SPATIAL ORIENTATION IN THE SQUIRREL MONKEY: AN EXPERIMENTAL AND THEORETICAL INVESTIGATION

by

DANIEL M. MERFELD

B.S.M.E., University of Wisconsin-Madison, 1982
M.S.E., Princeton University, 1985

Submitted in partial fulfillment
of the requirements for the
degree of

DOCTOR OF PHILOSOPHY
in
BIOMEDICAL ENGINEERING
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, Massachusetts
January, 1990

Signature of Author ______________________________

Certified by ______________________________
Professor Laurence R. Young
Chairman, Thesis Committee

Certified by ______________________________
Professor Richard M. Held

Certified by ______________________________
Dr. Charles M. Oman

Certified by ______________________________
Professor Conrad Wall III

Certified by ______________________________
Professor Lena Valavani

Accepted by ______________________________
Professor Alan J. Grodzinsky
Chairman, HST Committee on Biomedical Engineering

SPATIAL ORIENTATION IN THE SQUIRREL MONKEY: AN EXPERIMENTAL AND THEORETICAL INVESTIGATION

by

DANIEL MICHAEL MERFELD

Submitted to the
Interdepartmental Program in Biomedical Engineering
on January 12, 1989 in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy in Biomedical Engineering

ABSTRACT

The interaction of angular and linear stimuli produces a complex alignment of spatial orientation and the vestibulo-ocular reflexes (VOR). This phenomenon was studied by measuring three dimensional eye movements in six male squirrel monkeys during centrifugation in the dark. The axis of centrifuge rotation was always aligned with gravity and with the spinal axis of the upright monkeys. The erect monkeys were placed in one of four orientations: 1) facing into the motion; 2) facing away from the motion; 3) facing the center of rotation; 4) facing away from the center of rotation. These different orientations determined the direction of centripetal acceleration with respect to the monkey. Angular velocity trapezoids were utilized as the motion stimuli with a ramp acceleration (and deceleration) of ten degrees per second squared to a constant velocity of 200 degrees per second. This profile provided a final centripetal acceleration of 1 g, which yielded a gravito-inertial force which was tilted 45 degrees with respect to earth vertical and which had a magnitude of 1.4 g.

The orientation of centripetal acceleration dramatically altered the VOR by changing the axis of eye rotation, the peak value of slow phase eye velocity, and the time constant of per-rotary decay. The axis of eye rotation always tended to align with gravito-inertial force, the peak value of slow phase eye velocity was greater when the monkey faced the motion than when it faced away from the motion, and the time constant of decay was smaller when the monkey faced the motion than when it faced away from the motion. These findings were statistically significant ($p < 0.05$) and were very consistent across all monkeys.

As previous research has indicated, further modeling of the data indicates that the VOR may be separated into two reflexes: a linear reflex and a rotational reflex. The linear reflex decays as the axis of eye rotation aligns with gravito-inertial force (GIF). These results, along with other data in the literature, indicate that gravito-inertial force is resolved into two components; one representing an internal estimate of linear acceleration and one representing an internal estimate of gravity.

A "sensory conflict" model of spatial orientation is developed. The model is based upon observer theory, optimal observer theory (Kalman Filtering), and more general nonlinear concepts. The model hypothesizes that the central nervous system of the squirrel monkey "knows" the dynamics of the sensory organs and uses this knowledge to create an internal model of the sensory systems. The internal model receives as input the internal estimate of linear acceleration and angular velocity. The output of the internal model is the expected sensory afference. The expected sensory afference is compared to the actual sensory afference to yield sensory conflict. The sensory conflict information drives the internal estimate of linear acceleration and angular velocity toward the true values. The predictions of the model are compared to results from this thesis and from the literature.

Thesis Supervisor: Dr. Laurence R. Young
Title: Professor of Aeronautics and Astronautics

ACKNOWLEDGEMENTS

The acknowledgements have proven to be the hardest portion of this thesis to write. How can I thank those who have played such important roles? Thomas Paine once wrote,

> *"These are the times that try men's souls. ...in this crisis, (many) shrink from the service of their country (or friend); but he that stands it now, deserves the love and thanks of men and women."*

Thomas Paine, of course, was not writing about his Ph.D. thesis, but his words seem appropriate nonetheless. I thank all of those who have put up with me during these trying times and ask that you accept this mercifully short and simple thanks.

The faculty and staff of the Man-Vehicle Lab have earned my thanks. Professor Young provided advice and guidance whenever it was required. Dr. Oman planted some of the seeds which led to this thesis. Dr. Natapoff helped a great deal with experiment design. Sherry, provided assistance and counseling well beyond the call of duty. Bob R. also supplied a great deal of advice, some of which I mistakenly ignored. Thanks also to Marilyn, Barbara, Pat, Kim and Jim.

The rest of my thesis committee (Professors Held, Valavani, and Wall) deserve special thanks.

Without the Vestibular Research Facility (VRF) and the VRF staff most of this work would not have been possible. Many thanks to Dr. Tomko who was instrumental in making sure that this research was completed. The rest of the VRF staff (Shawn, Pat, Tim O., Tim F., Greg, Jim, Ginny, and Ann) often provided assistance beyond the call of duty.

Dr. Paige developed and performed the surgery. Without this work and his advice the project couldn't have been successful.

The Center for Space Research has provided a pleasant environment for this work and has also provided a large number of necessary services. Thank you!

Friends are probably the most significant and necessary part of anyone's life. Please accept this humble tribute. (Some of my friends have already been mentioned, and ya' only get acknowledged once.) Mark Twain once wrote,

> *"There's plenty of boys that will come hankering and gruvvelling around when you've got an apple, and beg the core off you; but when they've got one, and you beg for the core and remind them how you give them a core one time, they make a mouth at you and say thank you 'most to death, but there ain't-a-going to be no core."*

I guess Samuel Clemens wasn't lucky enough to have friends like mine!

Dava proved to be the best officemate, roommate, and soulmate I ever had. Brad is the best Australian friend I ever had. Dave S. and Ted shared the "most excellent" adventure I ever had. Mark was the most helpful peer I ever had.

At times when I needed refuge from MIT, a number of friends opened their homes to me. Thank you Eric and Laura; Ron and Noemi; Aga and Yusuf; the Clark family; and JoJo.

Thanks are also due to Mike M., Rich, Yasmin, Andrew & Aude, Roberta, Mary, Anthony, Kathy, Jennifer, Amy, Tom, Scott, Mike F., and Bryan.

I also want to recognize my family who have supported me in every possible way.

And finally, I would like to thank Eilis for moral support and so much more during those times when it was most needed.

To all of those, listed and unlisted, who have helped me survive an often turbulent trip. Thank you very much.

I would also like to thank those who have provided financial support during my stay at MIT: various NASA contracts (NASW-3651, NAG2-445, NAS9-16523), the NASA Graduate Student Researchers Program, the GE Forgivable Loan Fund, PSI, and the VRF (via the SJSU Foundation). Thanks are also due to Imagine That who provided a version of "Extend" to the MVL.

Biographical Sketch

Dr. Merfeld obtained a Bachelor of Science in Mechanical Engineering (B.S.M.E.) degree from the University of Wisconsin - Madison in 1982. He went on to Princeton University where he obtained his Master of Science in Engineering (M.S.E.) in 1985. His M.S.E. thesis, entitled *Magnetoplasmadynamic (MPD) Thruster Performance: Propellant Injection and Species Effects*, investigated the basic physics of MPD propulsion.

Dr. Merfeld has published papers in propulsion and space physiology and has an interest in human factors in the arctic.

Dr. Merfeld has a number of incredibly boring stories about trips to China and the Arctic. He also has plans for a number of boring vacations including trips to Alaska, the Orient, and the CCCP.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter One

Chapter Two

Chapter Three

Chapter Four

Appendix B

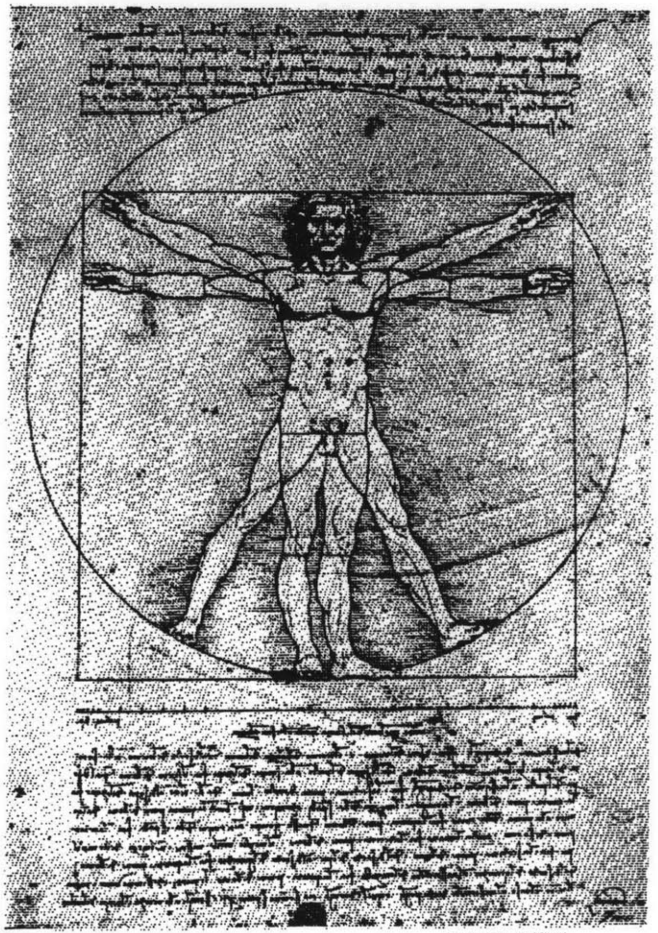
Appendix D

LIST OF TABLES

Chapter Three

I

INTRODUCTION



Leonardo da Vinci

"Our body is a machine for living. It is organized for that, it is its nature."
Leo Tolstoy

"Not that the story need be long, but it will take a long while to make it short."
Henry David Thoreau

CHAPTER ONE

INTRODUCTION

Perception of body orientation and perception of motion are among the most fundamental tasks performed by the central nervous system (CNS). Incorrect determination of orientation or perception of motion can be inconvenient or even lethal. When these important perceptual processes are impeded by ambiguous and contradictory sensory information, they can lead to inappropriate or destructive behavior.

A well known law of physics (equivalence principle) states that no physical device can detect the difference between gravity and linear acceleration. Gravito-inertial force (GIF) is the vector sum of gravitational force and the force due to linear acceleration. All animals on earth live in an environment in which gravity is always present, and in which they sometimes accelerate themselves linearly. Since the ability to assess self-orientation and self-motion is critical to posture and other voluntary tasks, many species have developed a specific sensory modality to detect gravito-inertial force. Since only total GIF can be sensed, these species must develop some consistent strategy to resolve GIF into estimates of gravity (or "down") and linear acceleration (or linear motion) and must produce the appropriate response to each.

