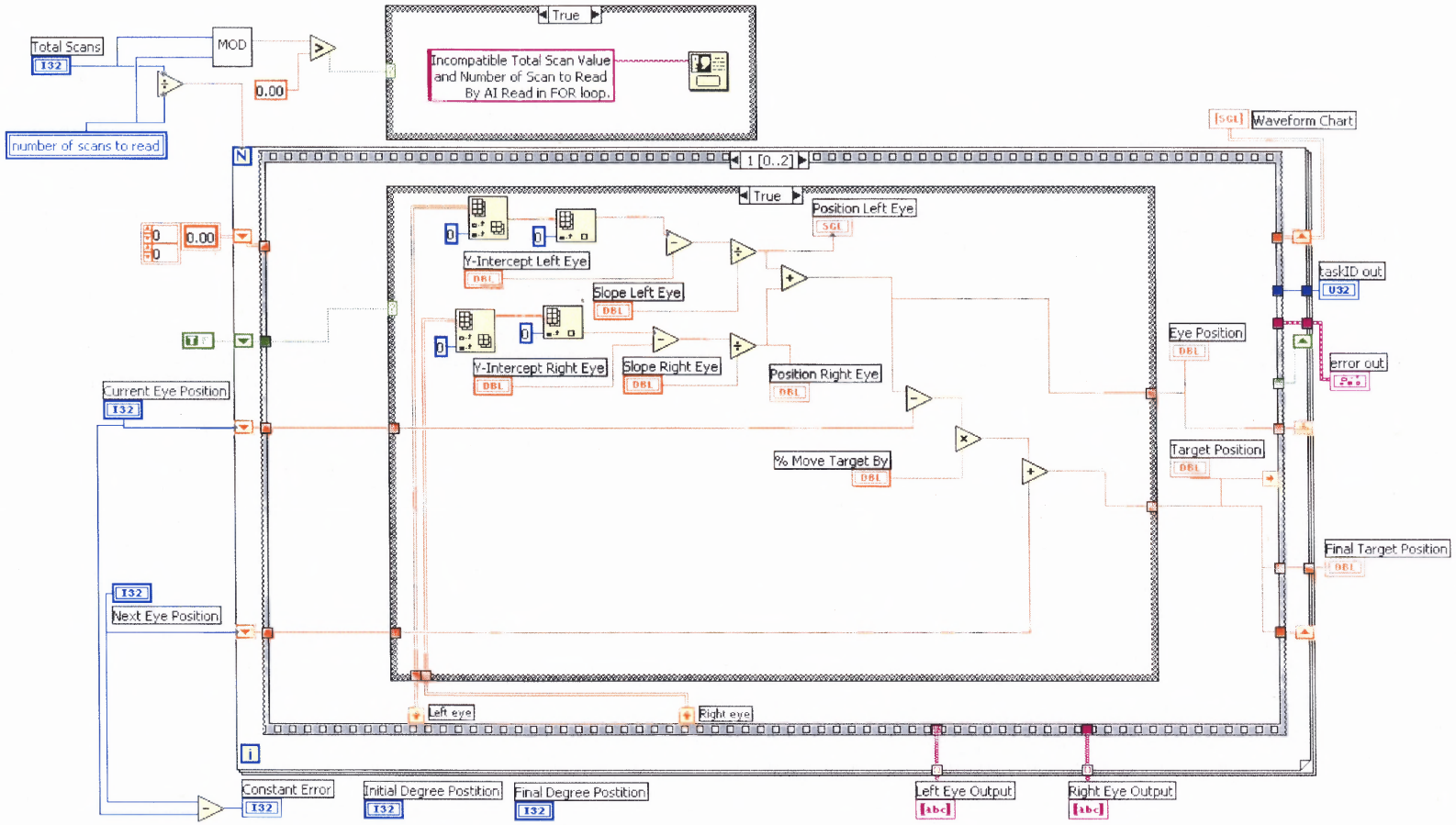
Methodology two, Figure 4.17, was more precise in how to determine the position of the next target. It no longer used the constant error of the step size and the displacement between the previous and current position of the eyes to determine the next target position. It is a more robust algorithm, which accounted for the previous eye position and the current eye position to determine the next position of the target.

This is accomplished by storing in memory the previous eye movement and location of the target before deciding the new target position. First, the difference is taken between the current position of the eyes and the previous position to determine the displacement. This value is then multiplied by *% Move Target By* variable to determine how much further the target should be moved. This new value is then added to the previous target position to determine the new position for the stimulus display and passed to the next sequence.

When partial feedback by the oculomotor system is allowed, this method enables the subject to “catch up” to the target, the error is not held constant but decreases over the time of the experiment. In the situation when oculomotor feedback is to be fully countered by the open-loop thus creating a full open loop system, the target continually moves not allowing the subject to “catch up” or the error is kept constant between the current eye position and the stimulus target. The stimulus target will reach a saturation point, at which time it will no longer move. Analyzing the position of the target depending on the location of the eyes assisted in the derivation of an equation explained in methodology three.



Figure 4.17 Open-Loop methodology two.



### 4.3.3 Methodology Three

The final methodology, Figure 4.18, is a more mathematical one which no longer needs to store information about the previous eye position or target. It is able to determine the next position of the target depending on a single variable, the current position of the eyes, and three constants, the amount of feedback the system allows controlled by the *% Move Target By* variable, the initial step size, and the initial starting position of the eyes before trigger press. The location of the eyes primarily determines the location of the target.

The motivation behind this equation came about from studying table 4.2. A pattern was seen in the current position of the target. This pattern depended upon the current position of the eyes, the initial position of the eyes, and the initial position of the target. When the eyes had moved a total of four degrees from the origin, and the target was allowed to move 60 percent (0.60) of the eye movement, then the target had to move 2.4 degrees beyond the original position. This pattern was exploited to bring about the following equation:

$$NTP = W + X + Y + Z \quad (\text{Eq 1})$$

Where

NTP = New Target Position

W = Initial Eye Position Before Trigger Button Press

X = (Current Eye Position - W)

Y = X\*(Feedback-1)

Where

Feedback = *% Move Target By* (in decimal form)

Z = Constant Error = Step Size

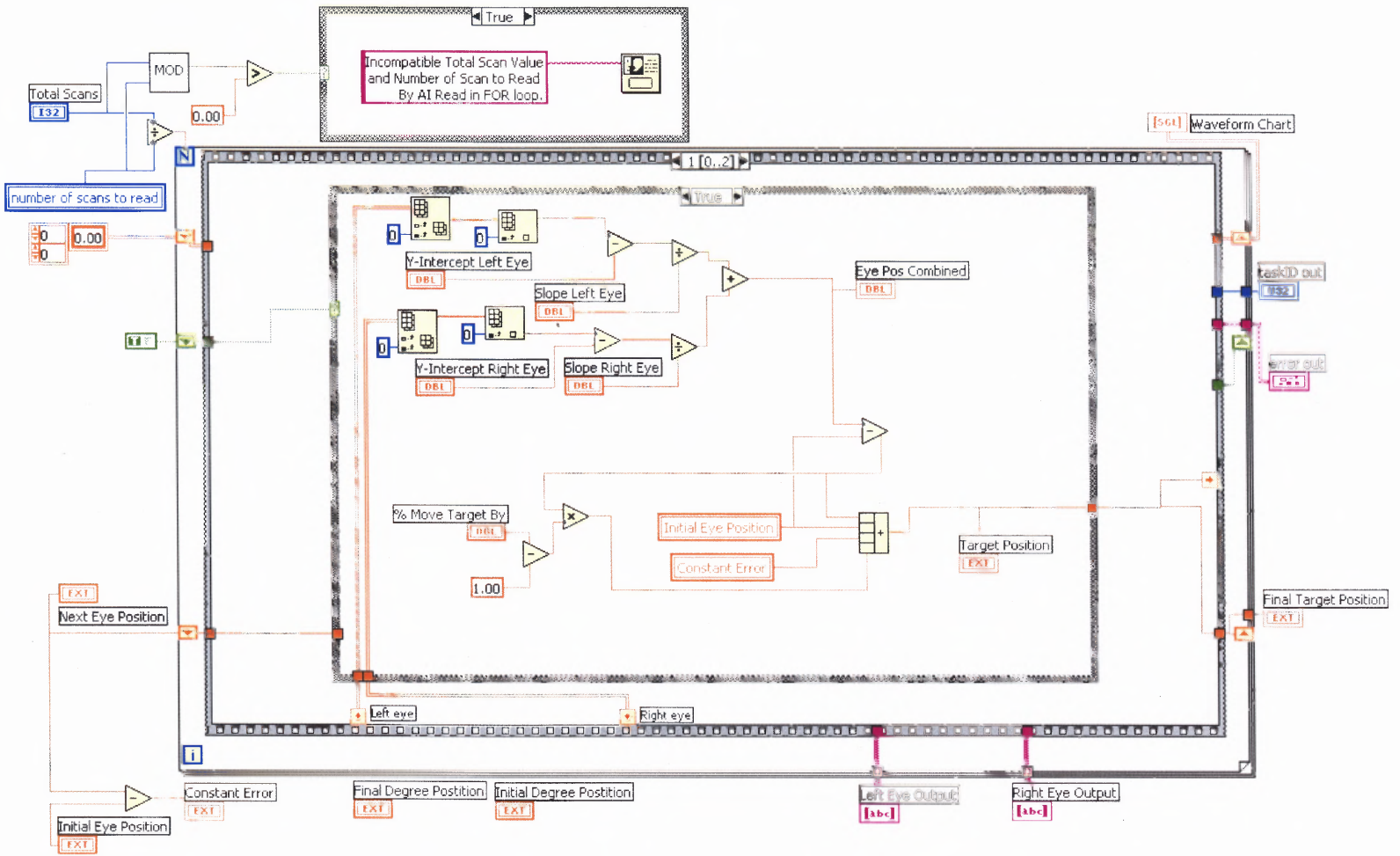
To ascertain the validity of equation one, values were plugged into W, X, Y, and Z to compare with the values in tables 4.1, 4.2, and 4.3. For instance, table 4.1 is for when the oculomotor system should act as a closed loop step response and the target should only make an initial movement and not change based on the current position of the eye. In table 4.1, a step of four degrees was presented to the subject starting at the

initial position of two degrees and ending at six degrees. If the eyes are currently at five degrees, the values  $W = 2$ ,  $X = (5-2) = 3$ ,  $Y = 3*(0-1) = -3$ , and  $Z = 6-2 = 4$  would give the following position of the target:  $NTP = 2 + 3 + (-3) + 4 = 6$  degrees. The target has not moved.

Taking a look at table 4.2, in which the target is slowing down and allowing the eyes to catch up, the same formula can be applied. If the same conditions follow for this example as in the previous example except the amount of feedback allowed is set to sixty percent, then the values for when the eyes are at five degrees are as follows:  $W = 2$ ,  $X = (5-2) = 3$ ,  $Y = 3*(.6-1) = -1.2$ ,  $Z = 6-2 = 4$ . The new position of the target would appear at 7.8 degrees. According to the table, this is the correct location of the target.

The final table, table 4.3, was also further studied. This table presented the target with full open-loop conditions in which the error between the eyes and the target were kept constant. If the same condition followed as in the previous two examples, except feedback was now set to one hundred percent and the eyes were currently positioned at five degrees, the values would follow:  $W = 2$ ,  $X = (5 - 2) = 3$ ,  $Y = 3*(1-1) = 0$ ,  $Z = 6 - 2 = 4$ . The target would appear at 9 degrees according to both the equation and the table.

Figure 4.18 Open-Loop methodology three.



This equation was tested on a variety of conditions of varying step size, initial position, and amount of feedback. For each of the conditions, a table was created and the equation was compared with it to determine the accuracy of the values. In each of the cases, the equation matched the value in the table when simulated with identical conditions. This equation worked very well for convergence movements, so question arose on whether target positions could be obtained through this equation for divergence movements. The outcome of the equation under divergence movement was successful; the equation values matched the values in the table.

#### **4.3.4 Optimal Methodology**

Both methodologies two and three are sufficient in obtaining open-loop data. They are able to complete an analysis of the eye movement in real time and determine the next position of the target. However, methodology three does offer an advantage over methodology two in its mathematics and memory usage.

Mathematics is a very powerful language for communication. It is able to communicate across languages and give the same result no matter who applies the technique. Many advanced algorithms are written mathematically to allow for completion of the program in a timely manner. Programming languages used to compile the algorithm and then parse through the code are largely mathematically driven. Writing a program in mathematics aids the compilation process when it is attempting to optimize the algorithm. Another advantage of applying the mathematical technique to the open-loop algorithm is in its usage of memory. With methodology three, information from the previous run did not have to be stored or analyzed to determine the next position of the target. This saves both time and space which another variable or operation could be

utilizing. Although this may be a small factor to consider with currently advancing technology, in real time systems every infinitesimal detail counts.

#### **4.3.5 Programmer Error**

During development, one of the major errors implemented in the code by the programmer was during analysis of the current eye position. During a vergence response, the positions of the eyes are summed for analysis. This is the nomenclature established by the vision research community, to date. The summation of right eye position and left eye position was also adhered to by all code in the Vision and Neural Engineering Laboratory at NJIT. During programming of the three methods, the right eye position and left eye position were summed, but then divided by two to find the average location of where the eyes currently appear. This average position was passed to the code, which determines the next position of the target.

A problem arose when displaying the next stimulus target to the subject. The sum of the positions of the eyes had to be passed and not the position of the individual eyes or the average of the two eyes. For instance, when 100% feedback is removed from the oculomotor system in order to maintain a fixed error in the amount of the initial step size, the target must move the same distance the eyes move at the same rate. Thus, if the right eye is positioned at degree two and the left eye is positioned at degree two and there was an initial step size of four combined, then the target should appear at eight degrees total or 4 degrees for each eye. With the bug in the code, the target actually appeared at six degrees and caused inconsistencies in the data acquisition and analyses. This problem was corrected in all three methodologies.

## CHAPTER 5

### METHODOLOGY OF ACCESSORY COMPONENTS

Several accessory components exist to assist the open loop in its design and functionality. Each component is programmed with a specific task, which is either integrated into the open-loop code or the subvi that is called upon when needed. These components help modularize the code for easy understanding of the flow and to allow future additions.

#### 5.1 Variable Initialization

##### 5.1.1 Front Panel and Text Variables

At the beginning of the experiment, all the variables must be initialized for program functionality, data acquisition, and analysis. The operator must be provided a method to accomplish this task rapidly and with accuracy. Figure 5.1 is an image of the front panel displaying all the variables. When the program first starts, a file dialog window appears asking the operator to enter the location and file name to where all the data should be saved during experimentation. At this same time, one of the most important tasks is to clear the data acquisition (DAQ) board from any previous information which may remain due to other programs that may have utilized the (DAQ) card in the past. This supports data integrity and program functionality.

The next task for the operator is to enter the subject's initials, experiment information, and date into the *Subject* Header, which gets stamped to every experimental trials during open-loop execution. The next variable determines how many experimental trials are given per calibration sequence. For open-loop, calibration takes place before every experimental trial. During calibration, the subject is presented with three position

degrees to fixate on in sequence, and the voltage values are recorded at each of these positions. The degree positions presented to the subject during calibration are controlled through the variables called *3 Point Initial Position*, *3 Point Second Position*, and *3 Point Last Position*.

The next two variables are very important since they are used through out the program extensively. These are the *Initial Degree Position* and *Final Degree Position*. Going back to subsections 2.5.2 and 2.5.3, the computer must be able to take two known degree positions and correlate them to two pixel positions for use throughout the program to determine all the unknown pixel positions for the corresponding degrees. These two known degree positions are used to match the targets presented on the monitor with the real world targets. This allows the computer to determine the pixel position of every degree position entered by the subject.



Figure 5.1 Variable initialization front panel.

Variable Initialization
Eye Movement Device Setup
3 Point Calibration
Open-Loop Data Collection
Programmer Variables

Subject Header

lapd\_openloop\_date

Calibrate every  time(s)

5 Point On or Off  Calibrate Only One Time

On  Yes

Off  No

3 Point Initial Position

3 Point Second Position

3 Point Last Position

Initial Degree Position

Final Degree Position

Number Of Stimuli

Calibration Time

Main Program Time

**% Move Target By**

Version 2: Uses previous and current eye position to determine next target position  
 Version 3: Used only current eye position to calculate next target position

		Field1 Lt E	Field1 Rt E	Bias Lt E	Bias Rt E	Field2 Lt E	Field2 Rt E
Stimulus 1	Open Loop V3	0.00 1	2.00 1	1.00 1	1.00 1	5.00 1	5.00 1
Stimulus 2	Open Loop V3	0.00 2	0.00 2	0.00 2	0.00 2	0.00 2	0.00 2
Stimulus 3	Open Loop V3	0.00 3	0.00 3	0.00 3	0.00 3	0.00 3	0.00 3
Stimulus 4	Open Loop V3	0.00 4	0.00 4	0.00 4	0.00 4	0.00 4	0.00 4
Stimulus 5	Open Loop V3	0.00 5	0.00 5	0.00 5	0.00 5	0.00 5	0.00 5
Stimulus 6	Open Loop V3	0.00 6	0.00 6	0.00 6	0.00 6	0.00 6	0.00 6
Stimulus 7	Open Loop V3	0.00 7	0.00 7	0.00 7	0.00 7	0.00 7	0.00 7
Stimulus 8	Open Loop V3	0.00 8	0.00 8	0.00 8	0.00 8	0.00 8	0.00 8
Stimulus 9	Open Loop V3	0.00 9	0.00 9	0.00 9	0.00 9	0.00 9	0.00 9
Stimulus 10	Open Loop V3	0.00 10	0.00 10	0.00 10	0.00 10	0.00 10	0.00 10

Completed Variable Initialization

### 5.1.2 Calibration Switches

On the right side appears four switches. The first switch controls three-point calibration. This switch is able to turn calibration on or off, but this parameter cannot be changed once the open-loop code execution has begun. The next switch allows calibration to take place, but only a single time during initial execution of the program right before open-loop proceeds. This was used to debug the code as well as provide the operator in future studies with greater flexibility on how the three-point calibration is utilized.

The third switch has a very simple task. During execution, the operator should be aware when calibration begins and ends. This switch, when set to 'on', informs the operator when calibration is about to proceed. The final switch allows two options to be presented to the subject. This switch is responsible for how the trigger button is employed in the three-point calibration. Does the user have control over when data acquisition begins during calibration or is this under computer control? When the switch is set to on, the user is able to control when data acquisition begins, otherwise it defaults to computer control.

Setting the switch to off enables all three degree positions to be presented to the subject one after another without a break in between. The other option allows the user to control when each degree position is presented to him or her. This switch makes it possible for the presentation of the targets to be controlled either by the user or the computer. As shown, calibration is a critical aspect of an open-loop experiment. Without it a real time data analysis could not be completed successfully. The number of options provided for calibration allows proper program execution to make the subject more comfortable along with acquiring eye movement data for the researcher.

### **5.1.3 Stimulus Presentation, Timing, and Move Target By**

At the bottom of the panel are several arrays where the operator can enter the stimulus to be presented to the subject. A minimum of one stimulus must be provided and a maximum of ten stimuli can be given to the open-loop program. The computer will randomly choose which stimulus is presented next, which is further discussed in subsection 4.1.1. The variable, *Number Of Stimuli*, is used by the program as to how many stimuli are available to choose from. The last three variables needed to be assigned are the *Calibration Time*, which determines for how many seconds a selected target should appear for and how long eye movement data should be recorded. *Main Program Time* controls the experimental trial time length and data acquisition time for each experimental trial of open-loop execution. The variable, *% Move Target By*, is used by the open-loop code to determine how much feedback is allowed to be introduced to the oculomotor system.

### **5.1.4 Complete Variable Initialization Button**

The last step before moving on to the phase of the program, which is eye movement monitor initialization and degree to pixel correlation that are discussed in subsections 2.5.2 and 2.5.3, is to press the *Completed Variable Initialization* button. This step is needed because the program is compiled and when it is executed in the Microsoft Windows® operating system, the program automatically assumes all the variables have been initialized and can proceed to the next stage. This button gives the operator time to initialize all the variables and then proceed.