From a practical viewpoint, the resolution of gravito-inertial force into its components is a very important neural process. A pilot performing evasive maneuvers will rapidly accelerate. The process through which pilots resolve GIF into acceleration and "down" will affect their sense of orientation, and hence, can be a matter of life or death. In a rotating space station the process by which the CNS resolves gravito-inertial force into its components will be more complicated than on earth. Head motion out of the plane of rotation will result in otolith sensed "coriolis forces" which, at least at first, will be very disorienting. Understanding the process by which gravito-inertial force is interpreted on earth is necessary to understand the adaptation process which will occur on a rotating spacecraft.

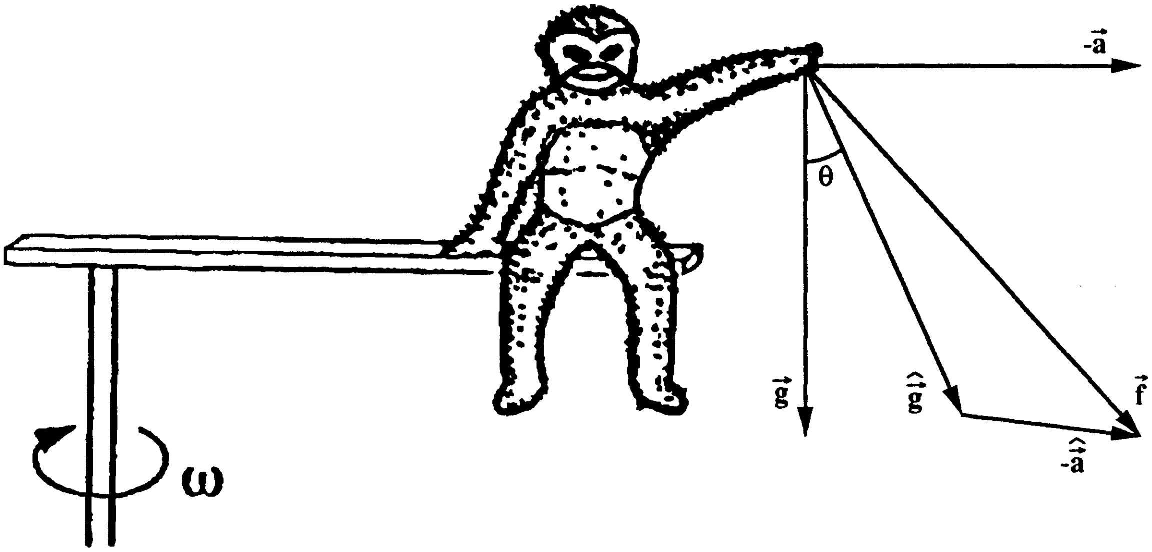
In all mammals, the otolith organs are the sensory organs which transduce changes in gravito-inertial force. These physiological linear accelerometers, like all linear accelerometers, respond both to linear accelerations and to changes of orientation with respect to gravity. This ambiguity is normally resolved by interpreting the signals from the otoliths in light of other sensory information. For example, a moving visual field could assure the subject that he was accelerating linearly. Similarly, a semicircular canal signal indicating rotation could confirm a change in orientation with respect to gravity.

One method used to study the processing methods of the central nervous system is to provide situations in which the conflicts are not easily resolved. In the case of linear accelerations, one experimental method often used is that of centrifugation. In this method, the subjects are rotated about an axis which is displaced by some distance from the subject. The noncentric rotation results in a centrifugal force which, in the subject's reference frame, is vectorially added to gravity to yield the total sensed gravito-inertial force. A task facing the central nervous system of the subject is to determine which vector component of the gravito-inertial force represents gravity or "down" and which component represents linear acceleration or linear motion.

The central hypothesis tested in this thesis is that the CNS resolves the gravito-inertial force into components such that the estimates of gravity and linear acceleration will sum to the total sensed gravito-inertial force vector (i.e. such that $\widehat{\vec{g}} - \hat{\vec{a}} = \vec{f}$). Specifically, I predict that the CNS uses all available sensory information to make an estimate of "down". In some manner, the CNS compares this estimate of "down" with the total sensed gravito-inertial force. The difference between these two quantities, one measured (GIF) and one computed ("down"), is then attributed to linear acceleration. (See figure 1.1.)

This hypothesis is investigated by measuring the vestibulo-ocular reflex (VOR) in six male squirrel monkeys during centrifugation. Reflexes are often measured as a tool to investigate physiological systems. For studies of body orientation and perception of motion, eye movements are often the most conveniently measured response. In all of the

GRAVITO-INERTIAL FORCE (GIF) RESOLUTION HYPOTHESIS



Notation:

$\vec{g}$ is gravitational force.

$\vec{a}$ is centripetal acceleration.

$\vec{f}$ is gravito-inertial force (GIF).

$\hat{\vec{g}}$ is the internal estimate of gravitational force.

$\hat{\vec{a}}$ is the internal estimate of linear acceleration.

Notes:

$\vec{g}$, $\vec{a}$, and $\vec{f}$ are measurable quantities which are determined by the physics of the situation being experienced.

$\hat{\vec{g}}$ and $\hat{\vec{a}}$ are hypothesized to be determined by the CNS to represent internal estimates of "down" and linear acceleration.

Observe that the internal representation of acceleration ($\hat{\vec{a}}$) can have a component which is parallel to the axis of rotation and another perpendicular to this axis.

Figure 1.1

experiments presented herein, the reflexive eye movements known as the vestibulo-ocular reflex (VOR) were recorded as the only physiological measurement.

Previous research has shown that the rotational VOR is sensitive to the presence of gravity (Raphan et al., 1981; Tomko et al., 1986). Other research has demonstrated the presence of linear VOR during centrifugation (Young, 1967). In this thesis I study the relationship between the rotational VOR *and* the linear VOR during centrifugation and use these results to support the GIF resolution hypothesis shown in Figure 1.1. I believe that these experiments are the first to simultaneously examine rotational responses and linear responses in an effort to track the resolution process.

1.1 Thesis Organization

The major portion of this thesis is included in Chapter 3 and Chapter 4. These chapters present experimental results and a model which helps explain the results. Chapter 2 presents the methods by which the data was analyzed and may be skipped without any consequence for those interested only in the physiological conclusions or implications.

Chapter 2, entitled "Analysis of Eye Movements", has three major components. In Section 2.1, I discuss the implications of eyeball kinematics and develop a method to calculate the exact eye velocity and the exact eye orientation from a set of non-orthogonal coils. Section 2.2 presents the method and computer algorithms which were used to remove fast phase eye movements from the data. Section 2.3 briefly discusses the least squares algorithm used to estimate the response time constants.

Chapter 3, entitled "VOR During Centrifugation", presents the experimental findings which support the GIF resolution hypothesis shown in Figure 1.1. In section 3.1 I discuss of a number of background studies. In Section 3.2 and Section 3.3, I present the experimental methods and the experimental results,respectively. In Section 3.4, two control studies are discussed. These studies along with the centrifugation study support the conclusions which are presented in Section 3.5. Finally, the findings and conclusions are summarized in Section 3.6.

Chapter 4, entitled "Sensory Conflict Model", also has three major components. Section 4.1 presents a historical perspective of sensory models which lead to the sensory conflict model. In Section 4.2, I develop the general theory underlying the model. In Section 4.3, the performance of the model is compared to experimental data from the literature.

Finally, in Chapter 5, evidence supporting the basic hypothesis is reviewed and suggestions are made for future investigations. The model is also summarized and the most important suggestions are reviewed.

1.2 Coordinate System

Choice of a single coordinate system helps keep the analysis and presentation simple. The most widely used coordinate system for vestibular research is that defined by Hixson et al. (1966). This orthogonal right-handed coordinate system is shown in figure 1.2. The positive x-direction is defined to point in the forward direction, the positive y-direction is defined to point toward the subject's (or specimen's) left, and the positive z-direction is defined to point out the top of the head. (See Figure 1.2.)

Oftentimes, eye movements are referenced to a left-handed coordinate system. In order to avoid the confusion associated with using two different systems, I have deliberately chosen to reference the eye movements to the right-handed coordinate system presented in Figure 1.2. This choice of reference frame may prove slightly confusing when I cite other literature. In those cases, I have attempted to parenthetically refer to the notation used in the literature. I feel that using a single coordinate system eliminates more confusion than it creates. If any questions about the reference frame arise, simply refer to Figure 1.2.

Eye velocity will be discussed throughout this thesis. It is important to remember that all eye movements are actually the rotation of a sphere-like eyeball. The rate of rotation of the eye is most accurately represented by a vector aligned with the axis of rotation. The terms horizontal eye velocity, vertical eye velocity, and torsional eye velocity will have their