## 5.2 Save to File

For analyzing data at a future time in Matlab<sup>®</sup>, each run must be saved to a data file with all the parameters and variables used through out the program for each experimental trial. A large array is created with all the information gathered during the initial setup stage, three-point calibration, and open-loop responses from the left and right eye to save into a data file, Figure 5.2. Before entering information into the array, the parameters and variables must be converted from the format they appear in such as integer, double, character, byte, and so forth, into a string. In the header array, the subject header is composed of the following information: total number of scans, the current date and time, the random delay before presentation of the stimulus, the stimulus given to the subject, the stimulus parameters, the percent the target moves by when digitizing eye movement responses during the real time analyses, and the final target position in degrees. The next third of the header array takes in the calibration mean voltage, standard deviation, and the corresponding degree position for the left eye. The last third of the header array accomplishes the same task as the second third, except for the right eye. This header array is then fed into another build array and adds the left eye scans as the second input and the right eye scans as the third input. This array is then sent into the Write File function and saved to the specified data file entered during initial variable setup.

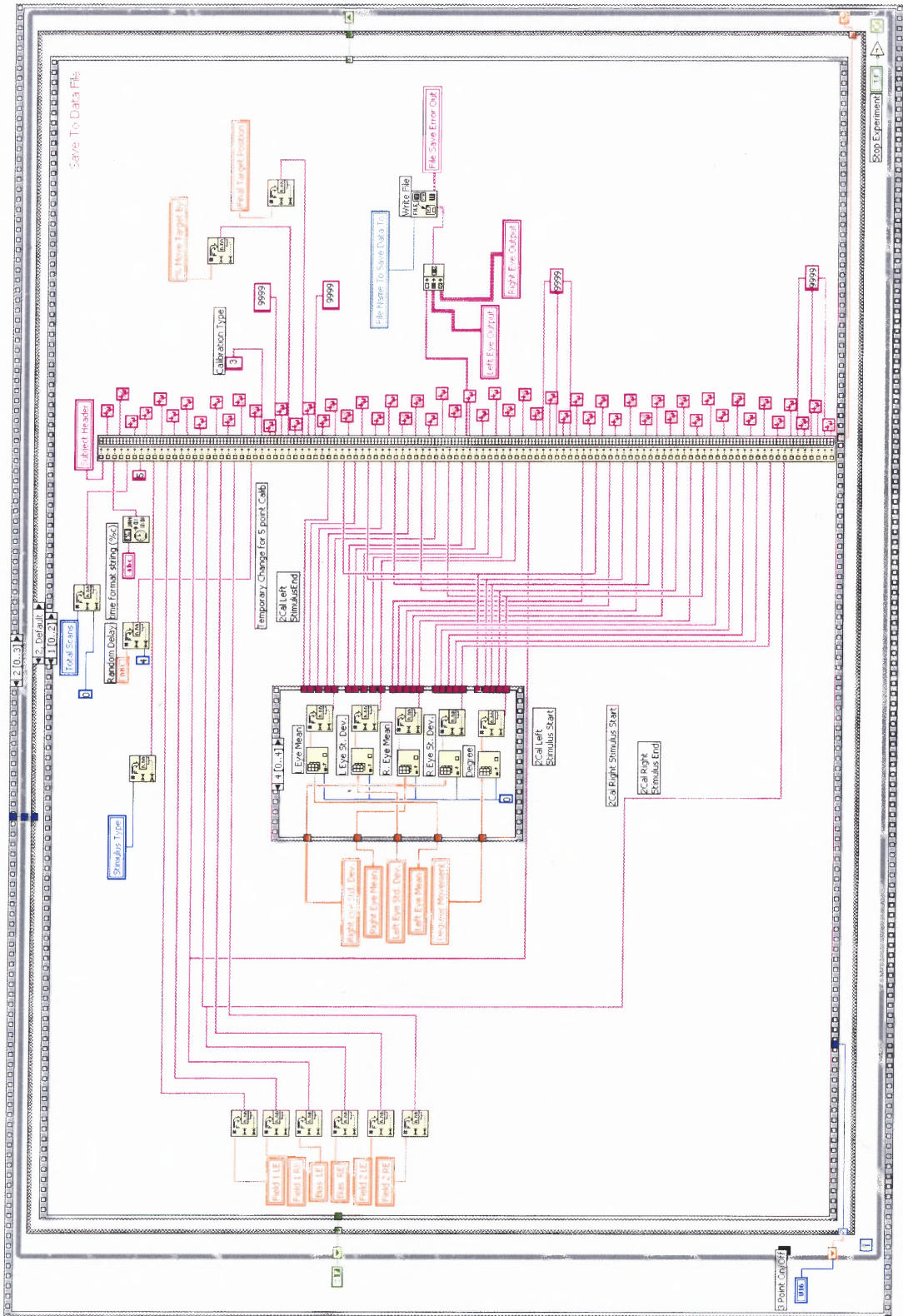
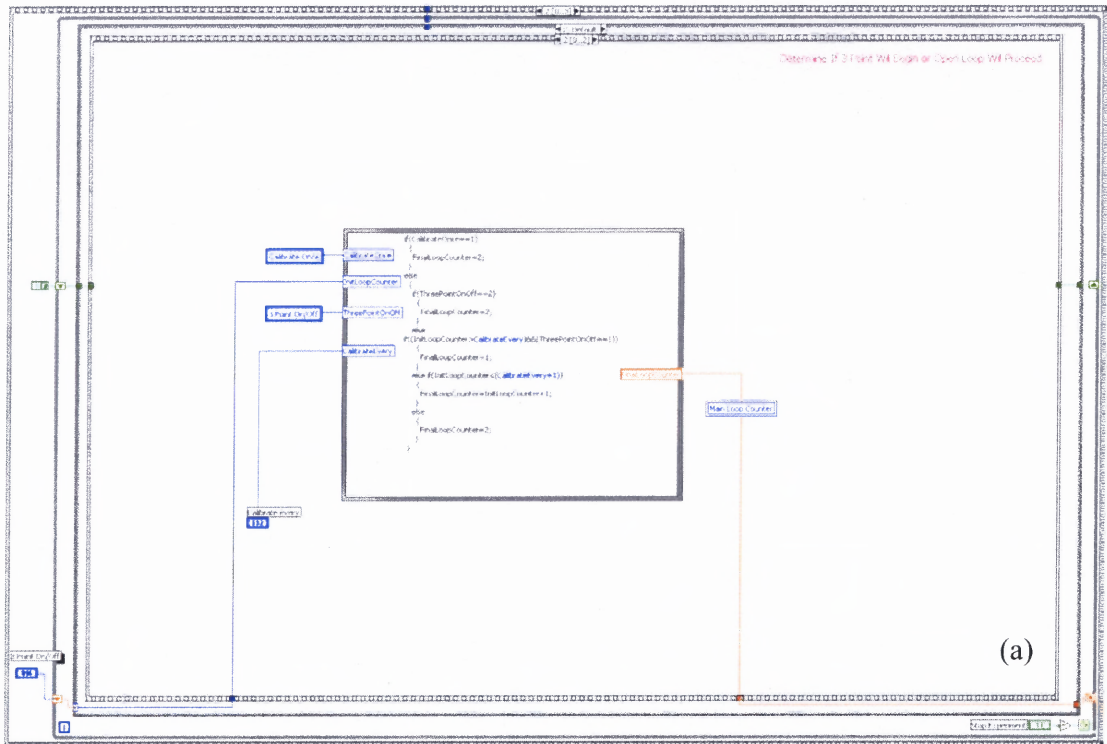


Figure 5.2 Save to data file.

### 5.3 Determine to Calibrate Or Not

After the storage of data to a file, a decision is made whether to proceed with three-point calibration or to continue to open-loop. Figure 5.3 displays the code which determines how the operator would like to proceed through the variables entered during the variable setup stage. Whether three-point calibration is executed or not is controlled through a single registry shift shown at the bottom left and right in the wall of the for loop, the orange arrow. The registry shift works by taking an initial value from outside the loop and allowing the first sequence of this loop to use this information. Next, within the loop the value could be changed and the new information could be passed to the next iteration of the while-loop for use. This registry shift controls which case statement is activated, whether it is three-point calibration or open-loop.

The next value in the shift register to pass to the next iteration of the while-loop is determined through four variables, *CalibrateOnce*, *InitLoopCounter*, *ThreePointOnOff*, and *CalibrateEvery*. If the variable *CalibrateOnce* is equal to one, which means to calibration already took place once and is not needed again throughout the program, then the output variable *FinalLoopCounter* is informed that no calibration is further needed and a value of two is set. The value of two in the *FinalLoopCounter* is fed into the shift register thus case statement two, the open-loop code is selected in the next iteration.



```

if(CalibrateOnce==1)
{
    FinalLoopCounter=2;
}
else
{
    if(ThreePointOnOff==2)
    {
        FinalLoopCounter=2;
    }
    else
    {
        if((InitLoopCounter>CalibrateEvery)&&(ThreePointOnOff==1))
        {
            FinalLoopCounter=1;
        }
        else if(InitLoopCounter<(CalibrateEvery+1))
        {
            FinalLoopCounter=InitLoopCounter+1;
        }
        else
        {
            FinalLoopCounter=2;
        }
    }
}

```

(b)

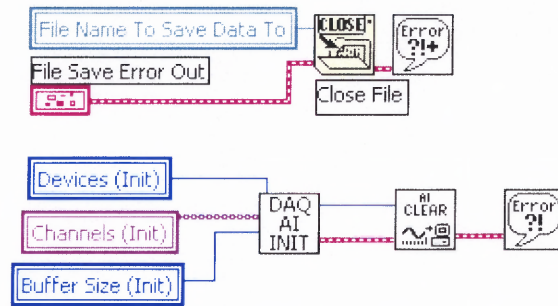
**Figure 5.3** (a) Code to determine what should proceed, calibration or open-loop. (b) Close-up of text in code.

For all other values of *CalibrateOnce*, the other variables are tested. The next variable to be tested is *ThreePointOnOff*. *ThreePointOnOff* is assigned either a one or a two. The value of two indicates the calibration remain off and a value of two is sent to the *FinalLoopCounter*. In the situation the *ThreePointOnOff* variable is set with the value of one, the *InitLoopCounter* variable is compared with the *CalibrateEvery* variable. When the *CalibrateEvery* variable is set to one, the code execute a three-point calibration before every run of the open-loop. If the *CalibrateEvery* variable is greater than one, for instance if it is five, then calibration proceeds once and is followed by five runs of the open-loop experimental trial. For all other situations not seen or determined by the programmer, the *FinalLoopCounter* defaults to a value of two in which the open-loop program experimental trial is executed and the three-point calibration is not initiated before the experimental trial.

#### 5.4 Exit Program

Upon exiting the *Open-Loop Program 1.0* two items must be accomplished to maintain the stability of the computer file structure, the operating system, LabVIEW operation, and the DAQ card. Figure 5.4 displays the file close subvi which takes as an input the file name which was entered during program variable initialization and the error code which has been passed through out the open-loop program. As long as no errors exist, the error subvi will popup with no message and the file structure stability is maintained. The last item initializes the DAQ card with the required device number, channels, and buffer size to reset and clear it in the end. This helps maintain the operational stability of the DAQ card, operating system, and LabVIEW.



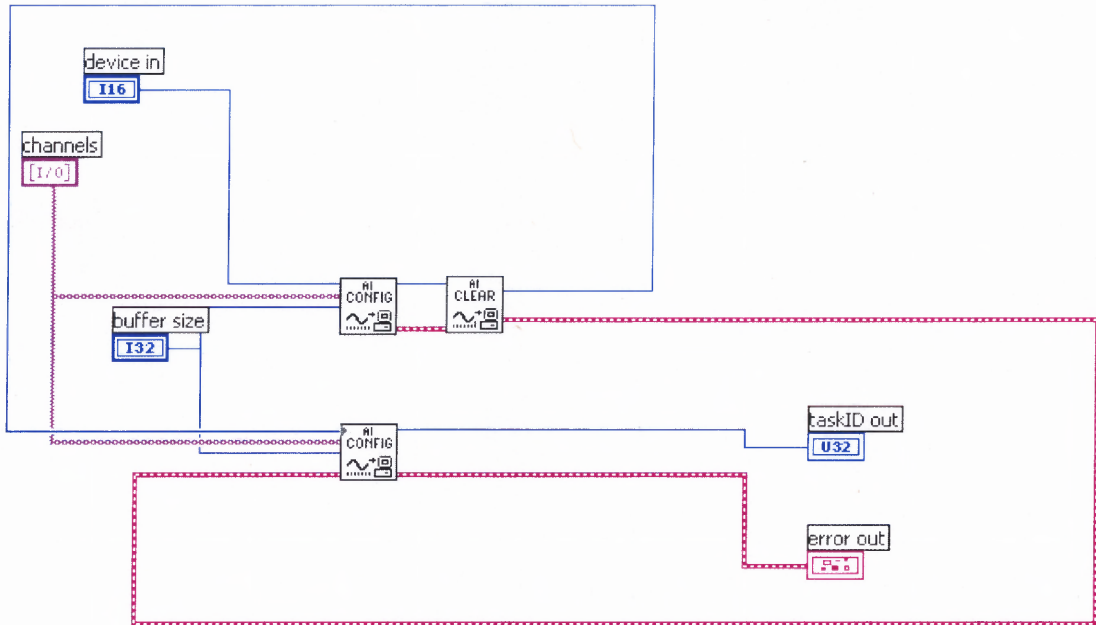


**Figure 5.4** Close data file and clear DAQ card.

## 5.5 SubVI's Created for Modularization and Enhancement of Architecture

### 5.5.1 DAQ Initialization

The DAQ initialization code is a subvi which can be called anytime during program execution, Figure 5.5. This subvi provides an important role in clearing the DAQ card and preparing it for data acquisition. Clearing the DAQ card is essential since it clears the buffer of any previous data. The subvi takes three parameters as input, the device number indicating which DAQ card to utilize, the channels from which data will be acquired, and the buffer size. These parameters are first passed into the AI Config vi and then the AI Clear vi, thus clearing and reinitializing the buffer. The last step is to reconfigure the AI Config vi with the same parameters for use with data acquisition and to pass the taskID containing the device number and the error code out.



**Figure 5.5** DAQ Initialization.

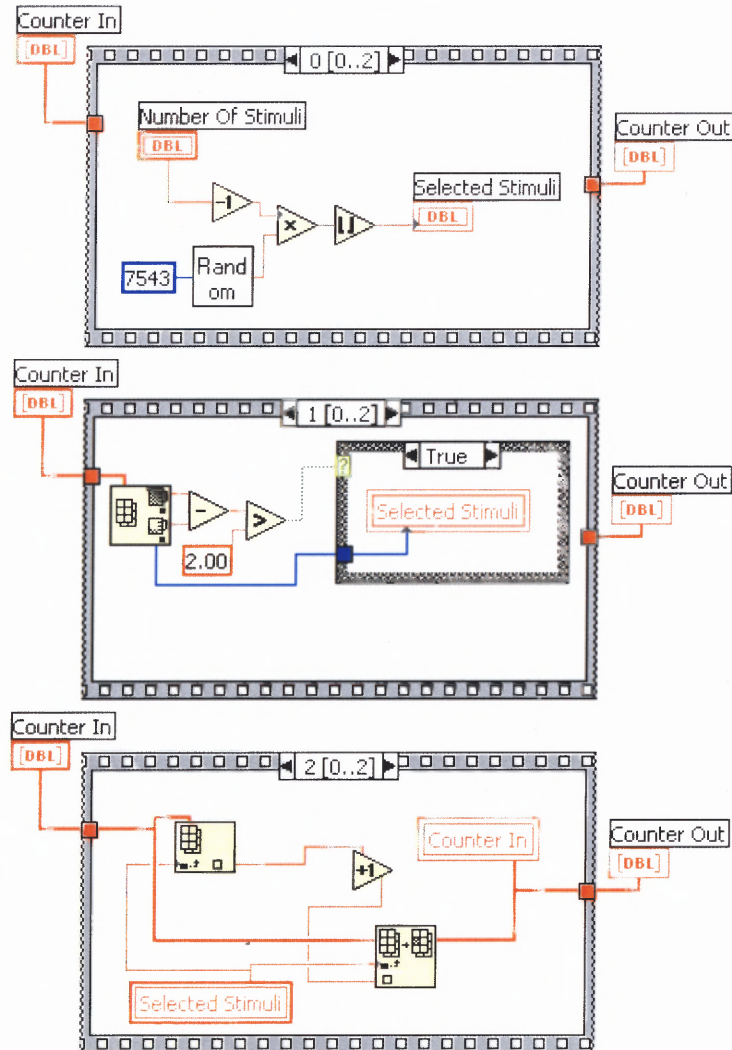
### 5.5.2 Stimulus Counter Organizer

The stimulus counter organizer has two tasks, Figure 5.6. The first task is to keep track of how many times a particular stimulus has been called upon. The next task is to select the particular stimulus to present to the subject. This is accomplished through three sequences and the input of two parameters. The first parameter is an array which contains how many times a particular stimulus has been called. The second parameter is the total number of stimuli the program can chose from, determined by the operator.

The first sequence is responsible for selecting the particular stimulus. This is completed by taking the number of stimuli and subtracting it by one because the array positions start from zero. A random number generator is called upon, discussed in subsection 5.4.3 to select a stimulus at random. This stimulus is then passed to an output variable called selected stimuli.

The second sequence verifies that the selected stimulus does not differ by more than two between the count of any other stimulus in the array. This is completed by finding the difference between the min and max values and comparing it with the digit two. When the difference is greater than two, the position of where the min value appeared is passed into the selected stimuli variable. This is to ensure that approximately equal amounts of stimuli are presented during an experimental session.

The final sequence increments the counter by one in the position which the selected stimuli points to. This is completed by first taking the input counter array and finding the value at the specified position in the array using the index array function. The value returned from the array is then incremented by one and inserted back into the same array at the same position. This array is then sent into the counter out variable and displayed on the front panel.

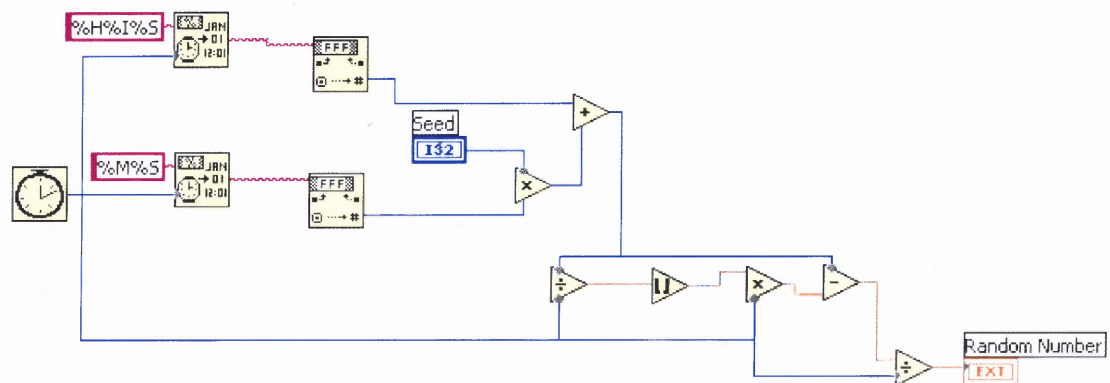


**Figure 5.6** Stimulus counter and stimulus selection.

### 5.5.3 Random Number Generator

The random number generator provided by LabVIEW is good, but always manages to favor a single stimulus. Due to this problem, a random number generator, Figure 5.7, was created utilizing a seed and the software clock. When given the proper seed, the random generator does not favor a particular stimulus and allows the runs to be more randomized. The method in which this technique works is as follows.

The software clock is used to obtain two digits. The first digit calculates the software clock as a 24 hour clock, a 12 hour clock, and in seconds, then puts these numbers together to obtain a new value. The second digit uses the software clock in minutes and seconds to come up with the new value. The second digit is then multiplied by the seed and added to the first digit. Lets call this value large denom. Next, the large denom is divided by the first digit and the floor function is called to remove the decimal point. This value is then multiplied by the first digit and subtracted from the large denom. Lastly, this value is then divided by the first digit, thus giving a value between zero and one inclusive. For many values of the seed, the random number generator acts as the one provided by LabVIEW, but there are a few values for the seed which provide an improvement. This has to be further studied to find the optimal seed value.