COORDINATE SYSTEM

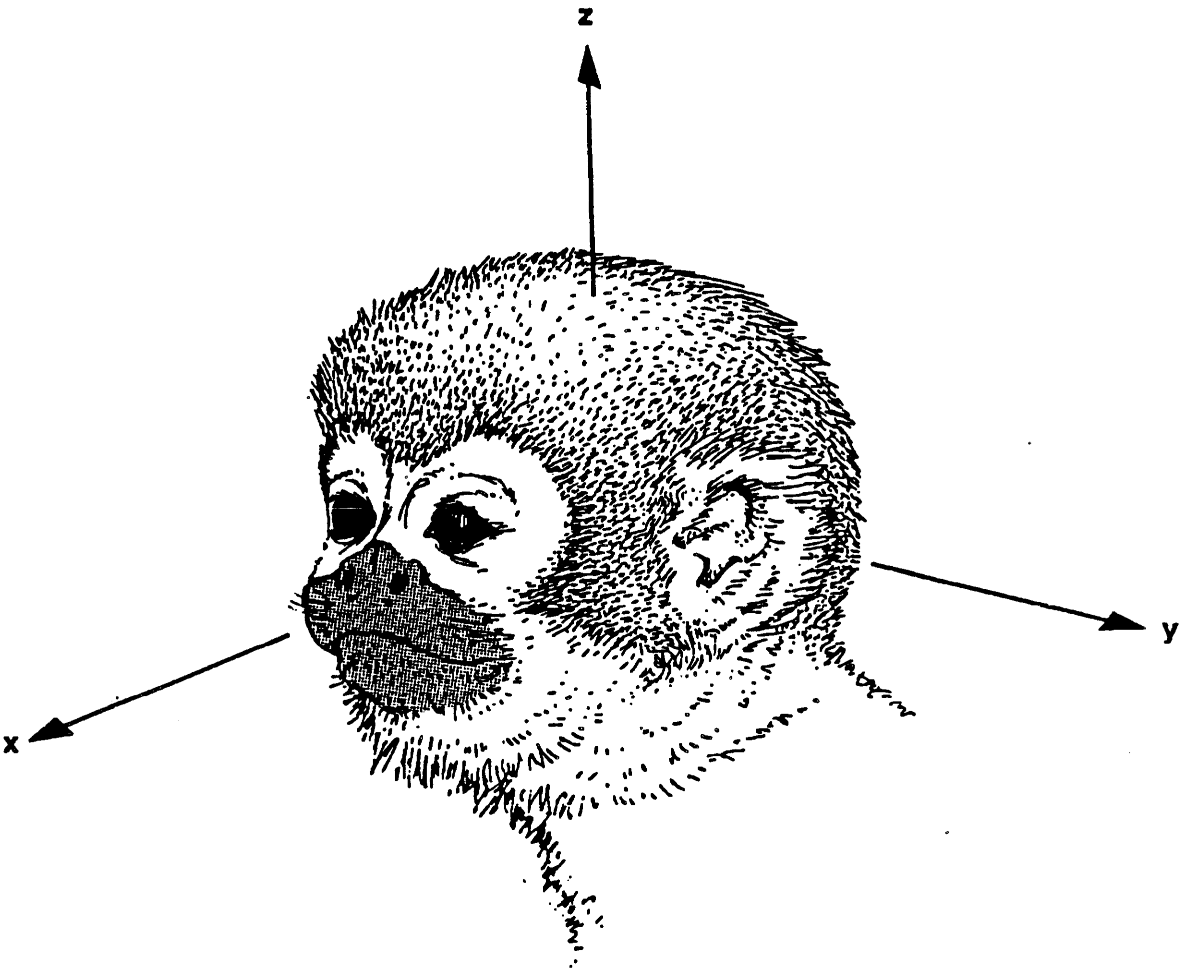
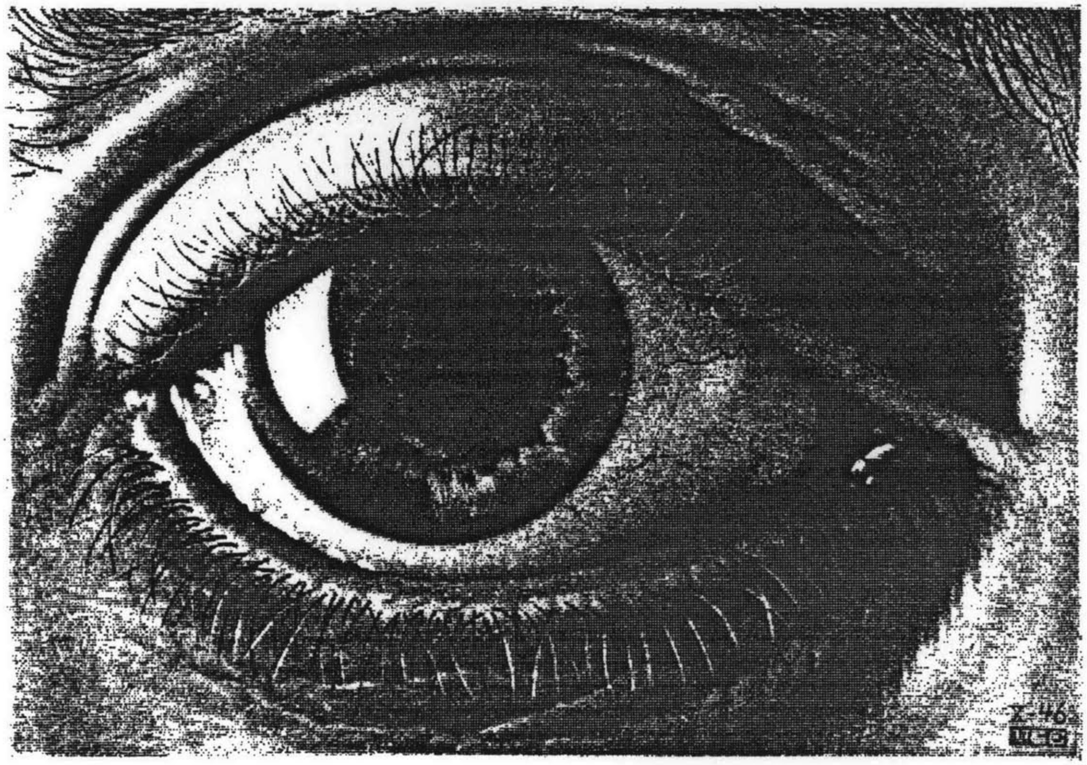


Figure 1.2

standard meanings and will always be used to refer to eye rotation with respect to a head fixed reference frame. Horizontal eye velocity corresponds to horizontal motion of the iris. This angular velocity is represented by a vector with its axis of rotation aligned with the z-axis of the head (ω_z). Similarly, vertical eye velocity will have its axis of rotation aligned with the y-axis of the head (ω_y), and torsional eye velocity will have its axis of rotation aligned with the x-axis of the head (ω_x).

II

ANALYSIS OF EYE MOVEMENTS



M. C. Escher

"There's language in her eye..."
William Shakespeare

"This shows how much easier it is to be critical than to be correct."
Benjamin Disraeli

CHAPTER TWO

ANALYSIS OF EYE MOVEMENTS

2.0 Introduction

In this chapter I present the methods which were used to analyze eye movements. Because I am particularly interested in estimating certain parameters (e.g. gains and time constants) related to slow phase eye velocity, I consider the analysis process to include three distinct stages:

1. Converting eye measurements to eye orientation and eye velocity.
2. Detecting and removing the fast phases.
3. Determining the gain and time constants of the slow phase eye velocity.

These three stages of processing will be discussed in sections 2.1, 2.2, and 2.3, respectively.

Conversion of eye measurements to eye orientation and then to eye velocity are complicated by the kinematics and kinetics of three dimensional rotations. Finite three-dimensional rotations are <u>not</u> commutative (i.e. the order of the rotations is important). Furthermore, the kinematics of rotation dictate that perpendicular measurements are not, in general, independent . Usually both of these facts are ignored, and this leads to errors which are usually small (<5%) but can be much larger as will be shown. In section 2.1 of this chapter I present a method which uses the measurements made using scleral search coils to find the exact three dimensional orientation and velocity of the eye. Test results are presented to confirm the theoretical calculations.

Detection and removal of fast phases from nystagmus is a complicated process which has been investigated by many researchers. The approach I take accounts for the three dimensional nature of the measurements. The software works in two sequential steps. In the first step, a method reminiscent of Michael's (1977)) is used to identify and remove almost all of the saccades. In the second step, a model fit is determined, and the residuals are evaluated to indicate the presence of saccades which were undetected in the first step. In section 2.2 of this chapter the methods are presented and algorithm performance is briefly discussed.

Determining the gain and time constants for slow phase eye velocity is inherently a nonlinear estimation process. In section 2.3 of this chapter, briefly discusses the algorithm used to estimate the parameters.

2.1 Calculation of Eye Orientation and Eye Velocity

2.1.1 Background

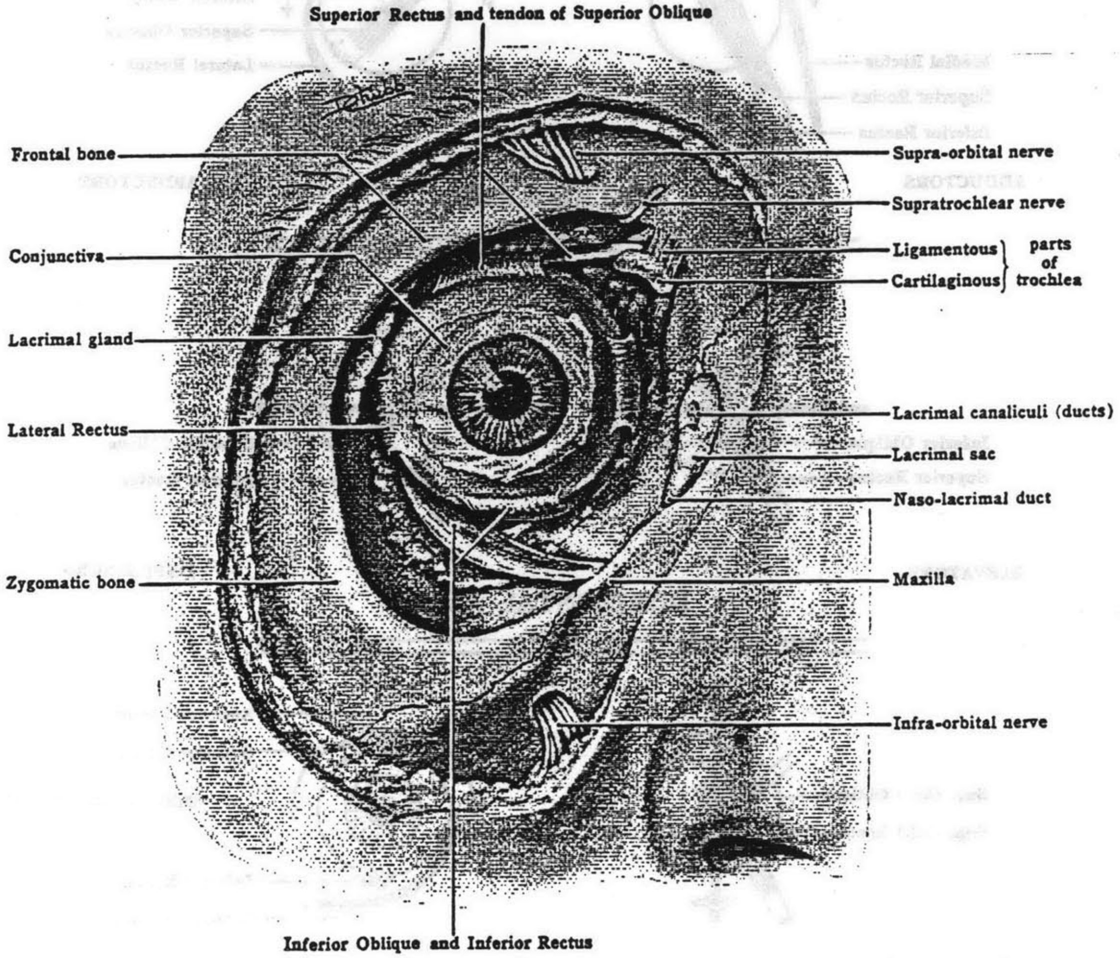
2.1.1.1 Eye Kinematics

The eyeballs, which are nearly spherical in shape, occupy the two orbital openings in the skull. (See figure 2.1.) Each eyeball is surrounded by fatty and connective tissue which allow the eyeball to rotate within the hard walls of the bony skull.