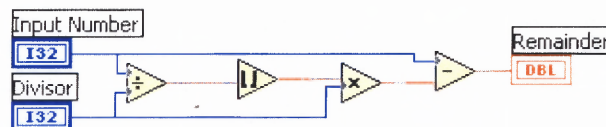
**Figure 5.7** Random number generator.

Another reason to why a stimuli is favored, which does not include the outer stimuli in they counter array is in the method in which the counter selection portion of the code has been written. When the operator requests the code to choose from four stimuli, the random number generator generates a value between one and four. When the value is between 1.0 and 1.5, stimulus one is selected. When it is between 1.5 and 2.5, stimulus

two is selected. A value of 2.5 to 3.5 selects stimulus three and a value between 3.5 and 4.0 selects stimulus four. The outer two stimuli, stimulus one and four, have only a fourteen percent chance each of being selected. Whereas stimulus two and three each have a probability of 33 percent of being selected. This tends to favor selection of the inner two stimuli for this case.

#### 5.5.4 MOD Function

This function is used to find a remainder when an input integer is divided by another integer, the divisor, Figure 5.8. This is accomplished by first taking the first number and dividing it by the second number and finding the floor of that number. This number is then multiplied by the divisor and subtracted from the first number, thus obtaining the remainder. It is a very simple mathematical task, which LabVIEW has not provided.



**Figure 5.8** MOD - Remainder.

## CHAPTER 6

### EXPERIMENTAL METHODOLOGY

The goal of this study was the creation of a program, *Open-Loop Program 1.0*, to allow for the acquisition of the fast preprogrammed initiating component of a vergence eye movement. *Open-Loop Program 1.0* flowed through six stages, listed below, each of which is important in its task.

1. Variable Initialization
2. Eye Movement Equipment Setup
3. Three-Point Calibration
4. Open-Loop Stimulus Presentation and Eye Movement Recording
5. Save all Data to File for Offline Analysis
6. Exit

To verify the three-point calibration and the open-loop code in *Open-Loop Program 1.0*, an experiment was conducted on two subjects who have normal binocular vision. Both subjects were experienced in eye movement research and were aware of the goal of this study. An analysis was completed on the linearity of the three-point calibration and a comparison completed of the data acquired during open-loop with data acquired by previous researchers [1, 25] whom have gathered open-loop vergence data under similar methodologies. Two experiments will be run under open-loop program conditions. The first allows the target to move one hundred percent of the total eye movement maintaining a constant disparity. This “opens” the feedback loop for the feedback portion of the vergence system. The second moves the target sixty percent of the total eye movement and lets the eyes “catch up” with the target.

## **6.1 Utilized Instrumentation and Software**

### **6.1.1 Data Acquisition Instrumentation**

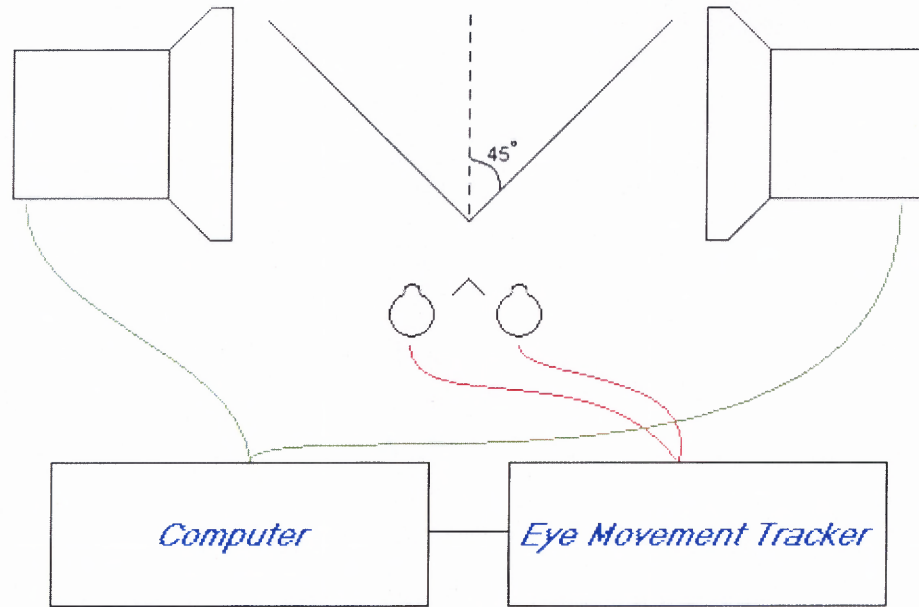
Vergence eye movements were collected using the infrared Skalar IRIS Limbus Tracking device model 6500. The device has a linearity of  $\pm 25$  degrees with a 2 minute arc resolution [10]. The limbus-tracking device is placed on the person's head and is connected to the eye movement tracker which is integrated with the computer through the Open-Loop program. The eye movements are sampled at a rate of 200 Hz, which is well above the Nyquist Sampling Rate and saved to a data file. Eye movements are collected and stored from each eye independently. A Dell Optiplex GX240 Pentium 4 computer with 256 MB RAM and Windows XP Professional 2002 computer will be used throughout data acquisition and analysis.

### **6.1.2 Visual Instrumentation**

The setup of the instrumentation is depicted in Figure 6.1. There were two partially reflective mirrors placed in front of the subject. The mirrors were approximately forty-five degrees apart from the subject's line of sight. The eye movements were sampled by the limbus tracking device placed on the subjects head. They then passed through the eye movement tracker for amplification of the signal and into the computer.

The computer sampled the incoming signal from the eye movement tracker and at the same time displayed the stimulus to the subject on two computer monitors placed on either side of the partially reflective mirrors. The two computer monitors utilized are manufactured by ViewSonic with model number E70fb. They are 16 inch (12 inches or 32.4 cm wide) color flat screen monitors. The stimulus shown on the monitors is reflected back to the subject through the use of the mirrors.





**Figure 6.1** Instrumentation Setup

### 6.1.3 Software

LabVIEW 6i was essential to *Open-Loop Program 1.0*. The code for conducting an open-loop experiment was written in the LabVIEW language. LabVIEW was responsible for conducting a real-time data analysis after acquiring the three-point calibration data to determine the location at which the stimulus should be displayed to the subject and digitizing all eye movement data.

Once all the eye movement data had been acquired and saved, it was analyzed in Matlab. Matlab utilized the three-point calibration and converted the voltage values read in through the eye movement tracker into degrees of movement. Matlab was essential in displaying the three-point calibration, position and velocity traces of each eye, the position and velocity traces of the combined response, and the main sequence. It also made analyzing the data simpler by finding the peak velocity and main sequence ratio for each eye movement. The main sequence is a common index used by eye movement

researchers to quantify the dynamics of eye movements [4]. The main sequence is the ratio of peak velocity to response amplitude.

## 6.2 Experiment Setup and Layout

Before the subject arrived for the experiment, several items must be accounted for and set up for experimentation to go as quickly as possible. This enabled the subject to experience less fatigue during experimentation and acquire more data during the experiment. This also allowed the operator to have more control during experimentation and not to forget any detail. There are two items to setup before subject arrival, the experimentation cubicle and the parameters in *Open-Loop Program 1.0*.

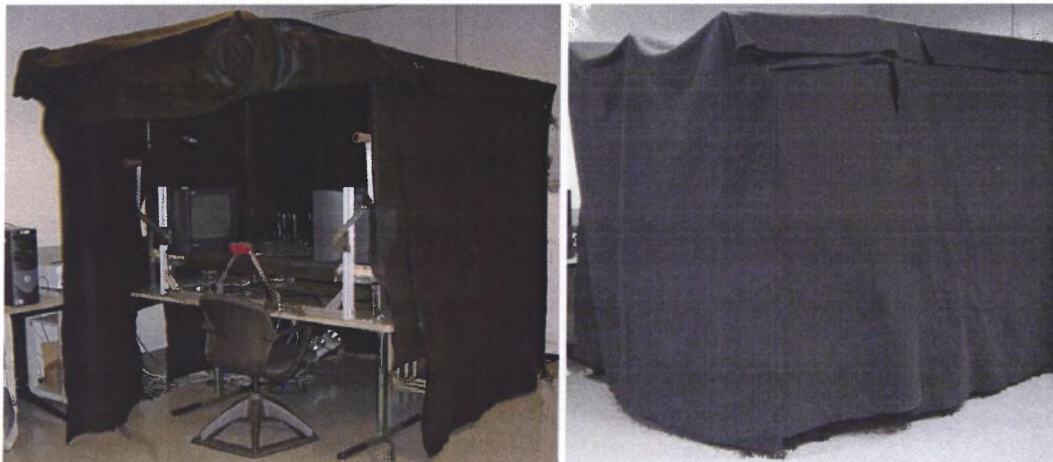
### 6.2.1 Experimentation Cubicle

Within the experimentation cubicle, shown in Figure 6.2, many physical items needed to be setup prior to subject arrival. To suppress subject head movement or input from the vestibular system as much as possible during experimentation, a chinrest was manufactured with subject comfort in mind, Figure 6.3. The chinrest was adjusted to keep the person eyes at the same level of the target and centered to the subject's line of sight. There are two bars onto which the subject is able to rest his or her chin and two additional bars are placed on the cheeks to suppress head movement and allow stabilization. On the chinrest were placed foam pads for added comfort and hygiene. The pads were replaced for every new subject.

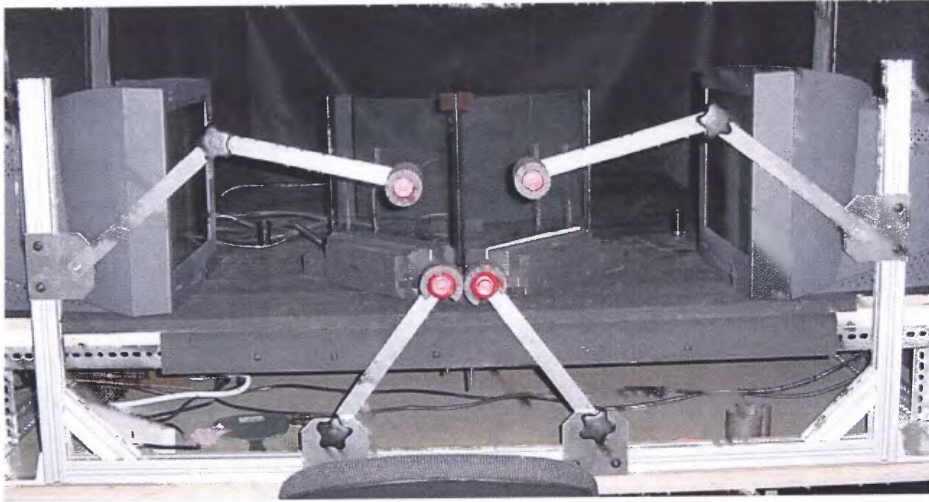
The next item was the lighting in the room. During experimentation, all the lights in the room are shut off in order to keep the subject in the dark. This allowed the subject to only look at what the operator has requested and presented to the subject and removed

all distractions. For this reason, the cubicle was surrounded by a black cloak to minimize the light entering from the operator computer monitor. The light coming from the windows was eliminated through a thick black plastic bag taped to the window.

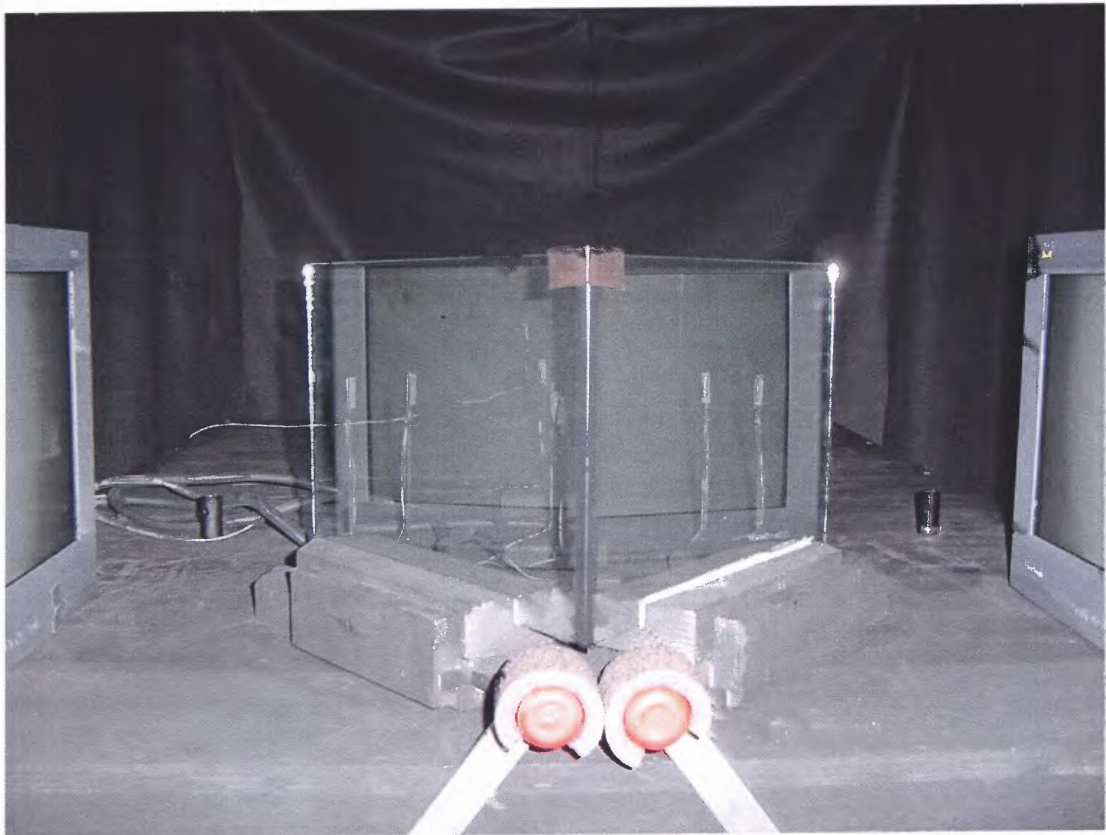
The last important item was the proper orientation of the mirrors, Figure 6.4. The mirrors had to be placed in the position which when presented with the target by the monitors matched the LED targets at the known degree positions and a board in the back indicating the current degree position. All three should overlap one on top of the other. The initial targets used for adjustment of the mirrors appeared at two degrees and five degrees for each eye during the eye movement tracker setup stage.



**Figure 6.2** Experimentation cubical.



**Figure 6.3** Chinrest



**Figure 6.4** Mirror Orientation

### 6.2.2 Program Parameters

The next step before subject arrival is to start the initialization of the *Open-Loop Program 1.0* with the proper parameters determined by the type of experiment to be run. Two experiments were run under open-loop conditions, one which moves the target by 100 percent and the second which moves the target by 60 percent of the eye movement. Common to these two experiments are all the parameters shown in Figure 6.5 except the parameter *% Move Target By*.

The subject header is first initialized with the subject's initials, the type of experiment, the date, and additional information the operator deems necessary. This is stamped to all the experimental trials in the data file.

The second section of parameters determines the position of where the target appears during three-point calibration. The experiment run for verification ran the three-point at two degrees, six degrees, and ten degrees. They appeared in the initial position, second position, and final position boxes, respectively.

The third section of parameters is utilized for mirror orientation. The computer presents a target oscillating back and forth which are to be matched with the real world targets presented at known positions as well as projected to a board in the back indicating the current degree position. The two known positions are entered into the initial target position and final target position entry boxes. The open-loop program uses the real world targets appearing at four degrees and ten degrees to match the oscillating lines on the computer monitors.

The fourth section is stimulus setup shown at the bottom of Figure 6.5. This controls the type of stimulus to present to the subject through a drop down menu and the

corresponding parameters. The operator is able to enter in multiple stimuli to present to the subject randomly to avoid subject prediction. The open-loop experiment presented only one stimulus to the subject with the parameters 2.00 in Field1 Lt E and Field1 Rt E, 1.00 in Bias Lt E and Bias Rt E, and 5.00 for Field2 Rt E and Field2 Lt E where Lt E stands for left eye and Rt E stands for right eye. The Bias box contains the parameters for the initial position of where the target first appears to each eye before the subject clicks the button to initiate the open-loop run. The Field1 box contains the parameters of the initial step size presented to each eye after the trigger button is pushed. The last parameter, Field2, stores the saturation point for each eye. This is the point at which the target ceases to move further.

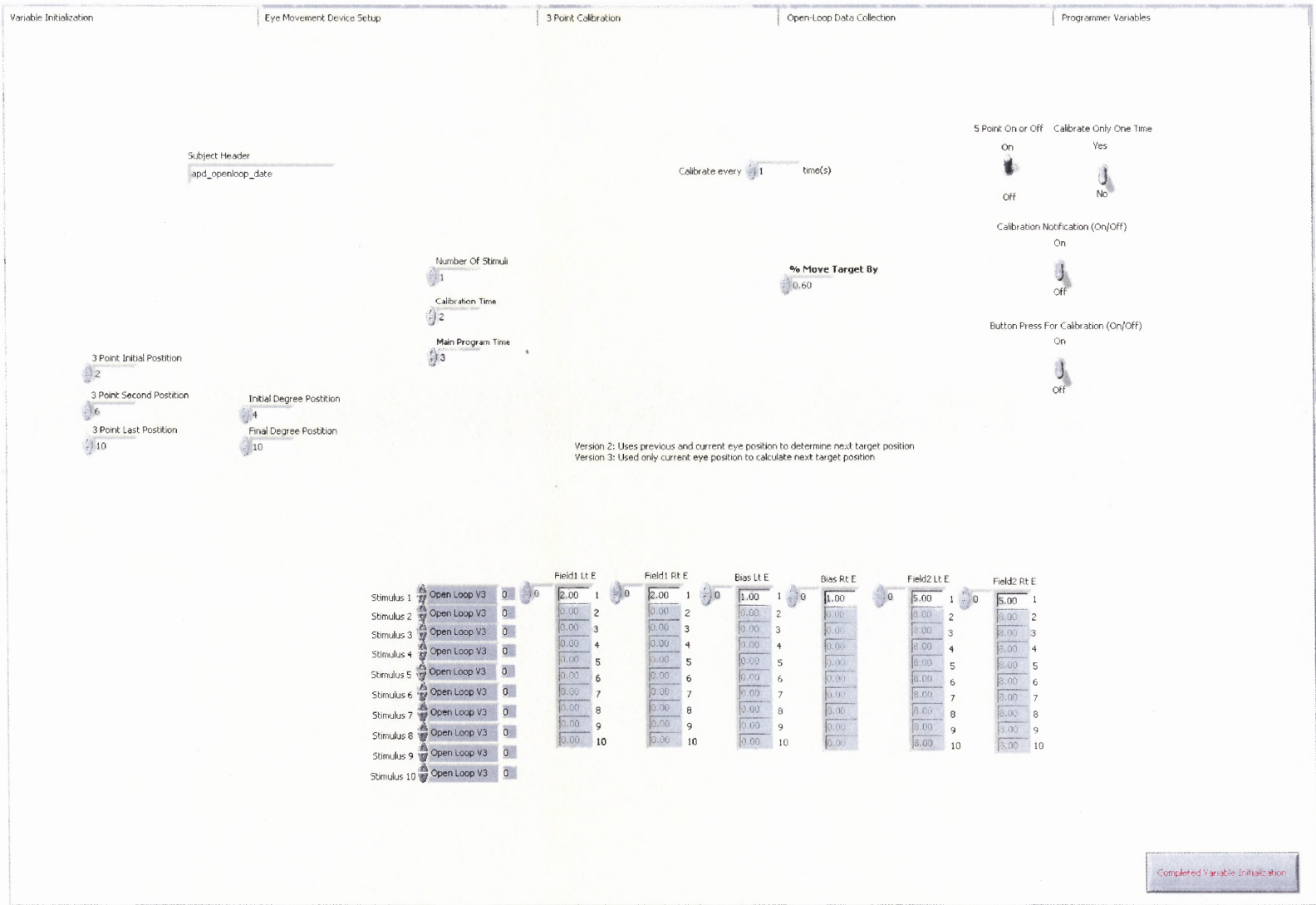
The fifth section controlled the timing of each run during calibration and open-loop and the number of stimuli to present. The calibration timing and open-loop program timing are kept separate. The target was presented to the subject during calibration for a period of two seconds and during open-loop execution for three seconds. The number of stimuli to be presented to the subject was selected as one, however up to ten could have been chosen.