Rotation of the eye occurs when any one of the six extraocular muscles is activated. These muscles are often divided into three sets of push-pull pairs. One pair is formed by the superior rectus and the inferior rectus. These muscles insert, respectively, on the top and bottom of the eyeball and are primarily responsible for vertical eye movements. A second pair of muscles is formed by the medial rectus and the lateral rectus, which insert, respectively, medially and laterally. This pair of muscles induces horizontal eye movements. The final pair of muscles, the superior and inferior obliques, have slightly more complicated insertions. The inferior oblique passes across the lower half of the eyeball to insert on the lateral side of the eye. The tendon of the superior oblique muscle passes through a ring of cartilage called the trochlea, which is firmly attached to the skull. This tendon then continues on the rear of the eyeball to the the insertion point on the lateral side of the eye. The inferior and superior obliques are mainly responsible for torsional rotations of the eye. Figure 2.2 graphically shows the extraocular muscles and their points of insertion.

Because the eye is neither perfectly spherical nor fixed to rotate about a single point, the eye can translate and the center of rotation can shift. It is universally accepted that these

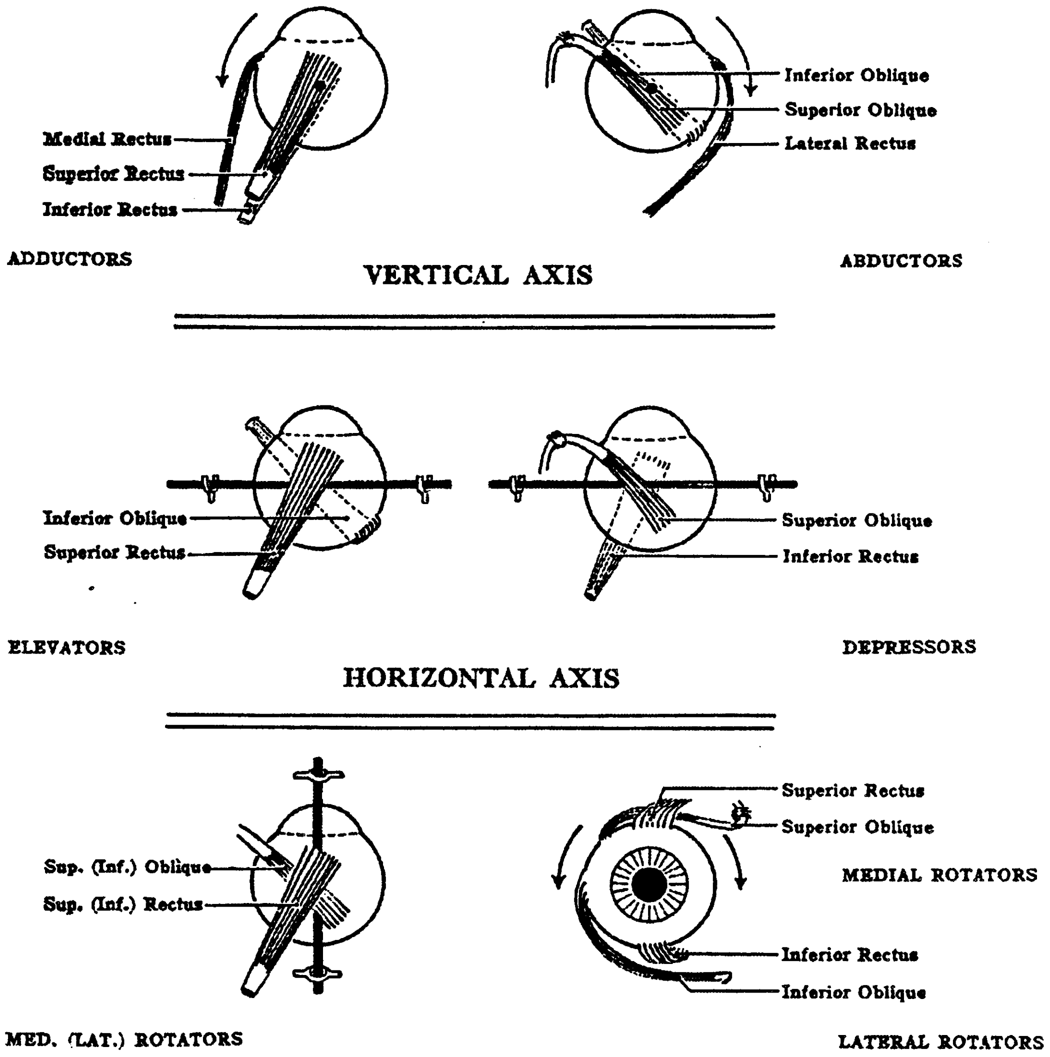
ORBITAL CAVITY



From: Anderson (1983)

Figure 2.1

EXTRAOCULAR MUSCLES



From: Anderson (1983)

Figure 2.2

effects are small, so we often assume that the eyeball is spherical, and that it rotates about a point fixed with respect to the eye and the head. Under these assumptions, eye motion has three degrees of freedom. Therefore, at least three parameters are required to completely define the orientation of the eye in the head.

The position and/or the rotation of the eye are not properly discussed unless a reference frame is assumed. For almost all eye rotations an appropriate reference frame is the skull since the actuators (extraocular muscles) are all fixed in the skull. This provides a reference frame which is independent of all exterior disturbances (e.g. head motion).

If the spherical eyeball has three degrees of freedom then, in general, at least three parameters are required to completely specify eye position and eye velocity. According to Alpern (1969), early researchers (Donders; Listing; and Helmholtz, 1925) suggested that eye position could be fully characterized with only two parameters. Invariably, these researchers constrained eye motion to voluntary control of gaze position which can be represented with only two parameters.

Three dimensional quantities like force and velocity are often represented by vectors having three components. It is, therefore, tempting to represent a three dimensional rotation by a vector. Mathematically, a quantity may be represented by a vector only if it commutes under addition. In mathematical notation:

$$\vec{x} + \vec{y} = \vec{y} + \vec{x} .$$

Force and velocity can be shown to obey the commutative law of addition, but it is easy to show that this does not hold for finite three dimensional rotations. The order of rotations is crucial unless the rotations are constrained to occur about a single axis or are infinitesimally small. Therefore, vectors cannot be used to represent finite rotations.

The rotation of a reference frame can be represented by a 3 x 3 matrix composed of direction cosines (called the rotation matrix or the direction cosine matrix), by a 4 parameter set called a quaternion (also known as Kayley-Hamilton parameters, Pauli spin matrices, Euler parameters, and the rotation group SU), or by a set of three Euler angles. Direction cosines and Euler angles are more physically intuitive than quaternions, therefore I will use these parameter sets for the theoretical development which follows. Some of the calculations are, however, implemented using quaternion theory.

It can easily be shown that angular velocity, representing an infinitesimal rotation divided by time, does obey the commutative law of addition. Therefore angular velocity can be represented by a vector. [For further discussion of rotational kinematics of the eye see Westheimer (1957) or Tweed and Vilis (1987). For a more general view of rotational kinematics and kinetics, see any analytical dynamics text (such as Hughes, 1986) or any classical mechanics text (such as Goldstein, 1959).]

2.1.1.2 Eye Measurement

For the experiments presented later in this thesis, scleral search coils (Robinson, 1963) were used to measure the orientation of the eye. Coil placement errors are, of course, unavoidable during the surgical procedure used to implant the coils. These placement errors can confound the measurement of eye orientation by introducing cross-talk between the measurements.

Haddad et al. (1988) evaluated the effect of coil placement errors and determined that with eye displacements of less than 10 degrees and with coil placement errors of less than 10 degrees, the eye position error was always less than 10%. By evaluating the distribution of eye position, they further estimated that the mean error is less than 5%.

For the data presented in the following chapter, the assumption that eye displacement is less than 10 degrees was often violated leading to large theoretical errors (approximately 30%) in eye velocity. Therefore, I needed to develop a method which corrects for coil placement errors to yield exact estimates for eye orientation and eye velocity. Theoretically, there are no errors inherent in this method of analysis since it is an exact solution of the kinematic and kinetic problems. This indicates that the resolution is limited only by the accuracy of the calibration procedures and by the inherent measurement limitations (e.g. measurement noise). Section 2 of Appendix B lists the C programs used to implement these algorithms.

2.1.2 Theoretical Derivation

2.1.2.1 Methods and Notation

Two right handed reference frames will be defined and used for these calculations. The first reference frame [x y z] is fixed in the head. The second reference frame [x' y' z'] is defined by the orientation of the eye coils. These two coordinate systems share a common origin which is fixed at the center of rotation of the eye. The orientation of the coordinate systems are identical when the eye is in the primary position. This orientation is shown in Figure 1.2.

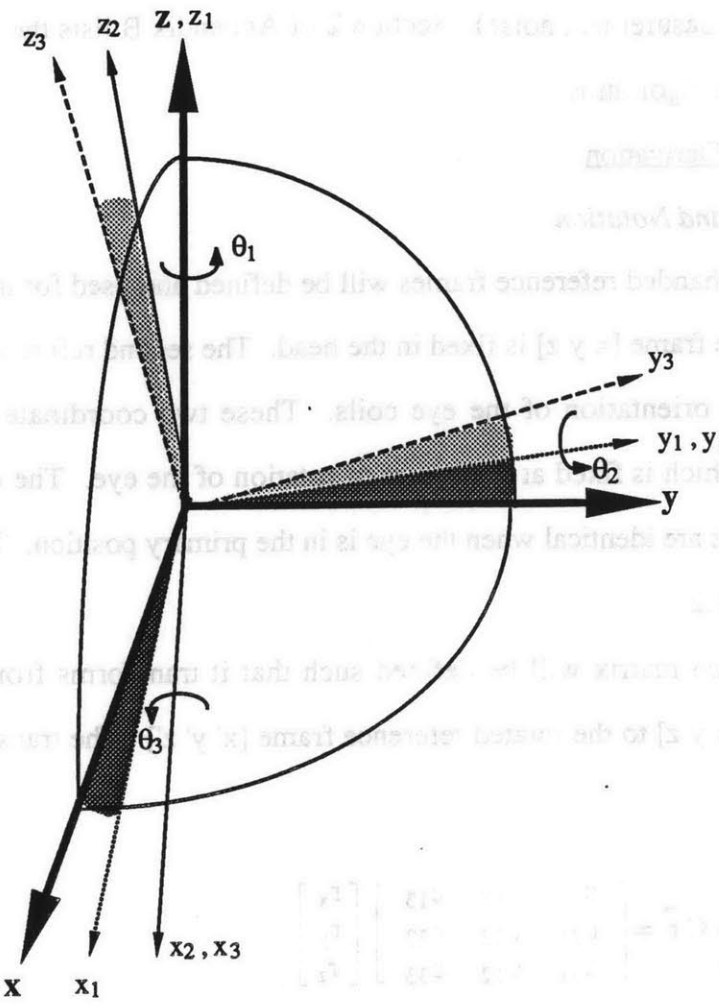
The rotation matrix will be defined such that it transforms from the head fixed reference frame [x y z] to the rotated reference frame [x' y' z']. The transformation can be written as:

$$\vec{r}' = C\vec{r} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}.$$

The Euler angle representation will be defined with the first angle (θ_1) defined as a rotation about the z-axis, the second angle (θ_2) defined as a rotation about the new y-axis, and the third angle (θ_3) defined as a rotation about the new x-axis. The sign convention used is that all rotations are defined positive when the rotation obeys the right hand rule. [θ_1 corresponds to Robinson's θ. θ_2 corresponds to Robinson's $-\phi$. θ_3 corresponds to Robinson's $-\psi$.] See Figure 2.3.

Each measurement in a Robinson coil arrangement is proportional to the cosine of an angle between two unit vectors. The first vector is defined by the direction of the magnetic field. The second vector is normal to the plane of the search coil. Two spatially orthogonal magnetic fields in phase quadrature allow us to measure two direction cosines

COORDINATE SYSTEM



Notation:
The sequence of Euler angle rotations is:

1. A rotation of θ_1 about z,
2. A rotation of θ_2 about y_1,
3. A rotation of θ_3 about x_2.

[x y z] is the initial reference frame.
[x_1 y_1 z_1] is the reference frame after a rotation of θ_1 about z.
[x_2 y_2 z_2] is the reference frame after another rotation of θ_2 about y_1.
[x_3 y_3 z_3] is the reference frame following the final rotation of θ_3 about x_2.
The final rotated reference frame [x' y' z'] is equivalent to [x_3 y_3 z_3].

Figure 2.3

with a single search coil. Each measurement will be proportional to the angle between the normal to the search coil and one of the magnetic fields. A second search coil, which doesn't overlap with the first, will allow us to measure two additional direction cosines as defined by the normal to the second coil and each of the magnetic fields. (See Robinson, (1963) for a more detailed discussion.) As will be shown, in an error-free and noise-free world, these measurements could be used to determine the four unknowns: the three directions of eye rotation and the angle between the search coils.

Notation

c'_{ij} represents the cosine of the angle between the i'th coil with the j'th direction in the [x y z] coordinate system. Coil 1 is defined to be the coil which encircles the iris of the right eye. The x-axis would pierce the center of the coil in the absence of coil placement errors. Coil 2 is the remaining coil which is lateral to the right eye such that, in the absence of placement errors, the negative y-axis would pierce the center of the coil. Direction 1 is defined by the x-axis which is fixed in the head, direction 2 is defined by the y-axis, and direction 3 is defined by the z-axis. With magnetic fields parallel to the y-axis and the z-axis, the four measurements yield c'_{12}, c'_{13}, c'_{22}, and c'_{23}.

c_{ij} represents the cosine of the angle between the i'th direction in the rotated [x' y' z'] coordinate system with the j'th direction in the [x y z] coordinate system.

The derivations which follow are divided into three parts. First, I will find the orientation of the coils with respect to the head. I will then use this result to determine the angular velocity of the eye with respect to the head. Finally, will correct for coil placement errors to find the orientation of the eye with respect to the head.

2.1.2.2 Coil Orientation

The goal of the following calculations is to determine the Euler angle representation of coil rotation with respect to the head when the measurement coils are not orthogonal. When the head is fixed with respect to the field coils, as is usually the case, the problem can be reduced to two simpler problems. First, use four measurements to determine the angle between the two eye coils. Once this angle is known, determine the coil orientation using three measurements.

First we will determine the orientation of the coils in the head. Using solid geometry we can show that:

$$(c'_{11})^2 + (c'_{12})^2 + (c'_{13})^2 = 1 \, .$$

Because the eye cannot rotate beyond 90 degrees, a further constraint is provided by:

$$c'_{11} > 0 \, .$$

Therefore:

$$c'_{11} = +\sqrt{1 - (c'_{12})^2 - (c'_{13})^2} \, .$$

A unit vector pointing from the center of rotation through the center of the first search coil ($\vec{c'_1}$) can be determined. Mathematically, it is written as:

$$\vec{c'_1} = \begin{bmatrix} c'_{11} \\ c'_{12} \\ c'_{13} \end{bmatrix} .$$

where c'_{12} and c'_{13} are measured and c'_{11} is calculated.

Similarly for the second coil we know that:

$$(c'_{21})^2 + (c'_{22})^2 + (c'_{23})^2 = 1 \, .$$

Or:

$$c'_{21} = \pm\sqrt{1 - (c'_{22})^2 - (c'_{23})^2} \quad .$$

Either solution is physically possible depending on the position of the eye. Therefore more information is required. An additional constraint is that the angle between the two eye coils is constant. Therefore, the dot product of the vectors representing the position of the eye coils must be constant. Mathematically this can be written as:

$$\vec{c'_1} \cdot \vec{c'_2} = c'_{11}c'_{21} + c'_{12}c'_{22} + c'_{13}c'_{23} = \cos(\gamma) = k \,.$$

Substituting the previous expression for c_{12}, we find that:

$$\cos(\gamma) = c'_{12}\, c'_{22} + c'_{13}\, c'_{23} \pm c'_{11} \left| c'_{21} \right| \,.$$

Since cos(γ) is a constant, one solution of this equation will be constant. It can easily be proven that this equation has only one constant solution. The alternate solution will vary with eye position. Thus, I can determine the angle between the two eye coils.

Once I have determined the angle between the eye coils, only three measurements are needed to completely define the orientation of the search coils in the head. The necessary measurements are those defined by Robinson (1963). The necessary calculations are shown below.

As before, I begin by determining the unit vector pointing from the origin through the first coil. If I arbitrarily define this axis to be the x'-axis, then:

$$\vec{x'} = \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \end{bmatrix} = \vec{c'_1} = \begin{bmatrix} c'_{11} \\ c'_{12} \\ c'_{13} \end{bmatrix} .$$

As before, I know that:

$$(c'_{21})^2 + (c'_{22})^2 + (c'_{23})^2 = 1$$

and that:

$$\vec{c'_1} \cdot \vec{c'_2} = c'_{11}c'_{21} + c'_{12}c'_{22} + c'_{13}c'_{23} = \cos(\gamma) = k \, .$$

These two equations can be simultaneously solved for tne two unknowns $(c'_{12}$ and $c'_{22})$. Due to the squares which occur in the first equation two solutions are possible. For eye movements which are limited to rotate less than 90 degrees from the resting position, only one of the solutions is possible. Therefore:

$$c'_{22} = \frac{c'_{11}\sqrt{-k^2 + 2c'_{13}c'_{23}k + (c'_{12})^2 + (c'_{11})^2 - (c'_{23})^2} + c'_{12}k - c'_{12}c'_{13}c'_{23}}{(c'_{12})^2 + (c'_{11})^2}$$

and:

$$c'_{21} = \frac{k - c'_{12}\,c'_{22} - c'_{13}\,c'_{23}}{c'_{11}}$$

A unit vector pointing from the center of rotation through the second coil can now be defined. It can be written as:

$$\vec{c'_2} = \begin{bmatrix} c'_{21} \\ c'_{22} \\ c'_{23} \end{bmatrix} .$$

Next we need to determine the rotation matrix which transforms from the head fixed coordinate system to the coordinate system defined by the eye coils. Rotation matrices are easily defined for orthogonal coordinate system, but since the search coils are not orthogonal, I will develop an orthogonal coordinate system which is fixed with respect to the search coils.

Any two vectors define a plane. Hence $\vec{c_1}$ and $\vec{c_2}$ define a plane. A unit vector can be found which is perpendicular to this plane. This vector is defined to be the z' axis. z' is found as shown below:

$$\vec{z'} = \begin{bmatrix} c_{31} \\ c_{32} \\ c_{33} \end{bmatrix} = \frac{\vec{c_1'} \times -\vec{c_2'}}{\left|\vec{c_1'} \times \vec{c_2'}\right|} .$$

Since the coordinate system is defined to be right handed:

$$\vec{y'} = \begin{bmatrix} c_{21} \\ c_{22} \\ c_{23} \end{bmatrix} = \vec{z'} \times \vec{x'} .$$

Therefore we have developed an orthogonal coordinate system [x' y' z'], which is fixed in the eye and is defined by the orientation of the coils. The orientation of [x' y' z'] with respect to [x, y, z] can be represented by a rotation matrix (C_{ch}) where:

$$C_{ch} = \begin{bmatrix} \vec{x'}^T \\ \vec{y'}^T \\ \vec{z'}^T \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} .$$

This rotation matrix represents the orientation of the search coils with respect to the head. If the Euler angle representation is desired, it can be calculated from the following equations:

$$\theta_2 = \sin^{-1}(-c_{13})$$

$$\theta_1 = \sin^{-1}(c_{12}/\cos(\theta_2)$$

$$\theta_3 = \sin^{-1}(c_{23}/\cos(\theta_2).$$

This method exactly determines the orientation of the coils with respect to the head and will be used to calculate eye velocity.

2.1.2.3 Eye Velocity

The goal of the following calculations is to determine the angular velocity of the eye with respect to the head. Since the eye coils are fixed to the eye, the angular velocity of the coils equals the angular velocity of the eye. For reasons of simplicity and accuracy, we will simply calculate the angular velocity of the eye coils.

A number of possible approaches are available to determine the rate of angular rotation when angular orientation is known. With additional manipulation, differentiation of the 9-element rotation matrix, the 4-element quaternion, or the 3-element Euler angle can yield angular velocity. I have chosen the approach of differentiating the the Euler angle representation because it requires the least computation and because it offers the most physically intuitive approach.

Differentiating each of the three previously defined Euler angles we obtain:

$$\omega_1 = \frac{d\theta_1}{dt} ,$$

$$\omega_2 = \frac{d\theta_2}{dt} ,$$

$$\omega_3 = \frac{d\theta_3}{dt} .$$

Each of these Euler rates is a vector quantity. ω_1 has a direction which aligned with the head fixed z-axis, $\dot{\omega_2}$ is aligned with the y_1 axis, and ω_3 is aligned with the x_2 axis. (See figure 2.3)

These non-orthogonal angular velocities can easily be converted to the head fixed coordinate system [x y z] by the transformation:

$$\vec{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = S(\theta_1,\theta_2)\,\dot{\vec{\theta}} = S(\theta_1,\theta_2) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix},$$

where,

$$S(\theta_1,\theta_2) = \begin{bmatrix} 0 & -\sin(\theta_1) & \cos(\theta_1)\cos(\theta_2) \\ 0 & \cos(\theta_1) & \sin(\theta_1)\cos(\theta_2) \\ 1 & 0 & -\sin(\theta_2) \end{bmatrix}.$$

The transformation matrix can also be inverted yielding:

$$\dot{\vec{\theta}} = S^{-1}(\theta_1,\theta_2)\,\vec{\omega}\,,$$

where,

$$S^{-1}(\theta_1,\theta_2) = \begin{bmatrix} \dfrac{\cos(\theta_1)\,\sin(\theta_2)}{\cos(\theta_2)} & \dfrac{\sin(\theta_1)\,\sin(\theta_2)}{\cos(\theta_2)} & 1 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \\ \dfrac{\cos(\theta_1)}{\cos(\theta_2)} & \dfrac{\sin(\theta_1)}{\cos(\theta_2)} & 0 \end{bmatrix}.$$

This method calculates the angular velocity of the eye given the Euler angle representation of eye coil orientation.