The sixth section contained four switches and a parameter box responsible for three-point calibration. The parameter box was set to one to allow calibration to take place each time before the open-loop code began. The three-point on off switch was set to on and the calibrate only one time switch was set to no to enable calibration and allow calibration to take place after every experimental trial of open-loop. The third switch, calibration notification, was set to off since the operator knew when calibration begun each time and did not have to be notified. The final switch, button press for calibration,

was also set to off to allow for rapid succession of calibration stimuli through the three-point calibration and enable the computer to control at which time the target is presented to the subject for eye movement recording.

The final parameter was the percent target moves by. In the first experiment presented to the subject, this parameter was set to 0.60, indicating 60 percent partial feedback. For the second experiment run on the same subject, the parameter was changed to 1.00, indicating 100 percent representing a full open loop experiment where the error was held constant throughout the experiment. This parameter controlled the amount the target moved by depending on the eye position for each of the stimuli.

Figure 6.5 Variable Initialization Front Panel.





### **6.3 Subject Instructions for Open-Loop Experiment**

#### **6.3.1 Subject Arrival**

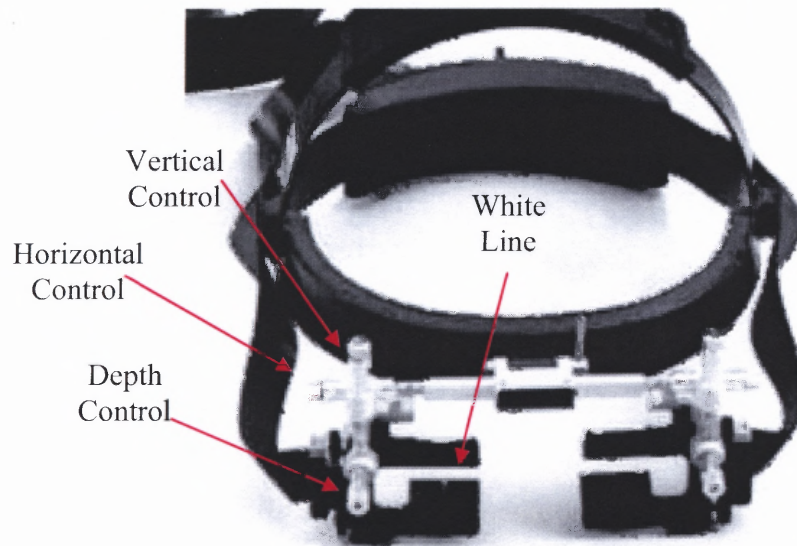
Upon arrival of the subject, the subject is first informed of the goal of the study, which is to verify the code in presenting a target to acquire the initiating component of a vergence eye movement and remove all flaws in the code. Once the subject had decided if he or she would like to continue in the research, the subject is asked if he or she has worked in a mechanical shop or has had Lasik or any other type of ocular surgery. Both of these criteria are necessary to ensure subject safety when using the eye movement monitor. The infrared light which is being emitted into the eyes and reflected back should not cause any harm, but there is always the risk of heating up any metal shards that may be within the eye or opening up the cornea from the lasik surgery, which is why this group of subjects were eliminated.

The experimental protocol is then explained to the subject. He or she is asked to sit in the subject seat and to center his / her head. To center a the subjects head the subject is asked to have a near target superimpose on project lines behind the near target. The subject is told to follow the oscillating target projected onto the mirrors and to align them up with the real world target. This is accomplished by closing the left eye to allow the right eye to see the target on the mirror. This target is then aligned with the corresponding real world target and a board indicating the current degree position. Next, the right eye is closed and the same procedure follows for the left eye. This procedure is repeated until the head is positioned in the proper position. This is explained in the beginning to save time and to allow the subject to get acquainted with the system.

After head alignment the use of the trigger button is explained to the subject as well as the procedure when experimentation begins. The subject was asked to track the target with the eyes as it moved from one location to the next. The subject was also informed that he or she will have control on when the computer begins moving the target and recording the eye movements through the use of the trigger button. During data acquisition, the subject was asked not to blink.

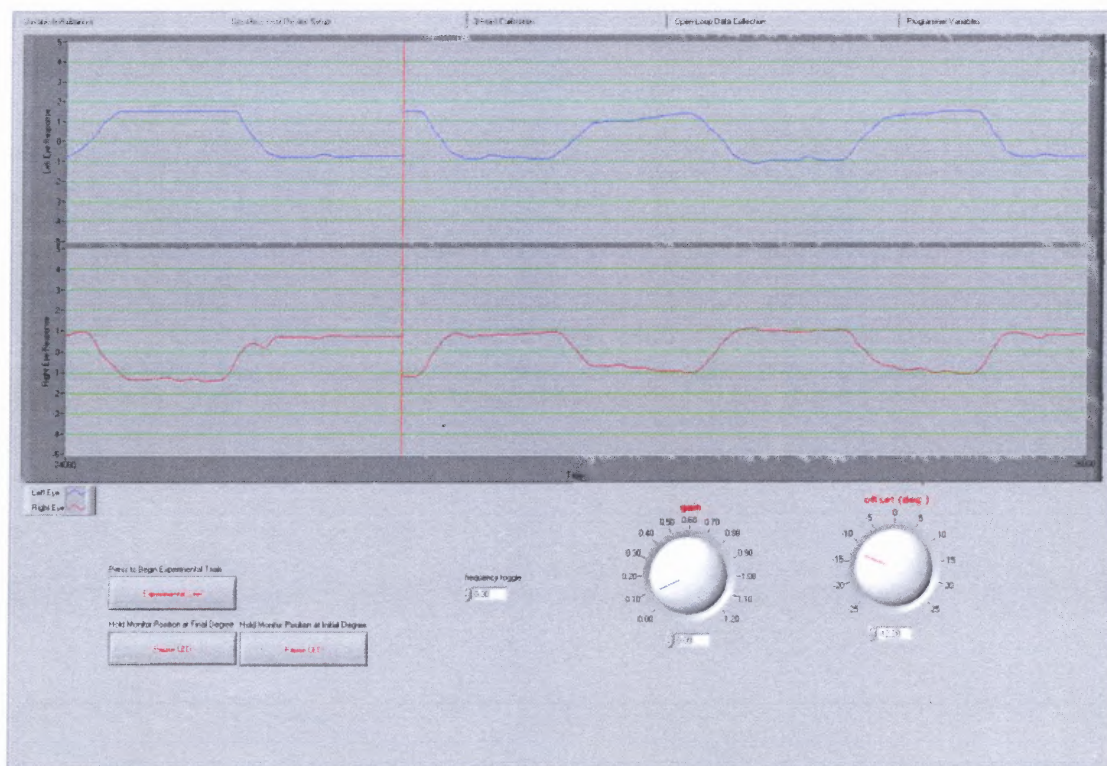
### 6.3.2 Limbus Tracking Device Setup

The next step was to place the limbus tracking device on the subject's head and adjust it to acquire a strong signal. Three knobs on each side of the eye piece control the vertical, horizontal, and depth of the infrared sensors. A white line in the center of the eye piece was lined up to the center of the subject's pupil.



**Figure 6.6** Adjustment knobs on limbus tracking device [10].

To determine if the signal acquired off the eye is strong, the subject is asked to line up their head as explained earlier and to follow the target on the mirrors as it moves back and forth between the two known positions. The operator is able to see in the movement of the eyes in real time, Figure 6.7, and determine if any adjustments need to be done to acquire a stronger signal. Once this stage is complete the operator informs the subject the experiment is ready to begin, gives him or her the trigger button, and shuts off all lights within the experimentation room.



**Figure 6.7** Data acquisition for adjustment of limbus tracking device. Top is left eye and bottom is right eye.

### 6.3.3 Experimentation

The experiment begins with the three-point calibration, Figure 6.8. During the three-point calibration, data were acquired from the eyes at two, six, and ten degrees of total vergence fixation and a regression analysis was completed. After the three-point calibration, the open-loop portion begins and presents an initial target for the subject to fixate on located at two degrees. Once the subject presses the trigger button, the target initiates a four degree step to six degrees. Starting at this point, the movement of the eyes are being tracked by the open-loop code and the target is pushed further. The rate at which the target moves is determined by how much feedback the operator allows the oculomotor system to have. This is controlled by the *% Move Target By* parameter in the parameter initialization section.

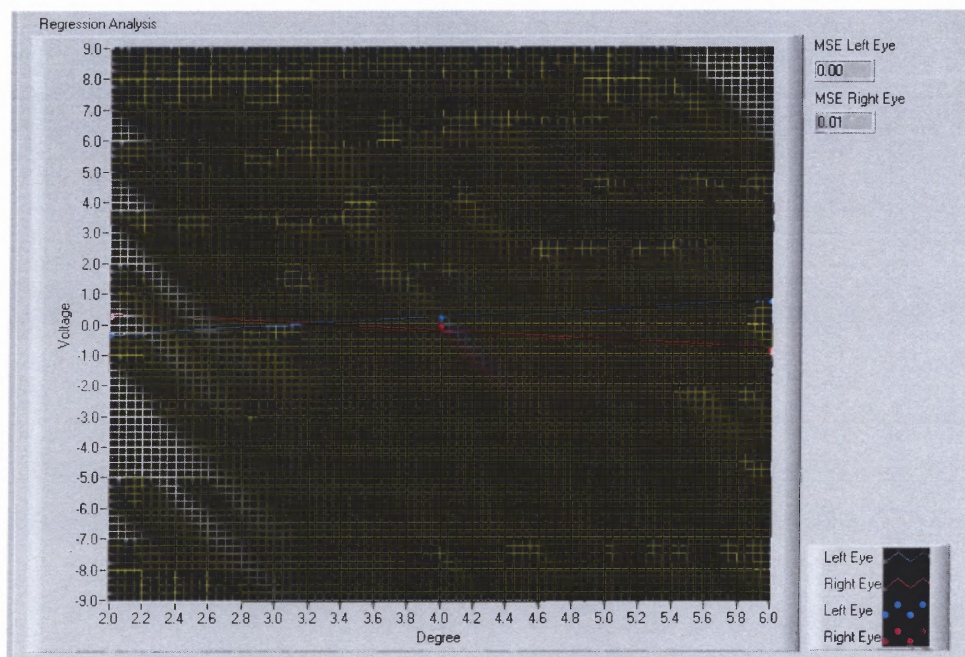


Figure 6.8 Three-point calibration XY-graph.

## 6.4 Data Analysis

### 6.4.1 Three-Point Calibration

The three-point calibration is analyzed for the linearity of the regression line using the correlation coefficient,  $R^2$ . The correlation coefficient describes the strength of the regression line. It is able to determine how strongly the regression line will be able to predict the next variable given the points used during the regression analysis. An  $R^2$  value of one indicates the regression line will be an excellent predictor of the next variable. In this situation, the line falls on top of every point used during the regression analysis. A value lower than one reduces the probability of obtaining a good prediction, thus all  $R^2$  values greater than 0.95 will be considered a good calibration.

### 6.4.2 Peak Velocity

When the subject initiates an eye movement to a step stimulus, there is a burst like movement with a high velocity. In an open-loop experiment, there are multiple bursts since the target is moving to keep a constant error or constant percentage of error present in the experiment. The eyes must continually try to keep up with the target, thus there are multiple peak velocities. The peak velocity will be measured in the data analysis code written in Matlab<sup>®</sup> for each of the peaks indicating a burst of eye movement and compared to previous work completed published in the literature [1, 25]. The peak velocity acquired through the analysis in Matlab<sup>®</sup> will be an approximation since the point at which it occurs will be selected visually and a mouse click will record the value at the specified location.

### 6.4.3 Main Sequence

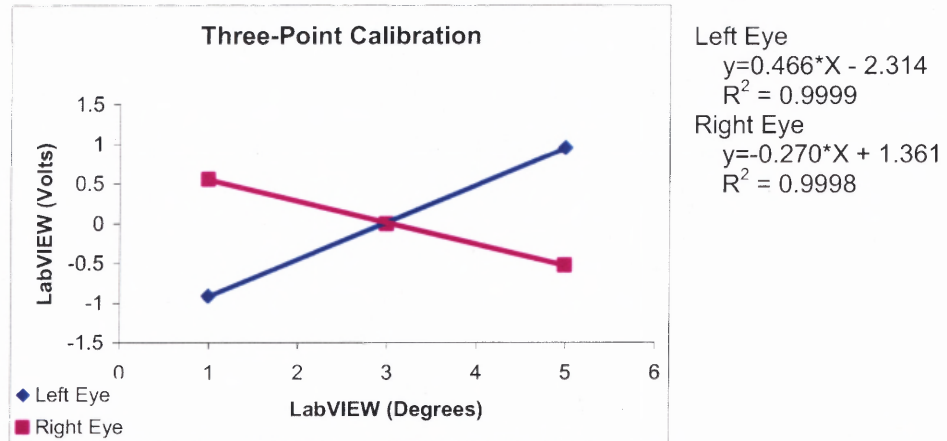
The main sequence is the ratio of peak velocity to position. This method is a quantitative tool commonly used by eye movement researcher to study eye movement dynamics. The analysis technique is able to quantify eye movements to correlate peak velocity with degrees of eye movement. The main sequence in this study is calculated for each of the high velocity movements and compared to data published in the literature [1, 25]. The main sequence calculated by this thesis is an approximation, since the peaks are not fitted to a polynomial curve. The researcher visually selected where the peak velocity and the beginning and end position appeared for each high velocity movement to calculate the main sequence.

## CHAPTER 7

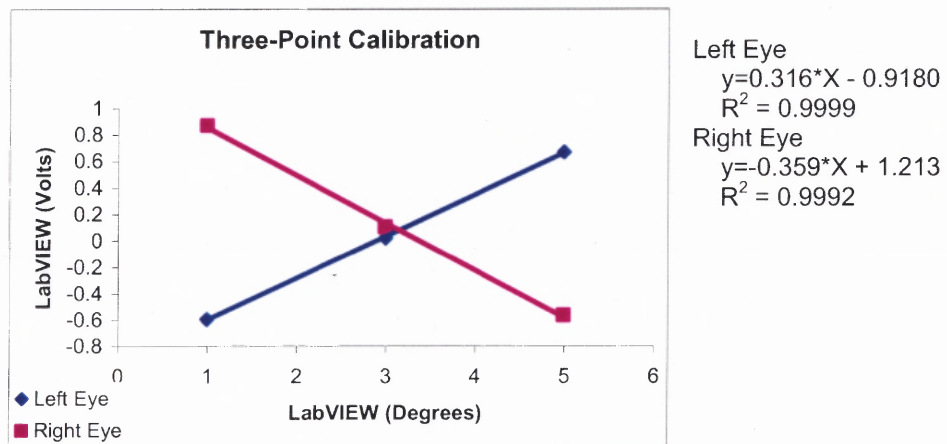
### RESULTS

#### 7.1 Three-Point Calibration

The following is data gathered during the three-point calibration, Figures 7.1, 7.2, 7.3, and 7.4. The three-point calibration acquired voltages values at three known degree positions, 1°, 3°, and 5°, per eye. The voltage values were then plotted against with their respective degree positions for each eye to yield a regression line. In addition to this, the correlation coefficient,  $R^2$ , is shown to describe the strength of the regression line. An equation of the regression line was also calculated during the open-loop code for the real time analysis to determine the current position of the eyes and to present the next position of the target. The following are a sample of three-point calibration graphs with the corresponding correlation coefficient and regression line equation for each eye.

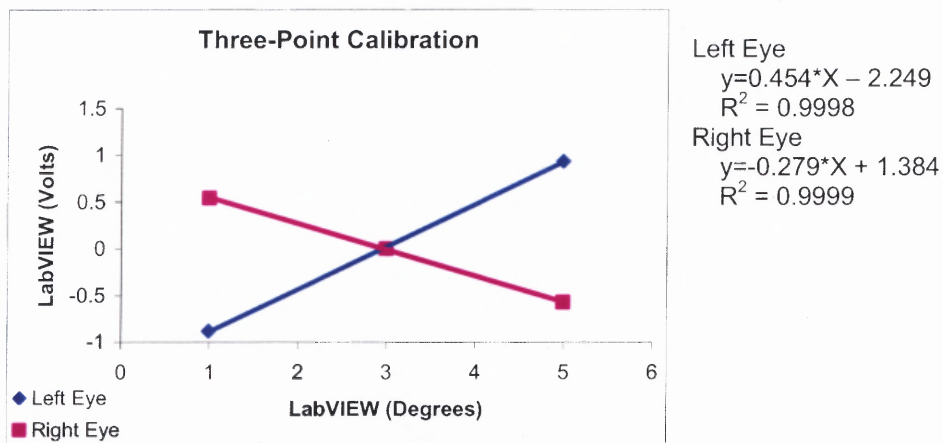


**Figure 7.1** Subject one three-point calibration regression analysis for the correlation coefficient and regression line equation when the stimulus is given 60 percent feedback of the eye movement response.

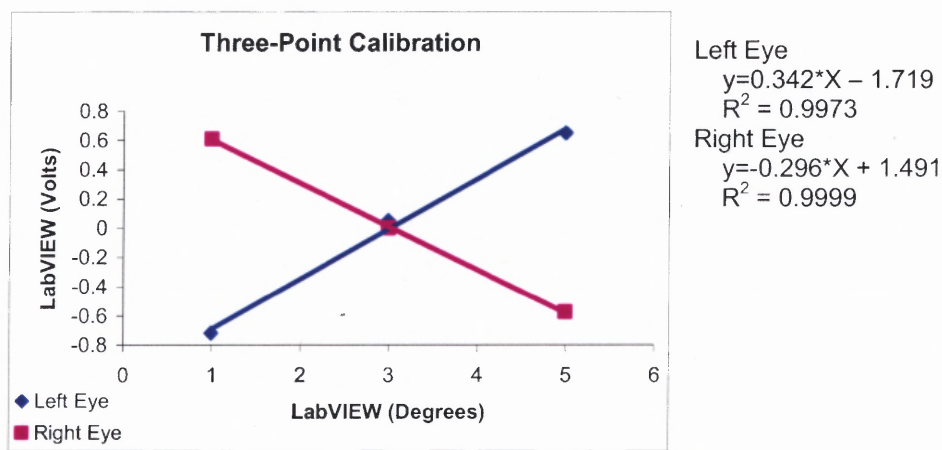


**Figure 7.2** Subject two three-point calibration regression analysis for the correlation coefficient and regression line equation when the stimulus is given 60 percent feedback of the eye movement response.





**Figure 7.3** Subject one three-point calibration regression analysis for the correlation coefficient and regression line equation when the stimulus is given 100 percent feedback of the eye movement response.

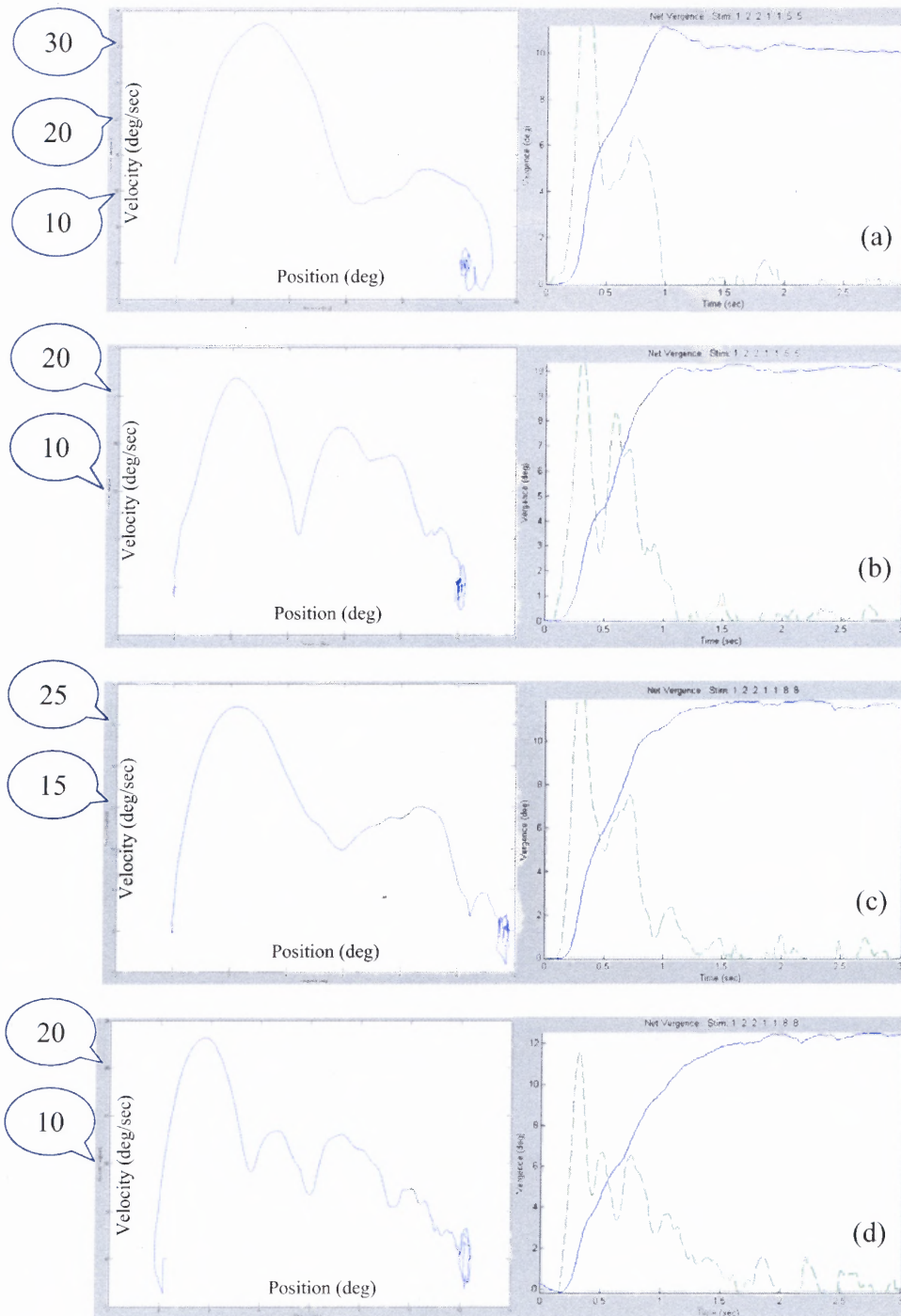


**Figure 7.4** Subject two three-point calibration regression analysis for the correlation coefficient and regression line equation when the stimulus is given 100 percent feedback of the eye movement response.

## 7.2 Stimulus Target Moves 60 Percent of Eye Movement

### 7.2.1 Raw Data

Shown below are the graphs displaying the raw data acquired when the stimulus target moved 60 percent of the eye movement response. This enabled the target to slow down and the subject to eventually fixate on the target position. Figure 7.5 displays a sample of the data from subject one, and Figure 7.6 is a sample of the data acquired from subject two. There are four sets of data shown for each subject (a, b, c, and d); each set containing the phase domain (plot of velocity as a function of position), left, and the position (solid blue) and velocity (dashed green) vs time traces, right. Notice the variability in velocity, number of peaks, and shape of the phase domain. In all the experimental runs, the target made an initial step of four degrees and slowly reduced the error between the position of the eyes and the target.

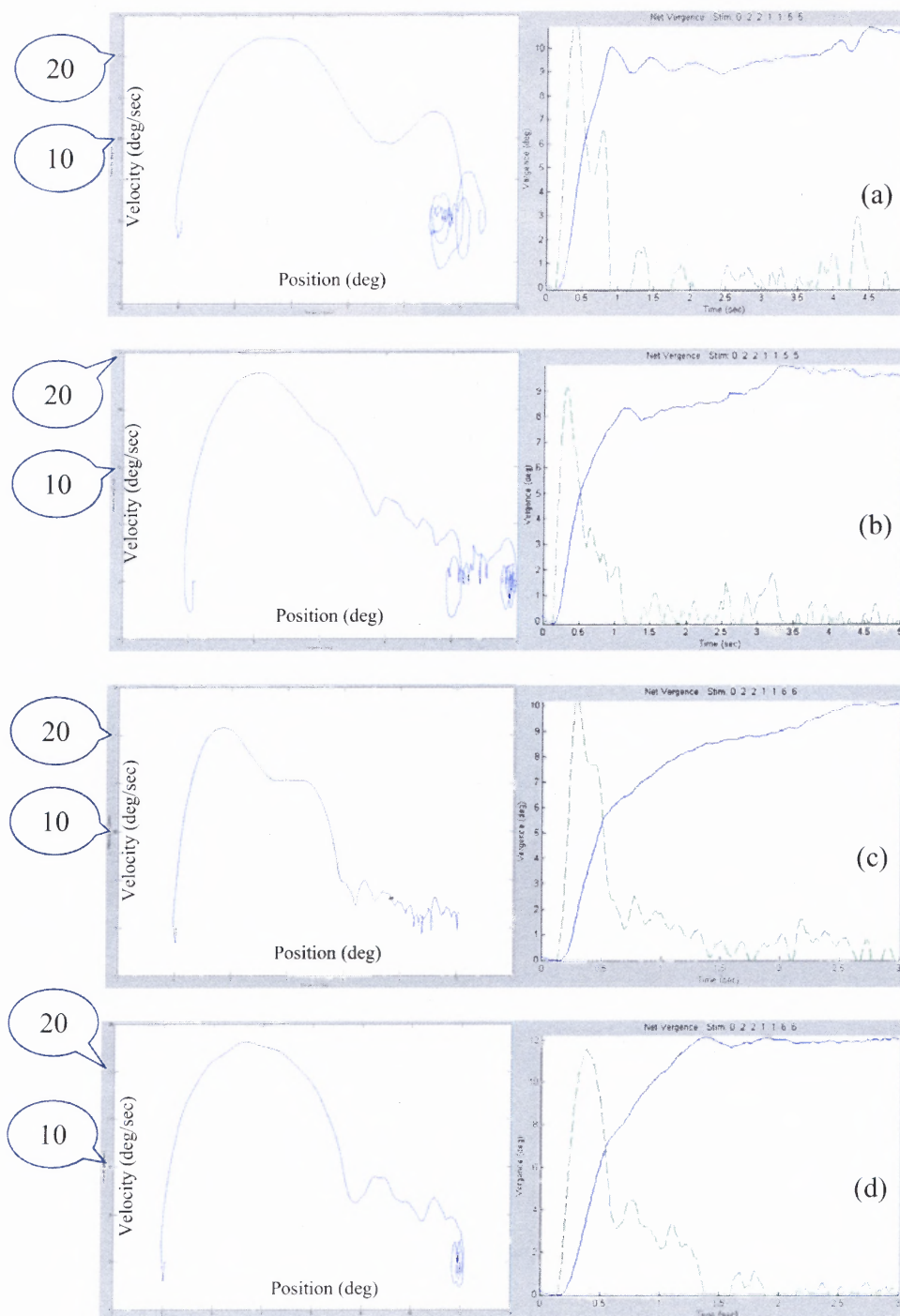


**Figure 7.5** Subject one raw data given a stimulus target with 60 percent feedback from the eye movement response. Left graph shows the phase domain (velocity plotted as a function and right graph displays the position vs time (solid blue) and velocity vs time (dashed green). (a, b) Subject's target saturates at 10 degrees. (c, d) Subject's target saturation set at 16 degrees, but saturation occurs at 12 degrees.

All the experiments run on subject one was for a total time of three seconds in addition to the 0.5 – 2 second initial delay need to avoid prediction. Notice the first peak in all the graphs in Figure 7.5 is much larger than the following peaks, almost twice the size in some runs. In the first two graphs, Figure 7.5(a,b), the target was set to saturate at ten degrees. At this position the target stopped moving, and the subject was able to complete the eye movement and fixate on the target.

The last two graphs, Figure 7.5(c,d), allowed the target to saturate at eighteen degrees, but neither the target nor the eyes reached that position. Rather, both systems stopped moving at twelve degrees. This occurred because at twelve degrees the eyes responses had a smaller error compared to the stimulus; thus the stimulus did not move substantially after the 12 degree target position. In the same last two runs, there are additional smaller pulses beyond the initial two to allow the eyes to reach the final target position. These additional pulses are smaller in magnitude resulting from a smaller degree of eye movement.

The final graph, Figure 7.5(d), shows multiple pulses slowly decreasing in intensity to reach the final target position. The velocity of the initial pulse, 23 degrees per second, was not sufficient to drive a large movement initially. Thus, the first try did not allow the eyes to catch up with the moving target. For this reason another pulse was initiated. This second pulse was also not strong in intensity and additional pulses were initiated with decreased amplitude to allow the eyes to fixate on the target which was decreasing in movement change due to the decrease in error between the response and the stimulus. The initial pulse in Figure 7.5(c) had a velocity of approximately 28 degrees per second. This increased intensity assisted in a greater degree of eye movement.



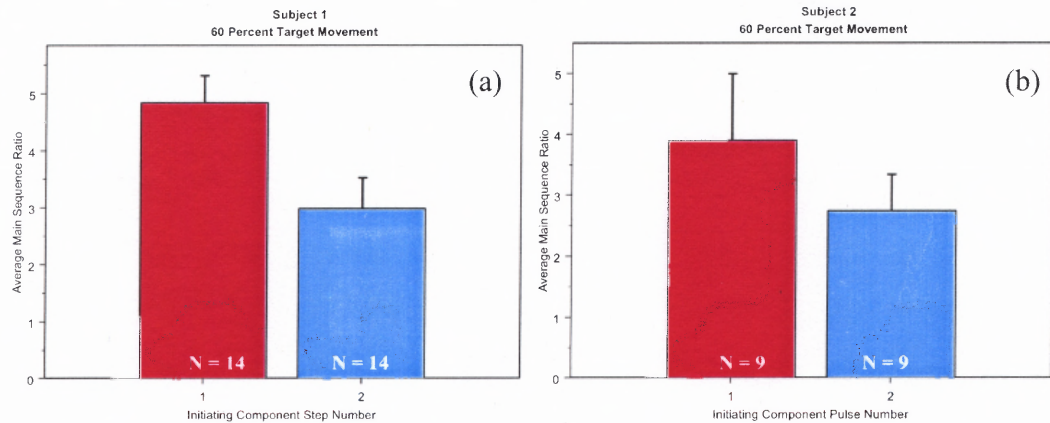
**Figure 7.6** Subject two raw data for when the target is able to move 60 percent of the eye movement. Left graph shows the phase domain (velocity as a function of position) and right graph displays the position (solid blue) and velocity (dashed green) as a function of time. (a, b) Subject's target saturates at 10 degrees. (c, d) Subject's target saturates at 12 degrees.

Subject two displayed similar characteristics as subject one, shown in Figure 7.6. However, subject two's initial dynamics shown in the first peak conducted greater degrees of movement. The second peak was smaller in size and showed a great amount of variability between runs. The first two graphs, Figure 7.6(a,b) were run for a total of five seconds and allowed the target to saturate at ten degrees, whereas the last two graphs, Figure 7.6(c,d), ran for three seconds and the target saturated at 12 degrees.

### **7.2.2 Main Sequence**

The main sequence ratio was calculated for both subject one and two on the first and the second step like movement. All step-like movements after the second were very small in amplitude or non-existent in the current data, thus no main sequence ratios beyond the second step-like response were calculated. Next, the mean and standard deviation were calculated for each subject's main sequence and graphed, Figure 7.7. The graphs of subject one, Figure 7.7(a), and subject two, Figure 7.7(b) both showed a larger main sequence ratio in peak one then in peak two. Subject one reached an average main sequence ratio in peak one of 4.84 and in peak two of 2.98, whereas subject two reached an average ratio of 3.90 in the first peak and 2.74 in the second peak.

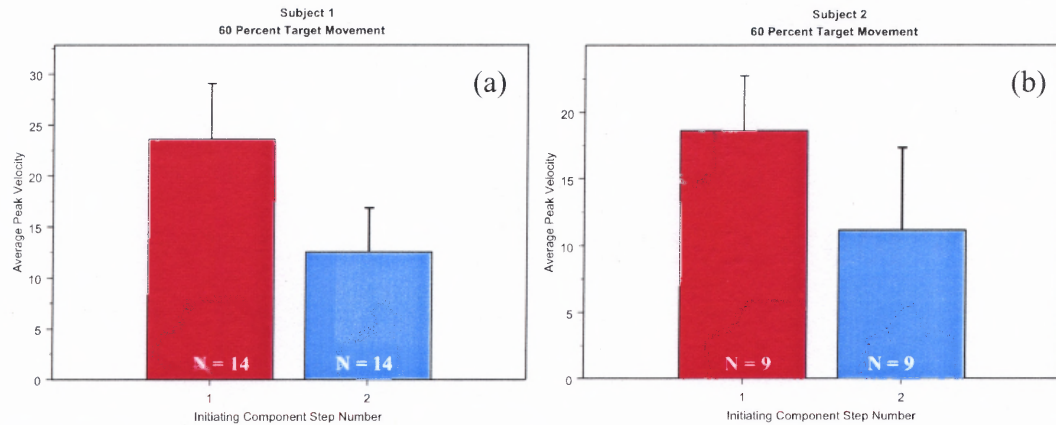
For both subjects, the initial high velocity movement in the main sequence was larger compared to the second high velocity movement. This occurred most likely due to the initial peak consisting primarily of the initiating component. However, since the target was slowing down as the eyes attempted to reduce the disparity, the second high velocity movement occurred at a time when the sustaining component is active. It is speculated that more feedback control is present in the second high velocity movement thus causing a decrease in dynamics compared to the initial high velocity movement.



**Figure 7.7** Main sequence ratio for when the target contains 60 percent feedback from the eye movement response. Only first two peaks are shown. (a) Subject one (b) Subject two.

### 7.2.3 Peak Velocity

The peak velocity for subject one, Figure 7.8(a), and subject two, Figure 7.8(b), is shown for the first two peaks only. Both subjects conducted an initial movement which had a greater velocity on average than the second movement. Subject one experienced an initial peak velocity of 23.61 degrees per second and 12.51 degrees per second for the second high-velocity response. Subject two on the other hand experienced 18.65 degrees per second in the initial peak and 11.16 degrees per second in the second high velocity response.



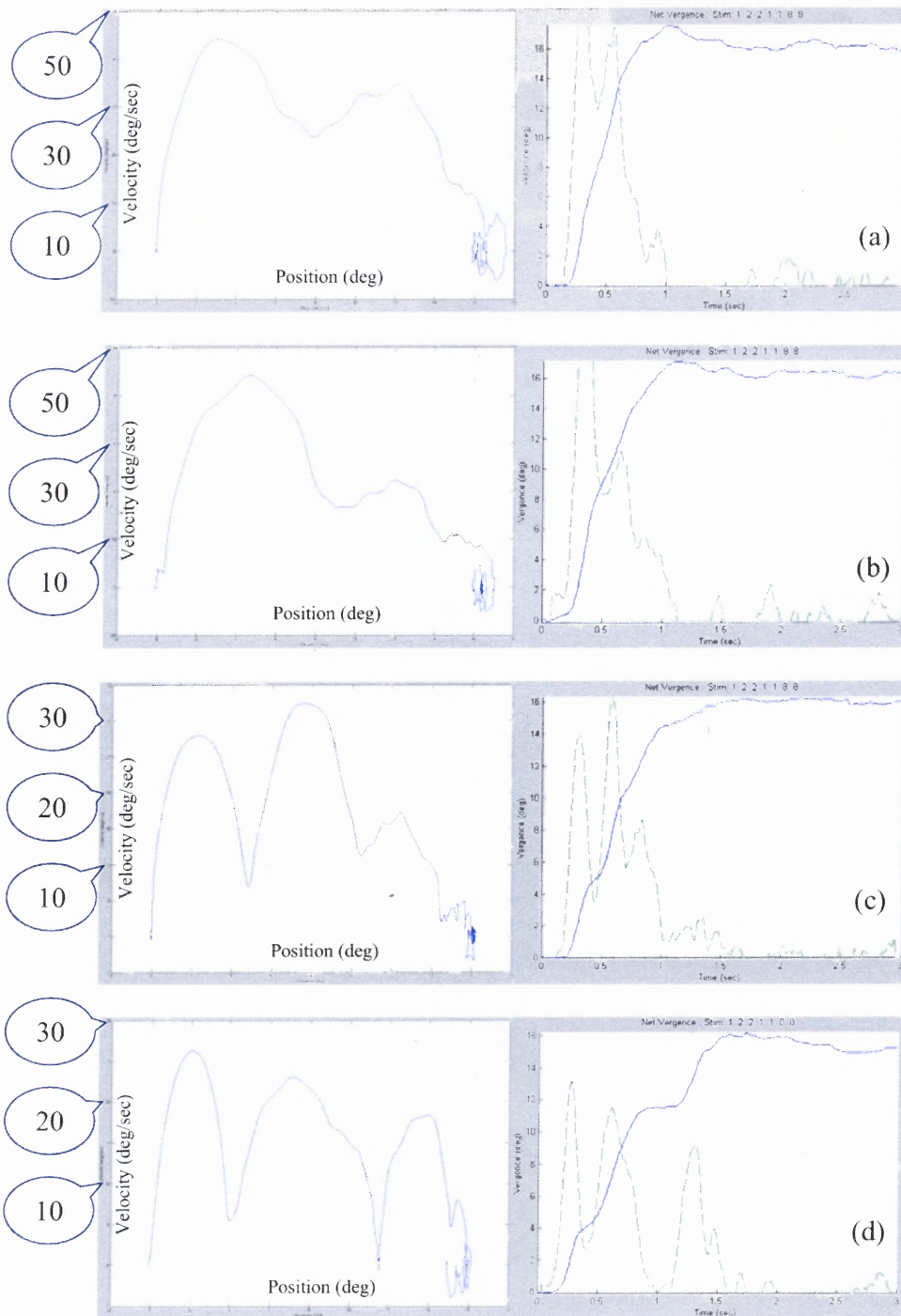
**Figure 7.8** Peak velocity for when the target changes with 60 percent feedback from the eye movement response. Only the first two peak velocity movements are quantified. (a) Subject one (b) Subject two.

### 7.3 Target Receives 100 Percent Feedback of Eye Movement Response

#### 7.3.1 Raw Data

The graphs displaying the dynamics of the oculomotor system when a constant error is maintained between the stimulus target and the position of the eyes is shown in Figure 7.9, subject one, and Figure 7.10, subject two. Both figures display four sets of graphs, with the right side showing the phase domain and the left side the position and velocity as a function of time traces. Also notice the variability in each of the graphs.