2.1.2.4 Eye Orientation

The goal of the following calculation is to determine the orientation of the eye with respect to the head from measurements which represent the orientation of the coils with respect to the head.

Using rotation matrices we can show that the rotation matrix describing the orientation of the coils with respect to the head equals the product of the rotation matrix of the coils with respect to the eye times the rotation matrix of the eye with respect to the head. This is written as:

$$C_{ch} = C_{ce}C_{eh} \quad .$$

It is easily shown that:

$$C_{eh} = C_{ec}C_{ch} = C_{ce}^{T}C_{ch} \ ,$$

where the superscript represents the transpose operation.

The orientation of the coils with respect to the eye is constant. Therefore, experiments may be conducted to yield the rotation matrix which represents the orientation of the coils with respect to the eye (C_{ce}) (e.g. have the subject look a point which is directly in front of the eye). Once this error quantity is experimentally determined, we can easily calculate the absolute eye orientation with respect to the head.

The basic procedure to determine the orientation of the eye in the head is:

1. Use four measurements to determine the angle between the two eye coils.
2. Use three measurements to find the orientation of the coils in the head (C_{ch}).
3. Use the error matrix (C_{ce}) to correct for the coil placement errors.

This simple procedure yields the orientation of the eye with respect to the head.

If an Euler angle (θ_1,θ_2, and θ_3) representation of eye orientation is desired. It can easily be determined from the rotation matrix (C_{eh}):

$$\theta_2 = \sin^{-1}(-c_{13})$$

$$\theta_1 = \sin^{-1}(c_{12}/\cos(\theta_2)$$

$$\theta_3 = \sin^{-1}(c_{23}/\cos(\theta_2).$$

2.1.3 Practical Considerations

In the previous section, a procedure which calculates the exact orientation and velocity of the eye from four error-free measurements was developed. In practice, measurement noise and calibration difficulties make four error-free measurements impossible.

As previously discussed, a search coil system measures the direction cosines between the axis of the coil and the magnetic field. The standard coil configuration sets the orientation of the search coils such that three of the measurements yield the maximum possible sensitivity; i.e. such that:

$$\frac{d}{d\theta_i}(\cos(\theta_i)) = -\sin(\theta_i)$$

is at an extrema. This occurs when θ_i is ±90°. However, if three of the coil measurements are maximally sensitive to changes in orientation, geometrical considerations show that the fourth measurement will demonstrate absolute insensitivity to changes is orientation (i.e. $\sin(\theta_4)$ equals zero). In the presence of noise and other measurement errors, this leads to obvious difficulties.

Therefore, an alternate method is presented to estimate the angle between the eye coils. Once the angle between the coils is determined, we will use the three maximally sensitive measurements to exactly estimate eye orientation.

The basic idea is to apply a controlled rotational stimuli and measure the response. It is widely known that the response normally aligns with the stimulus (e.g. yaw rotations yield horizontal eye motion while pitch yields vertical eye motion). By estimating values of the angle between the eye coils we will effectively change the axis of the response. By minimizing the difference between the axis of the response and the axis of the stimulus, we obtain a very good estimate of the angle between the eye coils.

I will use an example to both demonstrate the method and verify its accuracy. Previous researchers (Haddad et al, 1988) have shown that coil placement errors appear on the roll measurement during pitch stimulation. Theoretically we predict that the roll velocity should be zero during pitch. By iteratively estimating a value for the angle between the coils and minimizing the RMS value of the estimated roll response we can determine the angle between the coils.

The method was validated using the calibration jig shown in figure 2.4 (Haddad et al., 1988). Two coils were placed on the jig such that the angle between the coils was nominally 80° with less than 2° of inaccuracy. Figures 2.5a and 2.5b show the measurements of pitch and roll during a pitch of the calibration jib before any corrections were made. Notice the large amount of roll indicated even though the motion doesn't have a roll component. Now we estimate the value for the angle between the coils and measure the RMS of the roll channel. This process of estimating the angle between the coils and determining the RMS error is repeated until the RMS error is minimized. In this example, the value of the angle which minimized the RMS was 78.5°. This accuracy was well within the range of experimental error. Figure 2.5d shows the estimated roll response with an estimated angle of 78.5°.

This process was repeated with the angle between the coils nominally set at a value of 90° and 100°. The angle between the coils was estimated to be 89.5° and 98.5°, respectively. This accuracy was judged sufficient.

CALIBRATION JIG

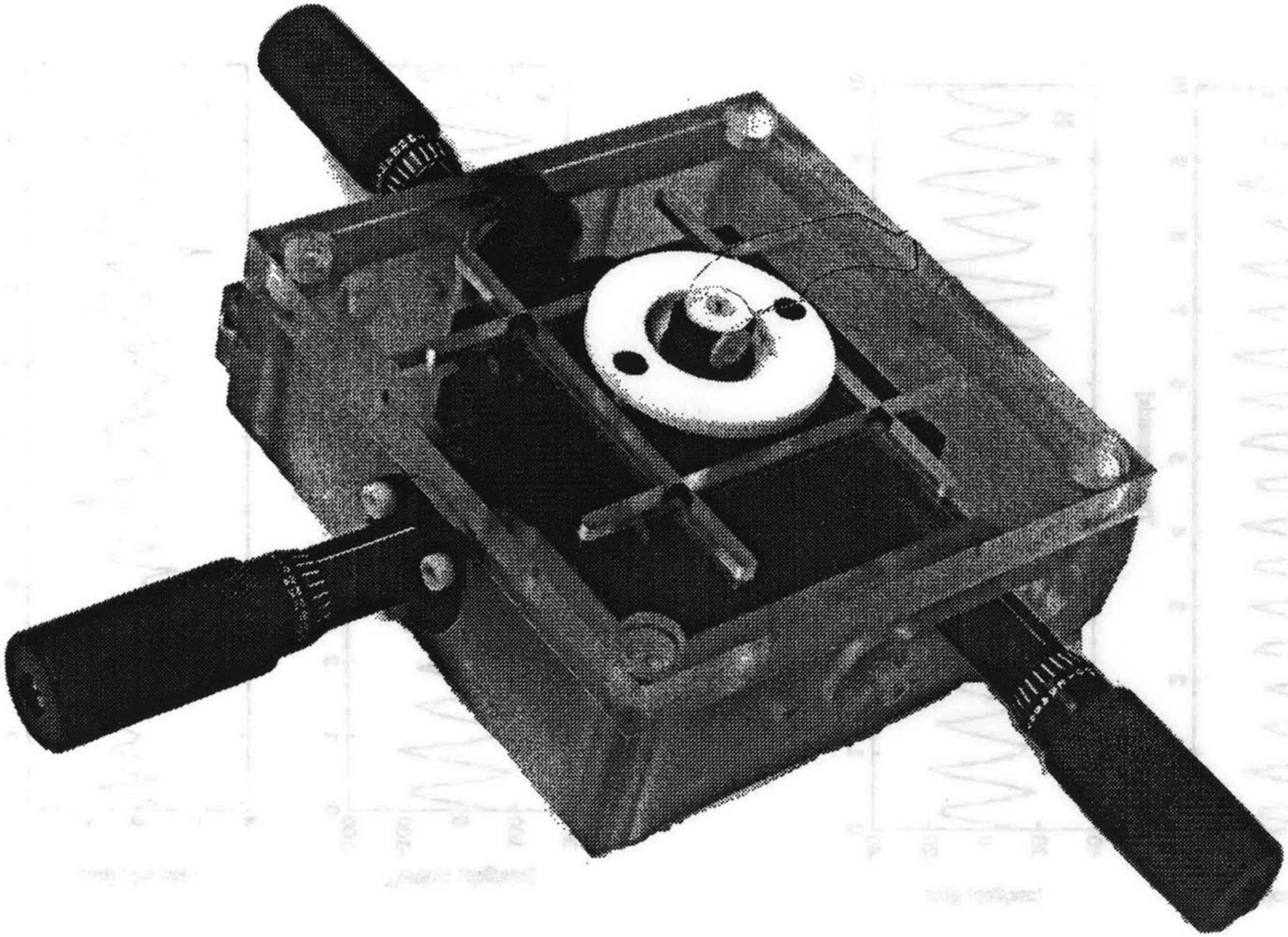
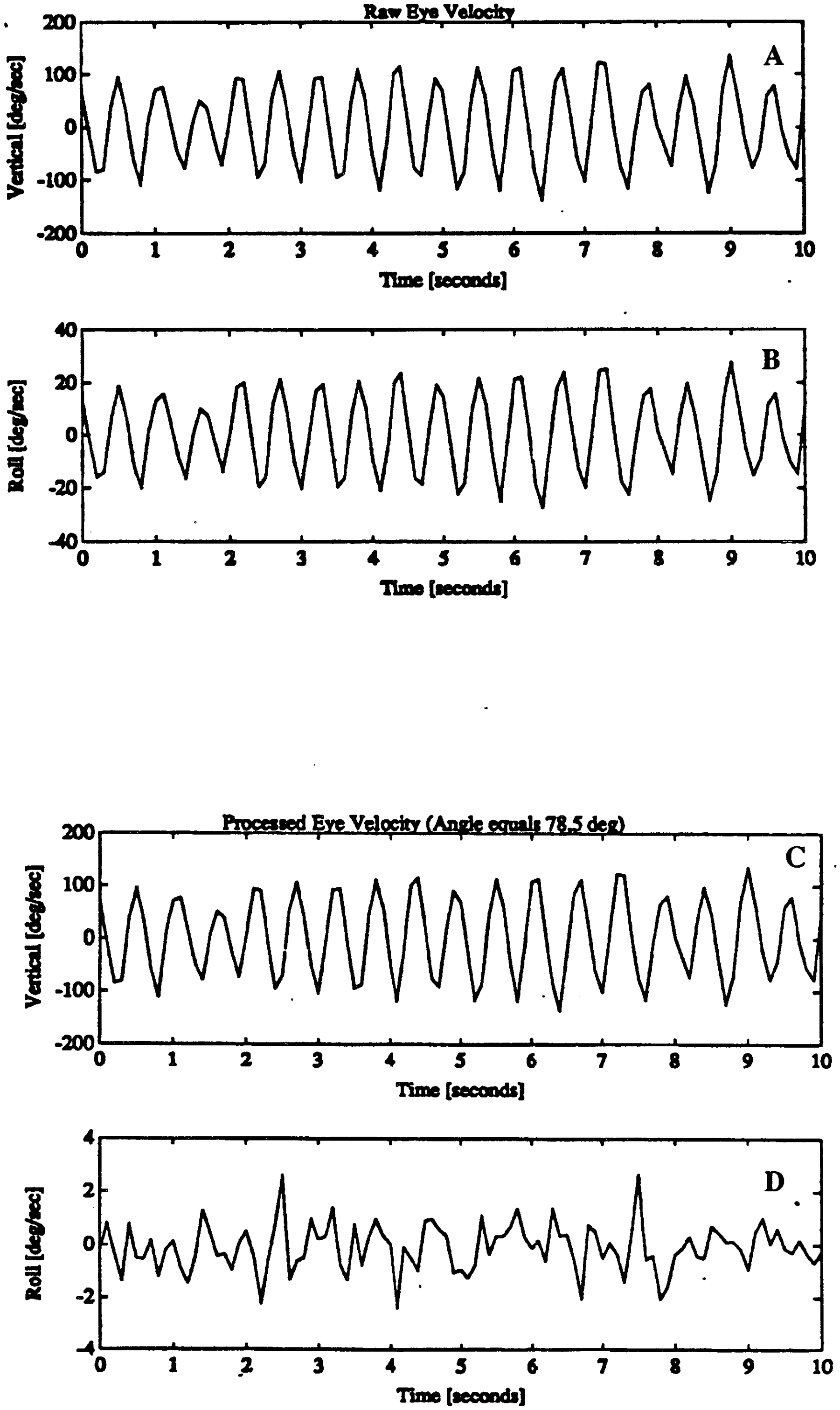