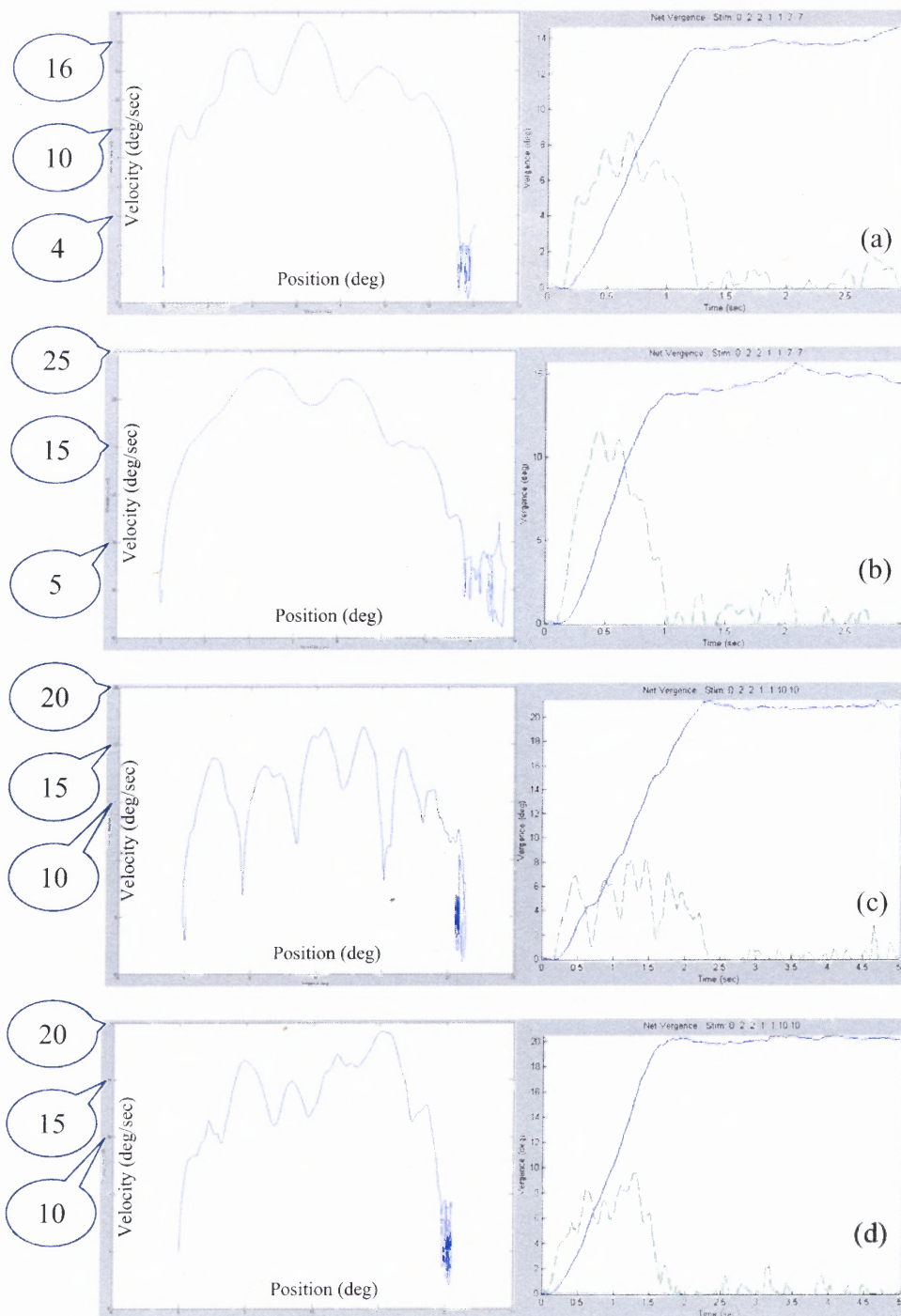


**Figure 7.9** Subject one raw data for when the target is able to move 100 percent of the eye movement. Left graph shows the phase domain (velocity as a function of position) and right graph displays the position (solid blue line) and velocity (dashed green line) as a function of time.

The experiment run on subject one consisted of all the same stimuli in which saturation always took place at 16 degrees. Notice the variability in the peaks, Figure 7.9, in the phase domain. Figure 7.9(a,b) display two noticeable peaks and a third very small one. The first graph, Figure 7.9(a) shows the first two peaks with similar amplitude sizes, where as the second graph, Figure 7.9(b) almost resembles an open loop experiment in which the stimulus target received 60 percent from the eye movement response. This is not the case, yet the second peak is smaller than the first.

In Figure 7.9(c,d), the dynamics of the eyes displayed three peaks. The first graph, Figure 7.9(c) showed the first two peaks with similar amplitudes, but the third one was less than half the size. However, in the last figure, Figure 7.9(d), this was not the case, in which all the peaks were of similar intensity.

All of the graphs show the eyes reached their final position of 16 degrees at which point the target had stopped moving. The method of getting to that final position was variable in the number of peaks and the amplitude of each peak.



**Figure 7.10** Subject two raw data for when the target receives 100 percent feedback from the eye movement. Left graph shows the phase domain and right graph displays the position (solid blue) and velocity (dashed green) as a function of position. (a, b) Subject's target saturates at 14 degrees. (c, d) Subject's target saturates at 20 degrees.

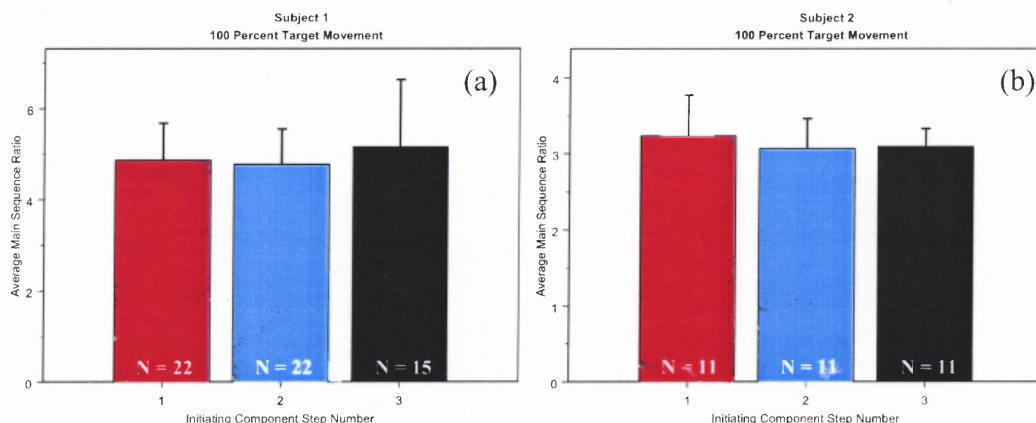
Subject two also had similar characteristics in the dynamics of the eye movement responses and showed a great deal of variability, Figure 7.10. In the first two graphs, Figure 7.10(a,b) the target saturated at 14 degrees. Both graphs look similar in the peak velocity of the peaks, but the number of times an eye movement exhibited a step-like response varies. In the last two graphs, Figure 7.10(c,d), the target saturated at 20 degrees, allowing the subject to conduct a much larger eye movement. The eyes conduct similar movements as in the first two graphs, but additional bursts of eye movements were required to get to the final position of the target. Lastly, the peaks are of similar amplitude, yet there is variability in the size and number.

### **7.3.2 Main Sequence**

The main sequence was calculated for both subjects, Figure 7.11. The first three velocity amplitude peaks were considered in the analysis only, because the subject had reached the target by that time or was very close and the fourth peak was not present in the majority of the runs. The main sequence for subject one, Figure 7.11(a), in the first peak was 4.86, in the second peak was 4.77, and in the third was 5.16. Subject two, Figure 7.11(b), had a main sequence of 3.24 in the first peak, 3.07 in the second peak, and 3.10 in the final peak.

For both subjects, the average main sequence for each high velocity movement was approximately the same. The reason behind this is because when the target initially conducted the four degree step, the eyes initiated a four degree step movement with the corresponding velocity. The target at the same time moved further to maintain the constant four degree error between the position of the eyes and the target. At this point, the ocular system recognized the target had moved and initiated another jump with the

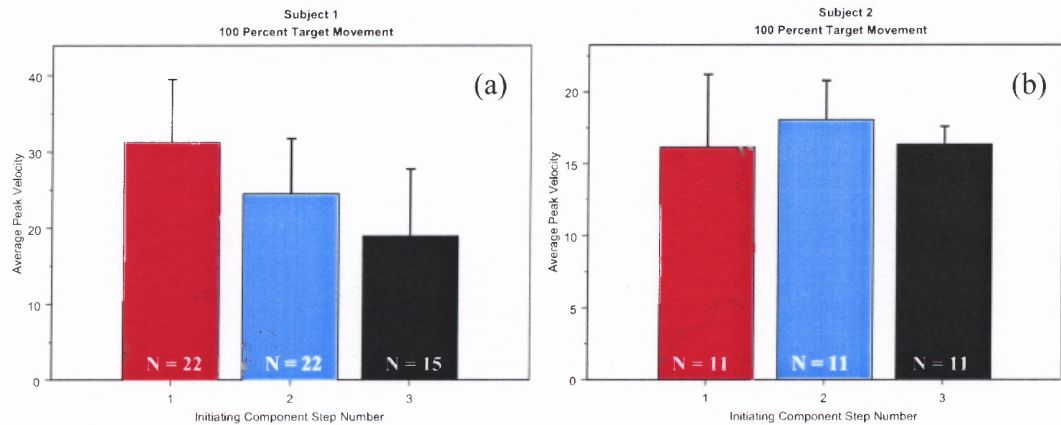
corresponding velocity to get to the target position. This continued until the target reached its saturation point, at which time the eyes were able fuse the stimulus and the sustaining component initiated to reduce the disparity.



**Figure 7.11** Main sequence ratio for when the target is able to move 100 percent of the eye movement. Only first three peaks are shown. (a) Subject one (b) Subject two.

### 7.3.3 Peak Velocity

Although the average main sequence ratio remained consistent in all three peaks in showing how much the subject moved for a given the initial velocity, the average peak velocity did not remain consistent in its magnitude for subject one, Figure 7.12(a). Subject one experienced larger peak velocities in the initial peak of 31.31 degrees per second, and slower peak velocities in the following peaks of 24.48 degrees per second and 18.96 degrees per second. For subject two this was not true, and the average peak velocity was 16.15 for the first peak, 18.04 for the second peak, and 16.31 for the third peak, Figure 7.12(b).



**Figure 7.12** Peak velocity for when the target receives 100 percent feedback of the eye movement. Only first three peaks are shown. (a) Subject one (b) Subject two.

## 7.4 Timing Analysis

### 7.4.1 Aliasing Technique

In an ideal situation, the aliasing technique is an excellent estimator on measuring the timing of a real time system. However, when applying this technique on the open-loop code, the target was not seen to oscillate between two known positions. Rather the monitors displayed between four and six lines when the function generator provided a frequency of 10 Hz. These lines showed slight movement in position on the monitors, yet were the most stable out of all the frequencies tested. The increase in frequency caused the lines on the monitors to move at an increasing rate and had a large amount of variability in the number of lines. This occurred for both open-loop methodologies two and three, discussed in sections 4.3. With these values and assuming all four to six readings were taken within half a period of the sine wave, each scan was analyzed and the target displayed within 8.3 to 12.5 milliseconds. These numbers were acquired by taking the inverse of the frequency, 10 Hz, when there were 6 or 4 lines on the monitors and multiplying it by 1/2. This number was then multiplied by 1/6 or 1/4 for when there

were 6 lines or 4 lines on the monitors respectively to attain the time it took for a single target to be displayed to the screen.

One factor which could have influenced this outcome is in the timing of each iteration of the for-loop in the open-loop code. If the amount of time elapsed by each iteration is not the same, then the program would sample the sine wave at different locations. This in turn would account for the variability in the number of targets presented on the monitors and movement of the targets.

#### **7.4.2 Software Timer**

The software timer was run on methodologies two and three to obtain the average time it took to parse through each iteration of the for-loop. A timing analysis was also completed on each segment of the for-loop which consisted of data acquisition, data analysis, and next target presentation.

Methodology two completed each iteration of the for-loop in an average time of 4.98 msec. The data acquisition portion completed its segment in an average of 2.87 msec. The data analysis completed in 0.01 msec. Lastly, the target presentation on the monitors to the subject completed in 2.06 msec.

Methodology three attained almost identical results as methodology two. It completed each iteration of the for loop in an average time of 4.98 msec. The data acquisition completed in 1.55 msec and the data analysis segment completed in .01 msec. The last sequence, target presentation on the monitors completed in 3.39 msec.

Since the open-loop code is sampling the eye movements at 200 Hz, it should complete the data acquisition, analyses, and presentation of the next target of each scan in five milliseconds. The rate at which each scan should take place is acquired by taking the

inverse of the frequency. The software timer used to calculate the average completion time of each iteration of the for-loop resulted in a value of 4.98 milliseconds for both methodologies two and three. This indicates the open-loop code is running at the correct rate with a real time data analysis.



## CHAPTER 8

### DISCUSSION

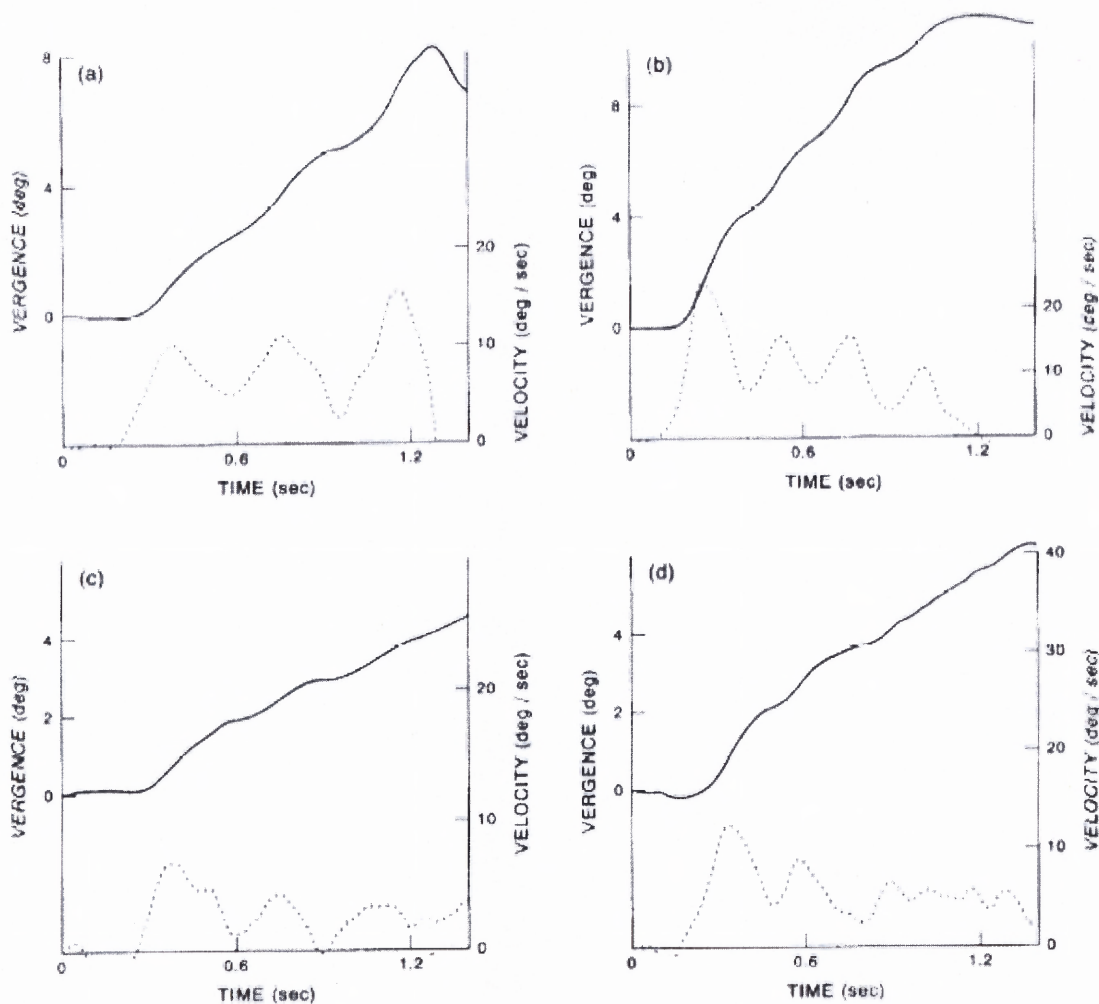
The three-point calibration is essential in assisting the open-loop program to determine the current position of the eyes. The calibration must be very accurate and precise in the regression analysis or the open-loop may not present the position of the target to the subject properly. The data collected on the three-point calibration demonstrates the accuracy in which it is acquired and analyzed. The  $R^2$  value for the runs presented how close the points were to the linear regression line. In all cases the value was greater than 0.99 thus concluding the regression analysis was linear.

The eye movements responses from subjects one and two appeared as a series of step-like responses superimposed on a ramp stimulus. The eye movement responses contained a series of burst movement until the final target position was reached. At this time, the velocity bursts seized and the eyes remained fixated on the target through the feedback portion of the vergence system otherwise known as the sustaining component. When the target stimulus received 60 percent feedback from the eye movement allowing the error to eventually decrease where the eyes could fixate on the target, the movement of the eyes to the target resembled a ramp type movement which was slowly leveling off. On the other hand, when a constant error was kept between the position of the eyes and the target, the ramp appeared to have a constant slope, which leveled off at the saturation point.

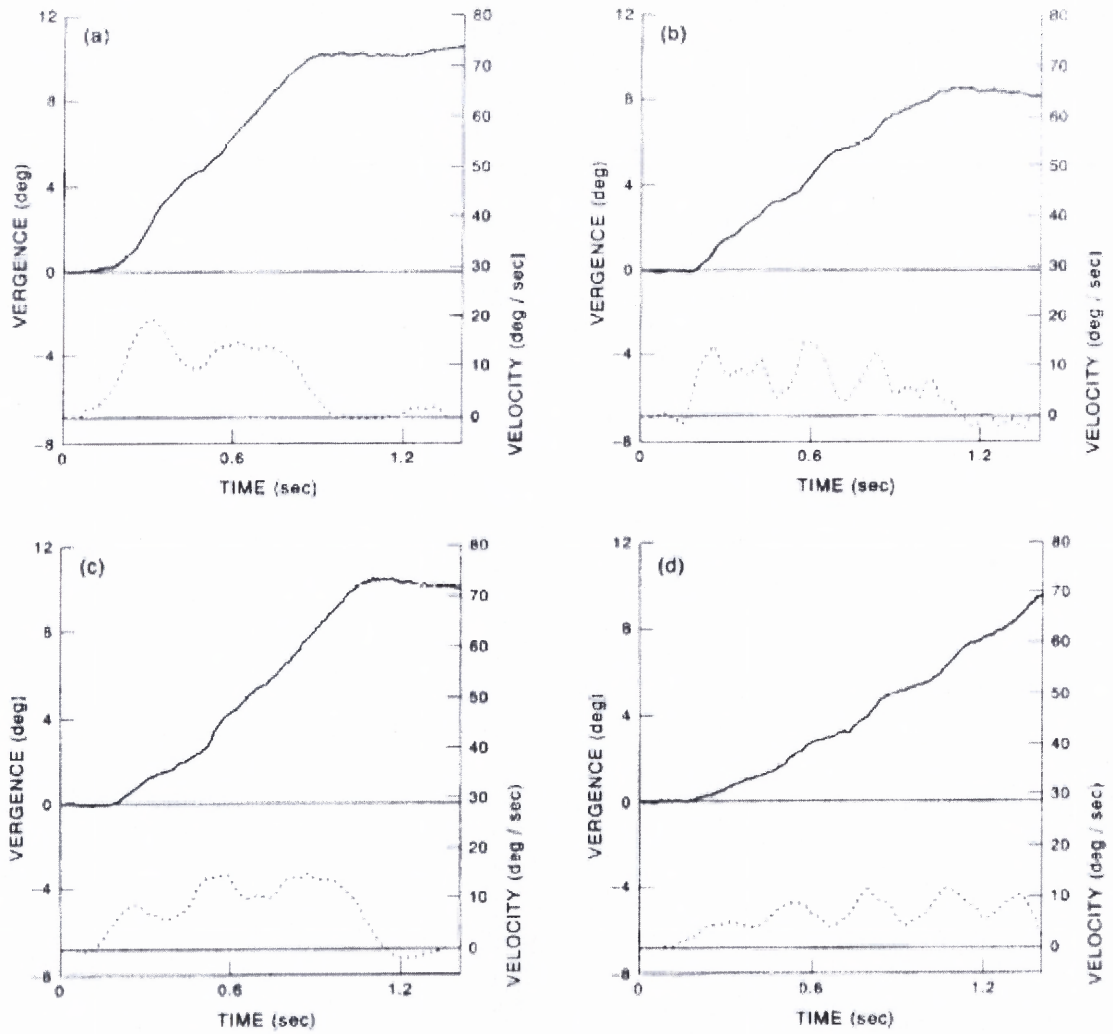
One of the first open-loop experiments performed on the vergence system found in the literature was conducted by Semmlow and colleagues [25]. Their experiment examined two open-loop stimuli, a two degree step and a four degree step. In

presentation of both stimuli, a constant error was kept between the subject and the target, consequently moving the target 100 percent the movement of the eyes. Figure 8.1 displays their raw data for both a four degree (top two graphs) and a two degree (bottom two graphs) initial step response to an open-loop stimulus from four subjects. Figure 8.2 shows the raw data from a single subject. The data shown in Figure 8.2 is the same subject used to validate the current open loop program written here in LabVIEW. In both figures, notice the eyes tracked the target in a ramp like fashion with multiple steps to initiate the eye movement. The number of peaks and the max velocity of each peak varied from run to run, showing much variability similar to what is presented in this thesis.

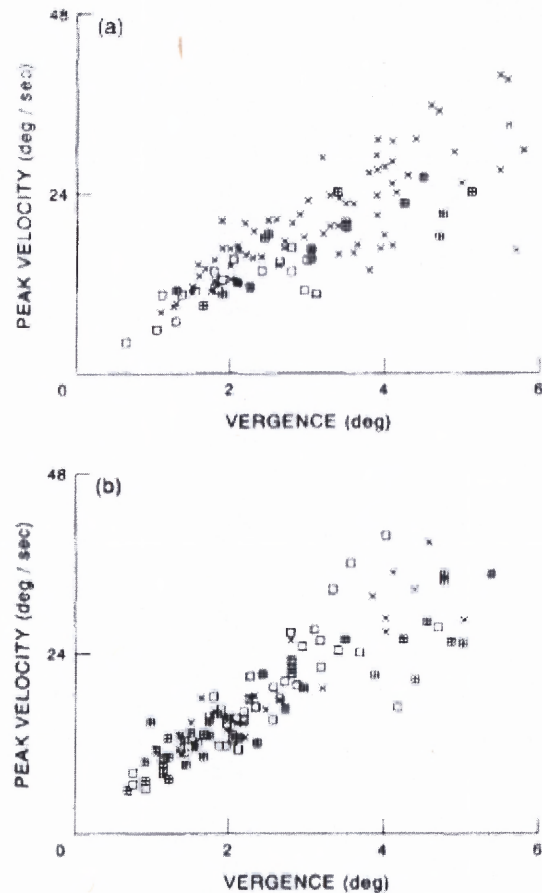
A main sequence analysis was included into the research conducted by Semmlow and colleagues [25]. His analysis plotted the main sequence for all responses in an open-loop and closed loop experiment, Figure 8.3. Subject one for this thesis and his experiment are shown in Figure 8.3(b) in which the average main sequence for the first peak and second peak are approximately six. The average main sequence for the same subject found by this thesis under similar condition was approximately five for all peaks.



**Figure 8.1** Data acquired by Semmlow and colleagues from an open-loop experiment where the stimulus received 100 percent feedback from the eye movement response. Data from four subjects (a,b) Received four degree initial steps. (c,d) Received two degree initial steps [25].



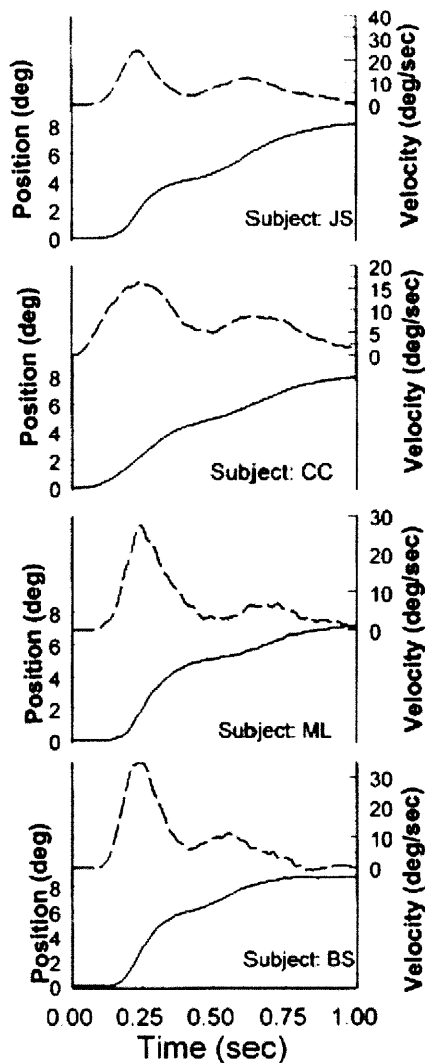
**Figure 8.2** Data acquired Semmlow and colleagues from an open-loop experiment where the stimulus received 100 percent feedback from the eye movement responses and were initially presented a four degree step. The subject who performed this movements is subject one in this thesis [25].



**Figure 8.3** Main sequence of the two subjects run by Semmlow and colleagues under open-loop conditions. + indicates closed loop responses, open square indicates first peak in the open-loop response, and plus inside square indicates second peak in the open-loop response. (a) subject one (b) subject two. Note subject two participated in this thesis research to validate this code and for this research was subject 1 [25].

Alvarez and colleagues [1] ran a similar experiment which allowed the target to receive 60 percent feedback from the eye movement. Their results looked similar to the ones produced by *Open-Loop Program 1.0* when applying the identical stimulus, Figure 8.4. Four subjects were given a four degree step and allowed the subjects to track the target. As the eyes moved, the targets slowed down the error between the responses and the stimulus decreased until the eyes were fixated on the target. The responses resemble ramps, which are slowing as the saturation point is reached. In addition, the maximum

velocity of the first peak appears almost double the maximum velocity of the second peak which is comparable to the results of this thesis.



**Figure 8.4** Data acquired by Alvarez and colleagues from an open-loop experiment where the stimulus received 60 percent feedback from the eye movement [1].

## CHAPTER 9

### CONCLUSION

The primary goal of this thesis was the creation of code, which would be able to collect eye movement data to study the fast, preprogrammed initiating component in an open-loop manner. A variety of methods had been employed in the past to accomplish this task, but the method of presenting the target to the subject, acquiring the eye movement data, and storing it for future study was developed using LabVIEW v6i for this thesis. The implementation of the code in LabVIEW provided the researcher with extended options for future studies and maintained a consistency in the software used by the Vision and Neural Engineering Laboratory.

An essential part of this thesis was the three-point calibration. The data acquired during this stage enabled the open-loop code to conduct a real-time data analysis to determine the next position of where the target appears to the subject. The operator is given significant control over this section to allow for a variety of experiments to take place.

Several methodologies were chosen to be implemented for open-loop presentation. The optimal method was methodology three which used only the current position of the eyes and a mathematical formula to determine the next position of the target. Although a timing analysis showed both methodology two and three to be identical, the usage of memory space and processor time was less with methodology three. In addition to this, since all computer equipment, compilers, processors, and advanced programs are very mathematically oriented, the mathematical model was the best to go with. The single formula enabled both convergence and divergence

experiments to take place for open-loop experiment and gave greater control to the operator.

The open-loop data acquired by this thesis was comparable to the data acquired by other researchers in the past when run under similar conditions. The raw data and main sequence ratios presented by this thesis were comparable to previous data published in the literature [1, 25]. This comparison facilitates the validation of this.

A question arose initially on whether LabVIEW would be able to present the subject with a stimulus, conduct a real time data analysis to move the stimulus, and record the data in a timely manner. After writing the code and comparing the results of this thesis with previous work, LabVIEW has successfully proven to process the data in real time.



## CHAPTER 10

### FUTURE RESEARCH

The open-loop code can be utilized for numerous studies in the future. One of the primary reasons for the creation of *Open-Loop Program 1.0* is its ability to adaptive vergence responses. For survival of a species, adaptation is essential and continually on going. To learn how the brain adapts and sends signals to the eyes is important to understanding how the body works and functions in unity.

In the past, adaptation of the oculomotor system has been largely studied by giving the subject large steps, small steps, and step ramps as the adaptive stimuli. These adaptation protocols were presented to the subject after a baseline study was completed to obtain the control dynamics. The baseline consisted of a single step. This single step was also presented during adaptation and compared to the baseline to measure if adaptation was taking place. In other words five conditioning adaptive stimuli were given with the test stimulus to adapt the oculomotor system. The last phase is usually recovery to measure how long or at what rate the dynamics of the oculomotor system de-adapt and if there is any fatigue present in the system. During this final stage, only the single step stimulus is presented.

Presenting a subject with a stimulus, which will conduct an open-loop experiment is one of the most adaptive methods in temporarily altering a person's visual dynamics. Employing a similar technique of baseline, adaptation, and recovery with open-loop, adaptation can take place at a faster pace and may show a greater change in dynamics. This effect is temporary because the person does not visualize the world as an open-loop system, and thus the dynamics are changed back to the original form.

Another area in which the open-loop data could have implications is in independent component analysis. Independent component analysis (ICA) is a method of taking a combined response from more than one source and splitting it up into its independent components. Currently, when a step response is presented to a subject, both the initiating and sustaining components get activated in the oculomotor system. The components are recorded as a combined response, which ICA takes and breaks up into its individual components. During an open-loop experiment, primarily the initiating component is activated when the target receives one hundred percent feedback from the eye movement response. Passing the data acquired during an open-loop experiment into the ICA algorithm should produce multiple pulses, the initiating component.

The software developed by this thesis, *Open-Loop Program 1.0*, could be further advanced by the addition of an adaptation protocol. The adaptation protocol would be able to acquire the data from the baseline, adaptation, and recovery stages in a single execution without giving the subject a break in between. This would give better results on how fast the subject adapts and returns to control dynamics conditions.

## APPENDIX A

### TRIGGER BUTTON SCHEMATIC

This appendix contains the code to allow for the use of the trigger button during experimentation. In the original *Vision Research Program Version 1.0*, the button press was originally controlled by a digital input. As the *Vision Research Program 1.0* advanced and the need for additional bits arose, the trigger button was removed and assigned to the analog channel TRIG1 on the BNC-2090 [9]. The AI Config.vi controlled which channel and port to be read in from, channel 2 and port 0. This information was then relayed to the AI Start.vi which configured the number of scans to five to read in at a rate of 1000 scans per second by the digital trigger. This information was then passed into the while loop to the AI Read.vi. The while loop iterated until a high, indicating a trigger press, was detected by the AI Read.vi. At which time, the while loop ended its iterative process.

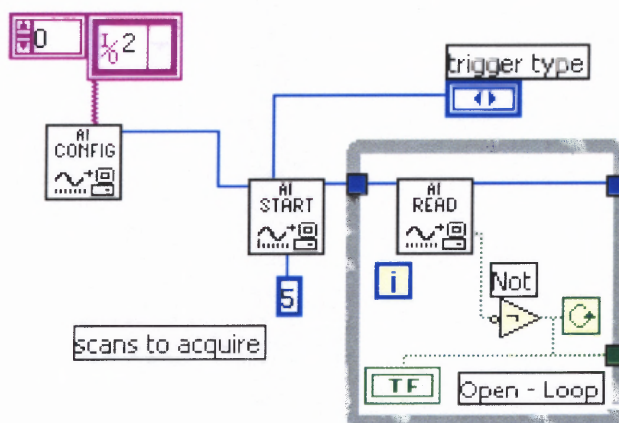


Figure A.1 Trigger button schematic in LabVIEW [9].

## APPENDIX B

### MONITOR STIMULUS PRESENTATION

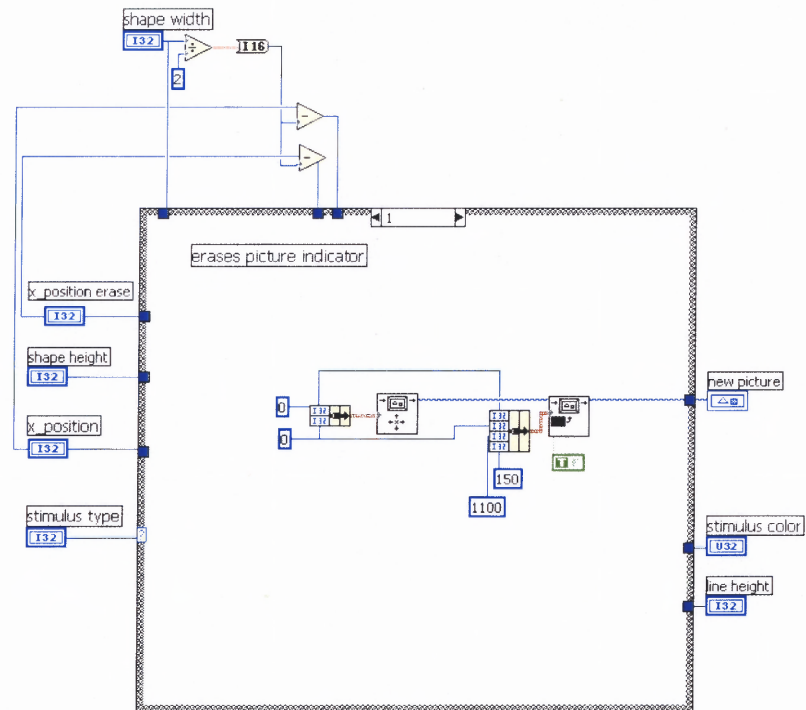
The following code was written to present the subject with a stimulus on the monitor. All of the code was written by Florence B. Chua in LabVIEW and incorporated into the *Vision Research Program 3.0* [2, 7, 8]. The *Open-Loop Program 1.0* is utilizing this code to present the subject with a target to track and collect vergence eye movements from. The advantage of using monitors is the height, width, color, and shape of the target can be controlled.

#### B.1 Front Panel



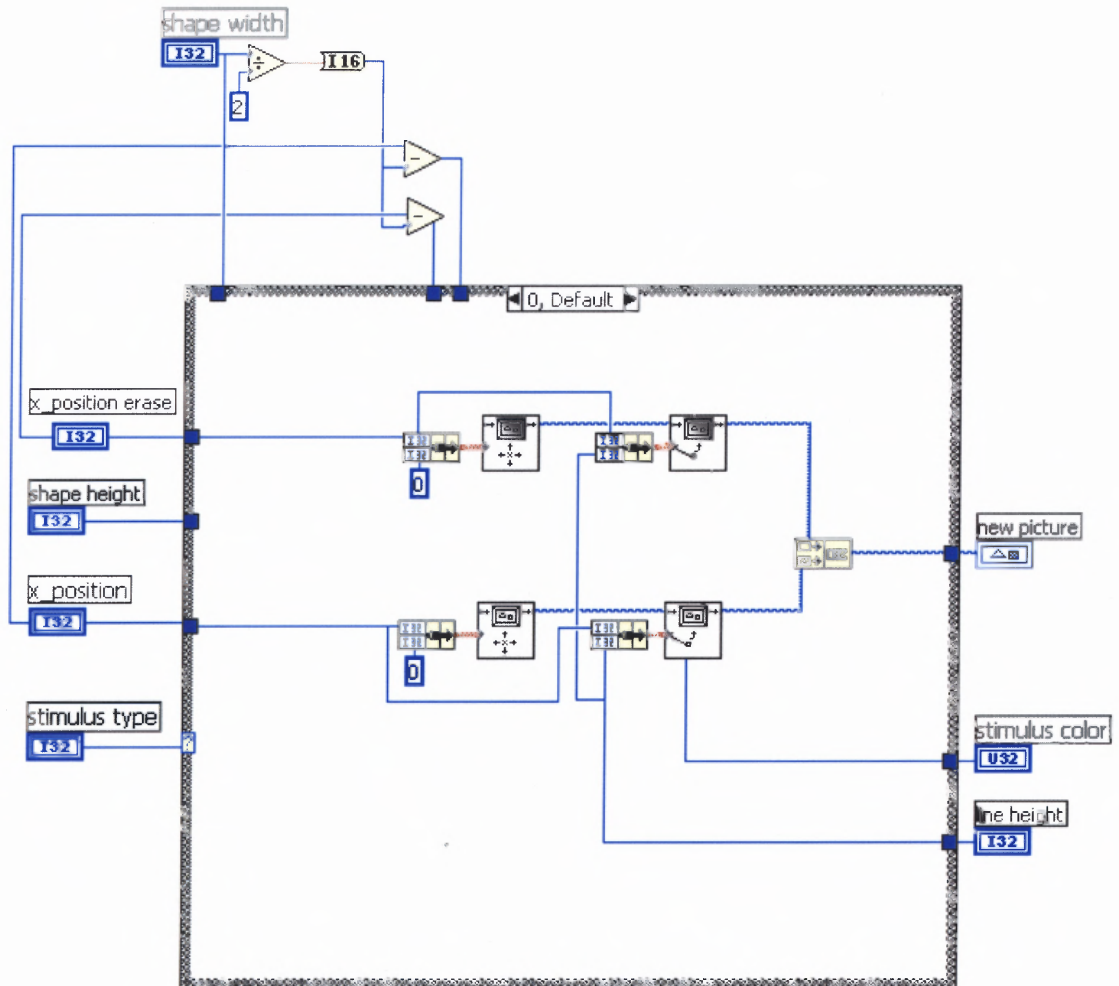
**Figure B.1** Front Panel displays a single green line to the subject.

## B.2 Clear Screen



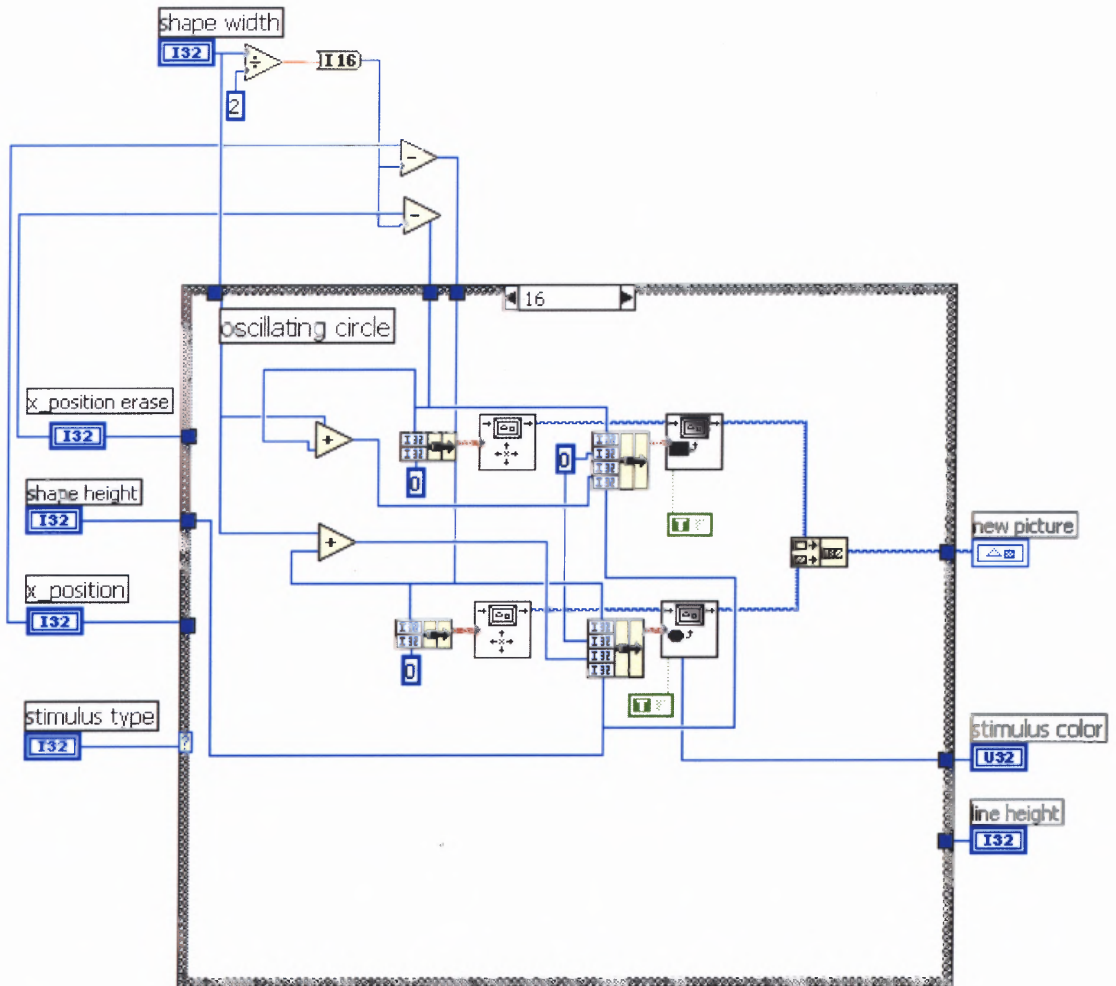
**Figure B.2** This case statement clears the screen when called upon [2, 7, 8].