Figure 2.4

TEST WITH CALIBRATION JIG



Raw Eye Velocity

Processed Eye Velocity (Angle equals 78.5 deg)

Figure 2.5

An identical process was used to estimate the angle between the coils in vivo. Figure 2.6 shows the uncorrected data and the corrected roll response for specimen F. This estimation process was carried out every test session for every specimen, and, as expected, was relatively stable over time for any single specimen.

2.2 Fast Phase Detection and Removal

Conjugate eye movements, the simultaneous motion of both eyes in the same direction, are an easily accessible physiological response which are often measured by researchers and clinicians. Large amplitude (> 1 degree) conjugate eye motion can be broken down into three categories: fast eye movements, slow eye movements, and nystagmus.

Fast eye movements include saccades and blinks. Saccadic eye movements begin at rest, accelerate quickly to a high velocity, and then decelerate quickly to a stop. Often the movement'are completed in less than 100 msec. These eye movements are ballistic in nature, in that, once begun, they cannot be altered. This type of motion is made while reading or when fixating stationary objects. Blinking results in a quick upward and then downward movement of the eye to near its original position. This type of motion generally is completed in less than 250 msec.

Slow eye movements show much more gradual accelerations. This type of eye movement can reach high velocities (> 50 degree per second), but they accelerate to these velocities much more gradually than saccades. Unlike saccades, these motions are not ballistic, and they are often used to track slowly moving objects.

Nystagmus is a type of eye movement which demonstrates a saw-tooth wave pattern. These eye movements combine components from both fast eye movements and slow eye movements. During nystagmus, slow eye movements (slow phases) and saccades (fast phases) alternate. This results in a saw-tooth appearance of the waveform. This type of eye movement is made during large whole-body rotations and during large amplitude motions of the visual field.

TEST WITH SLOW PHASE VELOCITY

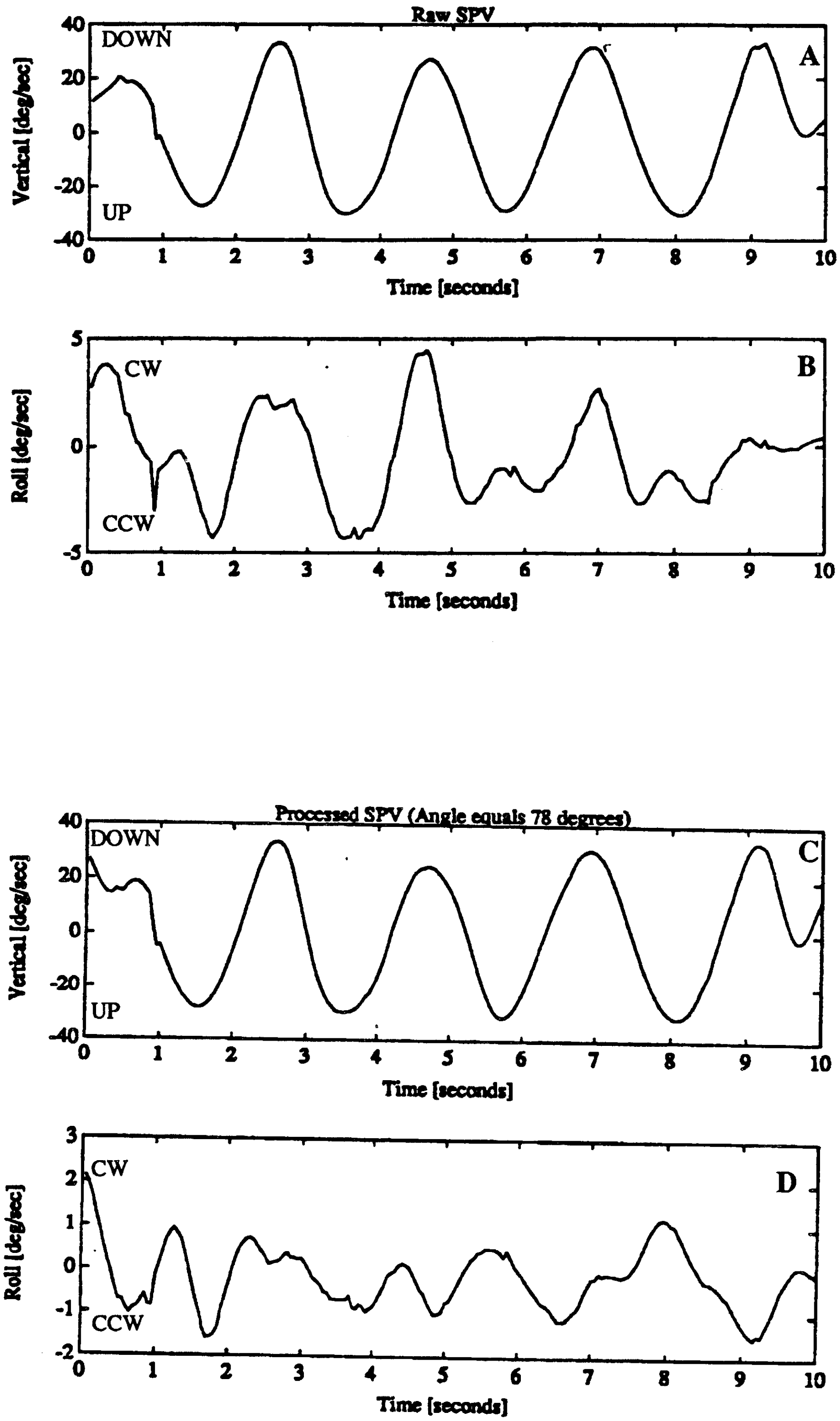


Figure 2.6

Since both the slow and fast components may contain information of interest to the researcher or clinician, it is useful to separate these components of nystagmus. Analysis of nystagmus (i.e. separation of fast and slow components) is a very time consuming manual task. Since the slow phases and fast phases share a significant band of the frequency spectrum, simple analog or digital filtering is not adequate to separate these components (Massoumnia, 1983). Therefore, more advanced digital methods are required to detect the presence of each of the components and to separate them. Furthermore, it has been shown that manual analysis of nystagmus leads to large person-to-person variability (Gentles and Barber, 1973), since human operators have personal biases.

For these reasons, the analysis of nystagmus is an important process. The digital computer seems particularly well suited to assist in this task.

2.2.1 Background

Because a large number of people study eye movements for a host of different purposes, a number of different approaches have been developed to analyze eye movements. These analysis techniques are difficult to classify because most modern analysis systems are hybrid systems that combine elements that cross any arbitrary classification boundaries. Nevertheless, it is helpful to attempt to group the analysis methods approximately. Hybrid systems that cross the arbitrary boundaries will be classified into the category which I feel best describes the most important system characteristics.

The analysis of nystagmus uses three classes of methods : manual, semi-automated, and automated. Manual methods are defined as all analog methods even though a number of short cuts have been used in the analog methods. Semi-automated methods are defined as those which, in some way, make use of the data processing capabilities of the computer, but require frequent interaction with a well-trained human operator. Automated methods are defined as those in which most, if not all, of the processing is performed by a computer, with little or no human intervention.

2.2.1.1 Manual Analysis

The simplest manual method of nystagmus analysis is hand analysis of eye position records. These methods use hand measurements to find the slope, and thus the velocity, of the slow or fast component of the nystagmus to yield a beat by beat estimate of the slow or fast phase of the velocity. In another similar type of method the number of fast phases are simply counted to yield an estimate of the strength of nystagmus.

The next level of manual methods introduces an analog differentiation of the eye position record to yield an estimate of eye velocity (Henriksson, 1955). Slow phase eye velocity can be manually scored from these records for further analysis.

Other simple analog devices are also used. These devices (Guedry et al, 1968; Voots, 1969) are relatively easy to use, but inflexible. The analysis yielded by these methods is limited to only one component of the response, and the methods may lead to operator dependent errors which, unintentionally lead to biased results.

The manual methods provide quick and and reasonably accurate results, but they are labor intensive, time consuming, and require highly trained technicians. Before inexpensive access to computers, however, they were the best choice available.

2.2.1.2 Semi-automated Analysis

The first semi-automated method was actually a hybrid analysis system (Anzaldi et al, 1975). A simple automated saccade detection algorithm used a difference calculation to estimate eye velocity. If the estimated eye velocity had a sign opposite that input by an operator and was larger than a threshold value, a saccade was indicated.

The most interesting feature of this system was that it included algorithms, particularly graphics packages, which allowed a trained operator to inspect and, if necessary edit the computer analyzed data.

The major problem of this algorithm relates to the automated portion of the analysis. The automated analysis assumed unidirectional nystagmus (i.e. slow phases in one direction and fast phases in the other). Unidirectional nystagmus is relatively rare and this severely limits the usefulness of the automated portion of the program. The operator interaction with the computer output, however, is a very important feature of the procedure.