## B.3 Display Line

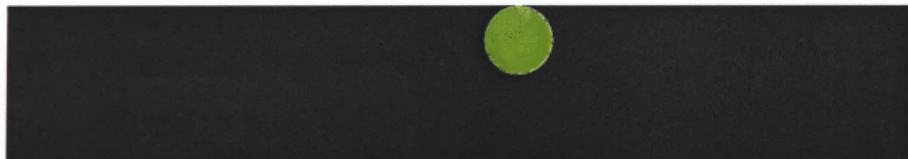


**Figure B.3** This case statement displays a line stimulus at a specified position, with the given height, width, and color. This statement can also remove a line at the specified position [2, 7, 8].

### B.4 Oscillating Circle



**Figure B.4** This case statement presents the subject with a circular figure at a specified position, with the given height, width, and color [2, 7, 8].



**Figure B.5** This is an example of the code written in Figure B.4.

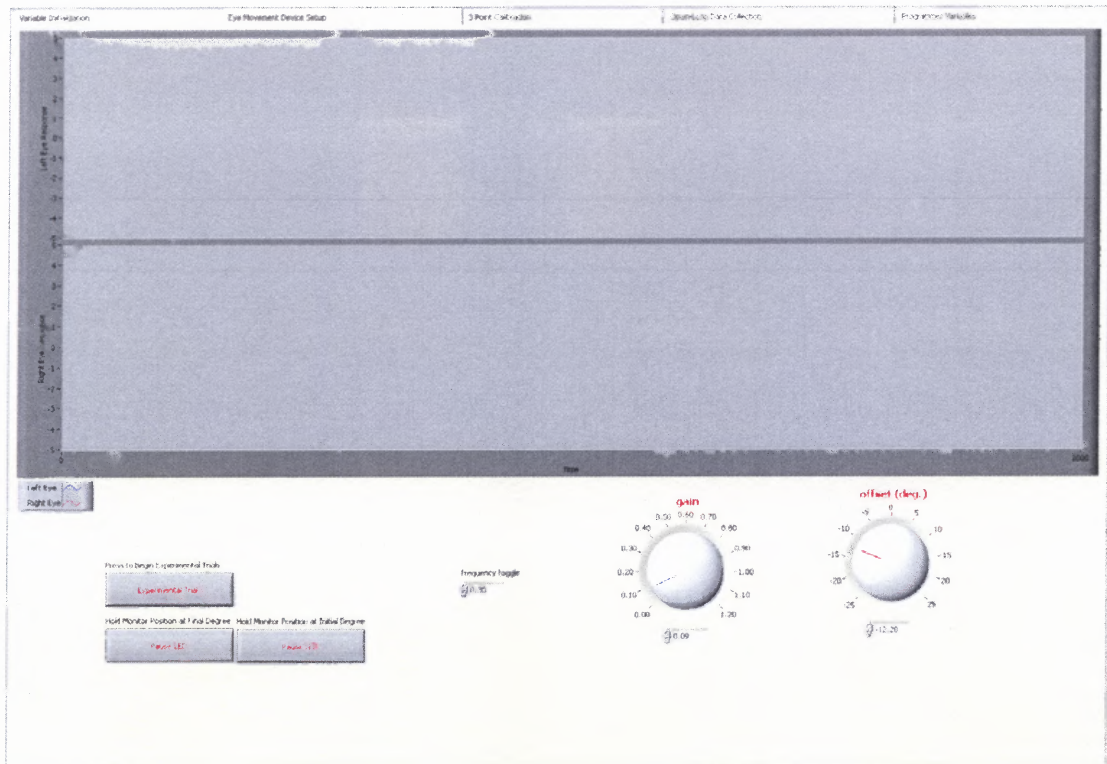
## **APPENDIX C**

### **LIMBUS TRACKING DEVICE ADJUSTMENT AND DEGREE TO PIXEL CORRELATION**

The following code was completed to allow for proper placement, orientation, and signal acquisition of the limbus tracking device for coherent eye movement data. The front panel provided the user with a method of adjusting the bias and offset to orient the location and width of the target on the monitors with real world targets at known positions. The targets were able to stop in their respective positions if asked for by the user. The eye movements were seen in a large graph from both eyes independently. The top portion of the graph displayed the left eye signal, and the bottom the right eye signal.



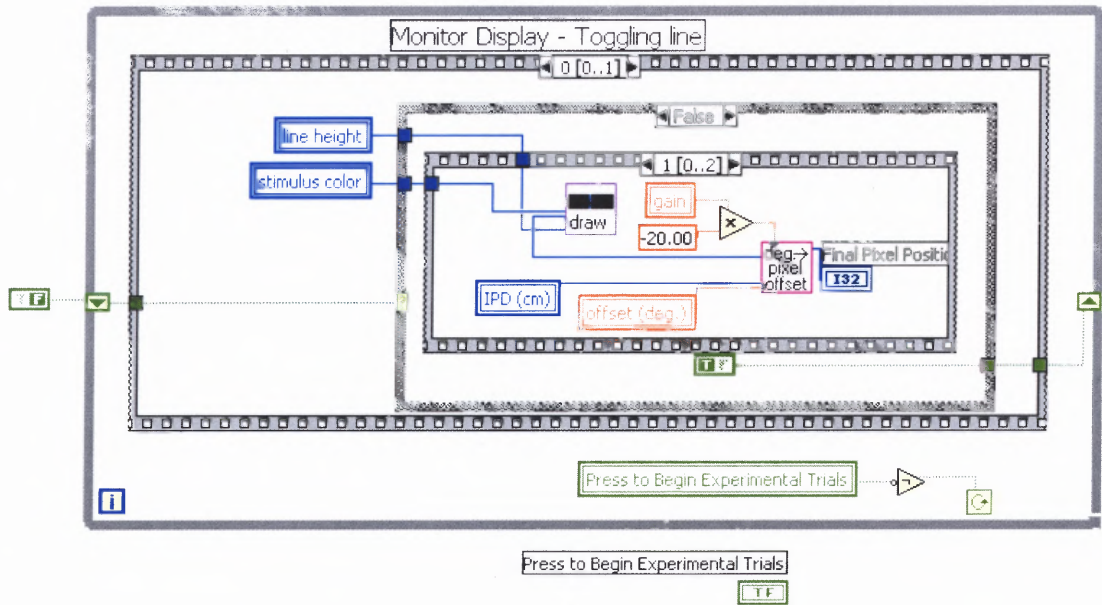
## C.1 Front Panel



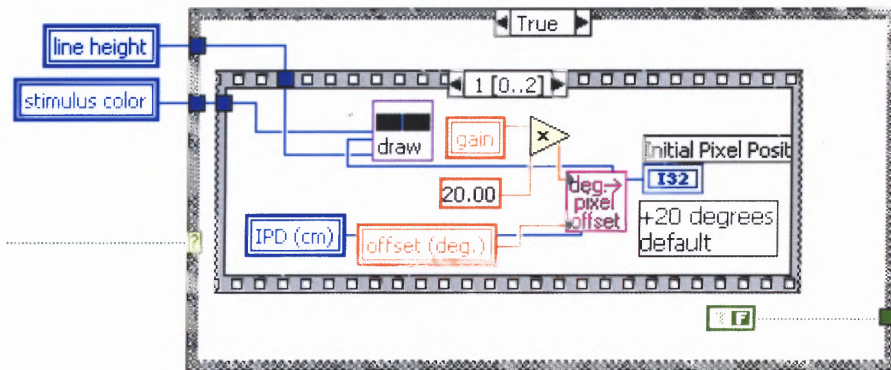
**Figure C.1** Front panel for the eye movement tracker setup and the degree to pixel correspondence.

## C.2 Code For Target Movement On Monitors

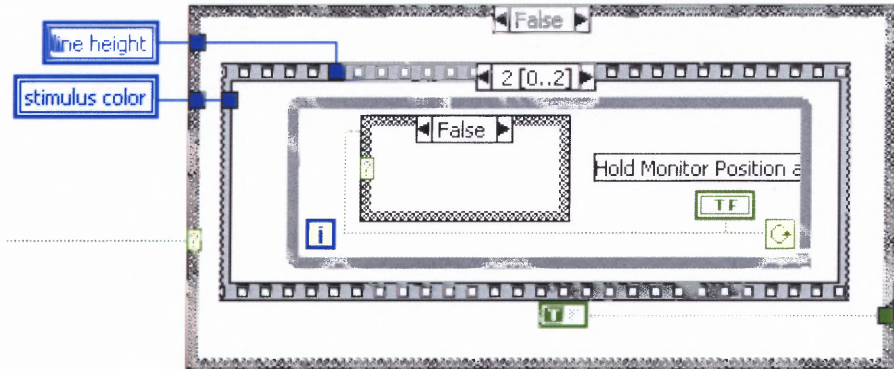
The following code, written by Florence B. Chua (Chua, 2003), is responsible for displaying the initial target, the final target, and pausing each of the targets in its position when displaying on the monitor. The two targets are shown one at a time. The offset and bias are also adjustable to allow the monitor targets to match the real world targets, which are at known positions. The pixel values determined during this stage with the matching degree positions are then passed throughout the program.



**Figure C.2** Entire structure of code and inside case statement (denoted as true/false) displays the final target. In the case of a vergence movement, false statement, it is  $10^\circ$  [2, 7, 8].

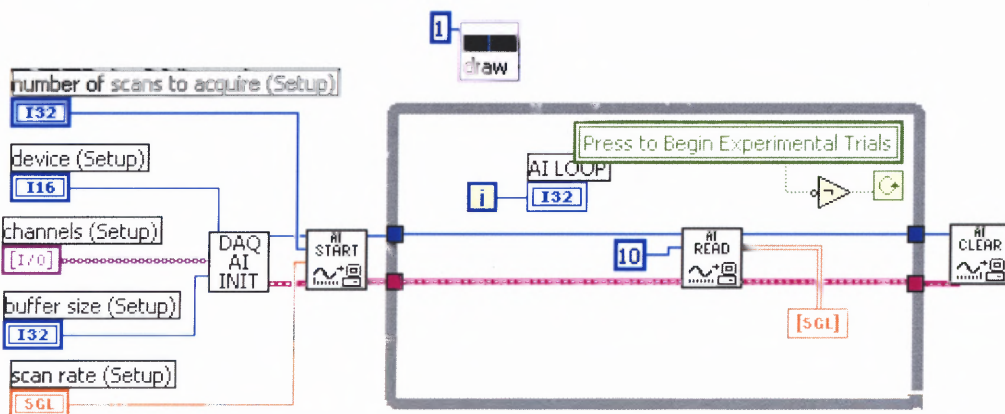


**Figure C.3** This is the other case statement, which displays the initial target. In the case of a vergence movement, it is  $4^\circ$  [2, 7, 8].



**Figure C.4** The following code is part of both true case and false case. This code enables the target to be stopped in its location, depending on whether the true case is called upon or the false [2, 7, 8].

This code in Figure C.5 is responsible for data acquisition. It reads in the eye movements and displays each eye to the graph independently. DAQ AI INIT initializes the card for reading in data. AI START.vi begins the acquisition process and data starts to come in at the specified sampling rate. Within the while loop is the AI READ.vi, which reads in the data from the buffer and displays it on the graph. Once the user hits the stop button to move onto the next stage of the experiment, the DAQ card is cleared using AI CLEAR.vi.

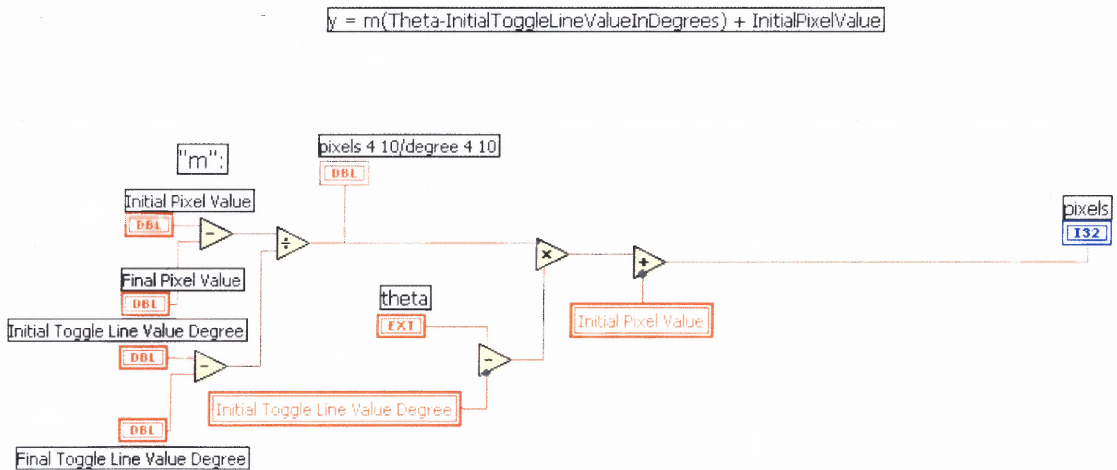


**Figure C.5** Data acquisition from each eye during limbus tracking device setup.

## APPENDIX D

### DEGREE TO PIXEL CONVERSION

The Following code written by Florence B. Chua [2, 7, 8] allowed for the known degree positions and the pixel values from the initial eye movement tracker setup portion of the code to be entered. The code took this information and was able to calculate the pixel position for the monitor of the degree location asked for by the operator. This code only works for vergence eye movements.



**Figure D.1** Front Panel displays a single green line to the subject [2, 7, 8].

## REFERENCES

1. Alvarez, Tara L., Semmlow, John L., Yuan, Weihong, Munoz, Paula. Disparity Vergence Double Responses Processed By Internal Error. *Vision Research*. 40:341 – 347, 2000.
2. Alvarez, T. L., Chua, F. B., Daftari, A. P., DeMarco, R. M., Herrera, L., Bergen, M. T., *Vision Research Program 3.0*, Sep. 2003.
3. Alvarez, T. L., Semmlow, J.L., Yuan, W. Closely Spaced, Fast Dynamic Movement is Disparity Vergence. *Journal of Neurophysiology*. 79:37 – 44, 1998.
4. Bahill, AT, Clark, MR, Stark, L. The Main Sequence, A Tool For Studying Human Eye Movements. *Mathematical Biosciences*. 24:191 – 204, 1975.
5. BK Precision Electronic Test Instruments (2003), *Model#4040A 20 MHz Sweep Function Generator*, retrieved November 17, 2003 from [http://www.bkprecision.com/www/np\\_specs.asp?m=4040A](http://www.bkprecision.com/www/np_specs.asp?m=4040A)
6. BK Precision® Corporation, *Instruction Manual for BK Precision® Models 2120B and 2125A 30 MHz Dual Trace Oscilloscopes*, Placentia, California, 2000.
7. Chua, Florence B. LabVIEW-Controlled Stimulus Presentation On A Monitor for Smooth Pursuit Eye Movement Experiments. *Thesis Submitted To New Jersey Institute of Technology*. May 2003.
8. Chua, Florence B. *Monitor Stimulus Display and Vergence Degree To Pixel*, April 2003.
9. Chua, Florence B. *Trigger Button*, Dec. 2002.
10. Cambridge Research Systems, Inc. (1999), *Skalar IRIS IR Light Eye Tracker*, Retrieved November 15, 2003 from: <http://www.crsLtd.com/catalog/skalar/iris/>.
11. eMedicine Inc. (2003) *Extraocular Muscles, Actions*, Retrieved November 13, 2003 from: <http://www.emedicine.com/oph/topic14.htm>.
12. Horng, Jai-Long, Semmlow, John L., Hung, George K., Ciuffreda, Kenneth J. Initial Component Control In Disparity Vergence: A Model Based Study. *IEEE Transactions On Biomedical Engineering*. 45:249 – 257, 1998.
13. Hung, George K. Dynamic Model Of The Vergence Eye Movement System: Simulations Using MATLAB/SIMULINK. *Computer Methods and Programs in Biomedicine*. 55:59 – 68, 1998.

14. Hung, George K., Semmlow, John L., Ciuffreda, Kenneth J. A Dual-Mode Dynamic Model of the Vergence Eye Movement System. *IEEE Transactions On Biomedical Engineering*. 33:1021 – 1028, 1986.
15. Jones, R., Fusional vergence; Sustain and Transient Components. *Am. J. Optom. Physiol. Opt.* 57:640 – 664, 1980.
16. Krishnan, V.V. and Stark, L. A Heuristic Model of the Human Vergence Eye Movement. *IEEE Transactions On Biomedical Engineering. Eng.* 24:44 – 49, 1977.
17. LabVIEW Technician. Personal Communication.
18. Marieb, Elaine N. and Mallatt, Jon. *Human Anatomy, Second Edition*, Benjamin Cummings, an Addison Wesley Longman Imprint, Menlo Park, CA 94025. 417 – 419, 1996.
19. MathWorks Inc. *Signal Processing Toolbox For Use with MATLAB® Users Guide*. The MathWorks, Inc., Natick, MA 01760. 1.19 – 1.21, 6.29 – 6.33, 6.134, 1998.
20. Mays, Lawrence E., Porter, John D. Neural Control of Vergence Eye Movements: Activity of Abducens and Oculomotor Neurons. *Journal of Neurophysiology*. 52:743 – 761, 1984.
21. Mitchell, D.E., Properties of Stimuli Eliciting Vergence Eye Movements and Stereopsis, *Vision Research*. 10:145 – 162, 1970.
22. National Instruments Corporation<sup>1</sup> (2003). *NI PCI-6024E*, Retrieved November 15, 2003 from: <http://sine.ni.com/apps/we/nioc.vp?cid=10968&lang=US>
23. National Instruments Corporation<sup>2</sup> (2003), *BNC-2090*, retrieved November 15, 2003 from <http://sine.ni.com/apps/we/nioc.vp?cid=1177&lang=US>.
24. Rashbass, G. and Westheimer, G. Disjunctive Eye Movements *Journal of Physiology. (London)* 174:245 – 264, 1961.
25. Semmlow, John L., Hung, George K., Horng, Jia-Long, and Ciuffreda, Kenneth J. Disparity Vergence Eye Movements Exhibit Preprogrammed Motor Control. *Vision Research*. 34-10:1335 – 1343, 1994.
26. Semmlow, J.L., Hung, G.K., and Ciuffreda, K.J. Quantitative Assessment of Disparity Vergence Components. *Investigative Ophthalmology & Visual Science* . 27:558 – 564, 1986.

27. Semmlow, J.L., Hung, G. K., Horng, J. -L., and Ciuffreda, K.J. The Initial Control of Disparity Vergence Eye Movements. *Investigative Ophthalmology & Visual Science* . 13:48 – 55, 1993.
28. Semmlow, John L., Yuan, Weihong. Components of Disparity Vergence Eye Movements: Application Of Independent Component Analysis. *IEEE Transactions On Biomedical Engineering*. 49-8:805 – 811, 2002.
29. Semmlow, John L., Yuan, Weihong, Alvarez, Tara. Short-Term Adaptive Control Process In Vergence Eye Movement. *Cahiers De Psychologie Cognitive / Current Psychology Of Cognition*. 21-4/5:343 – 375, 2002.
30. St. Lukes Cataract and Laser Institute (1996-2003), *Eye Anatomy*, Retrieved November 13, 2003 from: <http://www.stlukeseye.com/Anatomy.asp>. Illustrations by Mark Erickson.
31. Vander, Arthur, Sherman, James, Luciano, Dorothy, *Human Physiology The Mechanisms of Body Function, Seventh Edition*, WCB McGraw-Hill, Boston, MA. 248, 1998.
32. ViewSonic Corporation (2002), *ViewSonic E70fb*, retrieved November 17, 2003 from [http://www.viewsonic.com/products/crt\\_e70fb.htm](http://www.viewsonic.com/products/crt_e70fb.htm).
33. Westheimer G., and Mitchel, D.E. The Sensory Stimulus for Disjunctive Eye Movements *Vision Research* 9:749 – 755, 1969.