Another method using a semi-automated approach took advantage of knowledge of the input stimuli and the types of expected nystagmus responses to perform detection (Barnes, 1982). It relied on the pattern recognition capabilities of a human operator to detect fast phases. This system assumes normal linear transfer functions relating a slow phase velocity output to the input stimulus. Given an input waveform and a system transfer function, the predicted slow phase velocity output is easily calculated. Using cursor controls and graphic displays, the operator can set up thresholds on both sides of the predicted slow phase velocity. Points lying outside of the threshold limits are discarded and a least squares algorithm is used to interpolate slow phase velocity.

This procedure inherently assumes that the oculomotor response is normal. This often is not the case and for clinical data, where abnormal data may be the norm, it certainly is not an acceptable assumption.

Both of the semi-automated methods are labor intensive. However, even automated methods must allow for rudimentary manual editing.

2.2.1.3 Automated analysis

Several types of automated systems have been discovered independently by different groups. One common type of system calculates a difference or digitally differentiates eye position to obtain an estimate of eye velocity (Tole and Young, 1971; Baloh et al., 1976; Ni, 1980). Detection is performed by setting velocity thresholds. Another type of algorithm looks for alternating maxima and minima of the eye position trace (Honrubia et al., 1971; Tokita et al. 1975; Wall et al., 1981). The data between these peaks are evaluated to see if it represents a fast or slow component or some other type of event. A final type of system performs two digital differentiations to yield an estimate of eye acceleration (Gentles, 1974; Michaels, 1977; Massoumnia, 1983). Detection is performed by setting appropriate acceleration thresholds. This section will be organized with these class boundaries in mind.

Velocity Detection

A group of researchers at an ENT clinic in Sweden were among the first to recognize the role that computers could play in analyzing nystagmus and to write an algorithm (Herberts et al, 1968). The method is not precisely defined, but it appears that a difference calculation is performed and an appropriate threshold is set to detect the fast phases.

A similar, but more developed set of programs, called MITNYS (Tole and Young, 1971) and MITNYS-II (Allum et al., 1975), were developed by researchers at MIT. MITNYS-II used digital differentiation to estimate eye velocity. On the basis of the direction and velocity of eye movements a point was classified as a slow movement or a fast movement. This algorithm used a closed-loop feedback of the program's current estimate of the slow phase eye velocity. This sometimes caused instabilities. The program's output compared reasonably to the analysis performed by an experienced physician.

During the same period, a similar set of algorithms was developed independently by a group at UCLA (Baloh et al, 1976; Baloh et al., 1980). This group further specified that saccades were those eye movements which exceeded a velocity threshold for more than a minimum duration. In addition, these researchers provided means for manually editing the computer results.

A later group used similar techniques, but created a type of synthetic data for further analysis (Hansson et al, 1982). Fast phases were detected with a velocity threshold method. Once fast phases were detected, linear least square fits were performed on the slow and fast components of the response. The intersections of these lines indicated the turning points of the nystagmus. The linear least square fits were used to form "synthetic" nystagmus. The synthetic segments were classified as slow or fast phases using velocity and duration criteria. Finally the synthetic nystagmus is stored for further analysis. The authors claimed that the synthetic nystagmus fully represented the original nystagmus while drastically reducing computer memory requirements.

An interesting twist to this approach for saccade detection used a high pass filter to remove the slow phase components of the eye movement (Ni, 1980). A velocity threshold test was applied to determine fast phases, and a least squares curve fit interpolated within the saccadic interval.

Position Detection

The alternating minima and maxima of eye position data for nystagmus analysis were first used by a group of UCLA researchers (Honrubia et al., 1971; Sills et al., 1975). They calculated simple differences on the eye position data, and detected points of interest by changes in sign of that computed difference. These points of interest were then checked with threshold and minimum duration test to detect saccades.

A similar algorithm was developed independently at nearly the same time (Tokita et al., 1975). Again points of interest were chosen based upon sign changes of a difference calculation, and further tests were applied to verify that a saccade had occurred.

Wall and Black (1981) utilized similar methods. Alternating minima and maxima were detected using a simple difference technique. These points of interest were connected using straight line segments to yield a form of synthetic nystagmus. Slope and displacement thresholds are applied to the synthetic data to identify the fast phases. Further analysis is performed on the original data record where simple interpolation methods are used to fill the saccadic intervals. It is important to note that the synthetic nystagmus is used for saccade detection only and is not substituted for the original data record.

Another approach (Baland et al., 1987), like the position methods described previously, begins by identifying the local maxima and minima of eye position. The magnitude of the difference is compared to an ad hoc threshold which is chosen as one half of the mean value of the derivative calculated for the whole trace. (The authors claim that this threshold is related to the quality of the signal.) The data between the maxima and minima are numbered and the duration of the slow or fast component is evaluated. A phase is assume slow if its duration is long. These slow phases are fit with a sinusoidal curve having the same frequency as the stimulus. All previously labeled components are compared to this fit and the errors are evaluated. Components with large errors are rejected. This iterative process is repeated until no previously accepted phases are rejected. The major limitation of this approach is that it is limited to sinusoidal data analysis.

Acceleration Detection

Gentles (1974) was the first researcher to use eye acceleration as part of the process of analyzing nystagmus. A difference operation yielded an estimate of eye velocity, and a velocity threshold was used to determine the presence of saccades. A second difference operation yielding eye acceleration was performed, and the acceleration peaks were used to define the beginning and end of the saccade. A number of ad hoc adaptive features were used to give greater immunity to errors. The program output agreed fairly well with manual scoring.

In another approach (Michaels, 1977), the author used FIR filtering to detect fast phases. Eye position was convolved with a 9 point integer-valued finite impulse response (FIR) filter, and a threshold test was applied to the output of the filter. Frequency response of the matched filter showed that at low frequencies (0-7 Hz) the filter acted as a double differentiator. (The filters could have been modeled as a double differentiator in series with a low-pass filter.) The program analysis correlated well with the analysis performed by an experienced physician.

Later, this method was improved by redesigning the digital filters (Massoumnia, 1983) to yield better and less noisy estimates of eye acceleration. Because the eye acceleration signal has near-zero mean, detection was performed using eye acceleration and eye velocity.. The acceleration threshold used was an ad hoc sum of RMS noise and a value input by the system operator. A careful comparison of human scoring and algorithm analysis showed that for a slow phase velocity of greater than 20 degrees/sec, the program detected 100% of human scored saccades. But for slow phase velocity less than 10 degrees/sec, the program only detected 80% of the saccades detected by a trained human.

Perhaps more importantly, these authors (Michaels, 1977; Massoumnia, 1983) recognized that estimates of peak slow phase velocity or the total number of detected saccades are not good measures of algorithm performance. Instead, the probability of event detection, the probability of missing an event, and the probability of false alarms should be used to evaluate algorithm performance.

2.2.2 Methods

As scientists and engineers, we artificially divide eye movements into the orthogonal components that we measure (usually horizontal and vertical components and, sometimes, the torsional component). Of course, as previously discussed, the actual eye movements are the rotation of a sphere-like eyeball which may have components of motion along any of the measurement axes. Two or three orthogonal measurements divide the vector of eye motion up into its Cartesian coordinates. By analyzing each of the components individually we ignore important information available to us during the analysis of nystagmus.

The analysis approach used in this thesis attempts to make use of all available information while analyzing nystagmus. This approach transforms the Cartesian measurements to a spherical coordinate system. In spherical coordinates the magnitude of velocity or acceleration is represented by a single parameter and is independent of the direction in which the vector points. The total magnitude of eye acceleration is compared to a threshold set by the user. If the acceleration exceeds the threshold a "fast phase" is detected and removed. This simple process detects almost all of the fast phases in the data sets which were analyzed. This step is further discussed in section 2.2.2.1.

After detecting the majority of the fast phases, a second processing step is utilized to remove most of the saccades missed by the first step of processing. This second pass at the data is done using a least square parameter estimation process (Ljung, 1987). The residuals of the model fit are compared to a threshold set by the user. If the residual is larger than the threshold an "event" is detected and removed. This processing step is further discussed in section 2.2.2.2.

All of the fast phase detection and removal algorithms were implemented in Matlab (copyright by Mathworks) using the macro (M-file) capabilities of Matlab. In the entire data set, a few saccades (< 10) were manually removed.

2.2.2.1 Acceleration Detection (1st Pass)

A literature review and personal inquiry have convinced me that eye acceleration is the most appropriate domain to detect events if the signal is relatively noise free (like the search coil measurements analyzed in Chapter 3). Previous users of this approach (Michaels, 1977; Massoumnia, 1983), digitally differentiated eye position twice to yield an estimate of eye acceleration. A threshold was set based upon user experience and the quality of the signal. The major difference between this approach and Massoumnia's algorithm is that I utilize a spherical coordinate system to calculate the total magnitude of eye acceleration. All previous approaches, to my knowledge, analyzed each component of the eye motion separately.

There are a number of advantages of a multi-dimensional approach to the analysis of eye movements. These are:

> 1) In contrast to a Cartesian analysis, which is biased toward the detection of "events" aligned with the measurement axes, the analysis in spherical coordinates is unbiased in direction.
>
> 2) Beginnings and endings of "events" are more easily determined (e.g. slow saccades often occur because a blink or another saccade occur in another axis).
>
> 3) Computation time will be reduced if the detection algorithm is run only once instead of once for each measurement axis.

Each of these advantages is briefly addressed below.

Advantage # 1

The bias of the Cartesian system of analysis is demonstrated by an oblique saccade (i.e. any saccade whose direction is not aligned with the measurement axes). This type of saccade will have components of its acceleration which project onto at least two of the measurement axes, and each of the projections will always be less than the total magnitude of the saccade. Another saccade, identical in all respects except that it is aligned with a measurement axis, will have a larger component along the measurement axis than the oblique saccade. This saccade will be more easily detectable than an oblique saccade. This shows that Cartesian analysis is biased toward events which occur along the measurement axes.

Advantage #2

Figure 2.7 shows a "slow" saccade. If we look only at the horizontal component of the saccade, the end of the saccade is very difficult to detect. If we also include the vertical component of the eye movement we observe that the "slowness" of the eye movement is explained by a nearly simultaneous vertical saccade. The beginning and the end of the event are more easily detected when both horizontal and vertical signals are utilized.

Figure 2.8 shows another "slow" horizontal saccade. Again, the end of the saccade is difficult to detect if we ignore the vertical eye movement. And once again, if the vertical eye movement is included in our analysis, we see that the "slowness" of the eye movement is explained by a nearly simultaneous blink (Bell's phenomenon). The beginning and the end of the event are more easily detected when all information is utilized.

Figures 2.7 and 2.8 are not rare occurrences. Both were found in a single data set along with many other similar events. A very rough estimate for eye movements of squirrel monkeys indicates that at least 3% of fast phases behave like these "slow" events.

Advantage #3

If more than one component of eye motion is to be analyzed, computational time can be reduced. It obviously takes less time to run an algorithm once than it would to run

SLOW HORIZONTAL SACCADE (with Vertical Saccade)

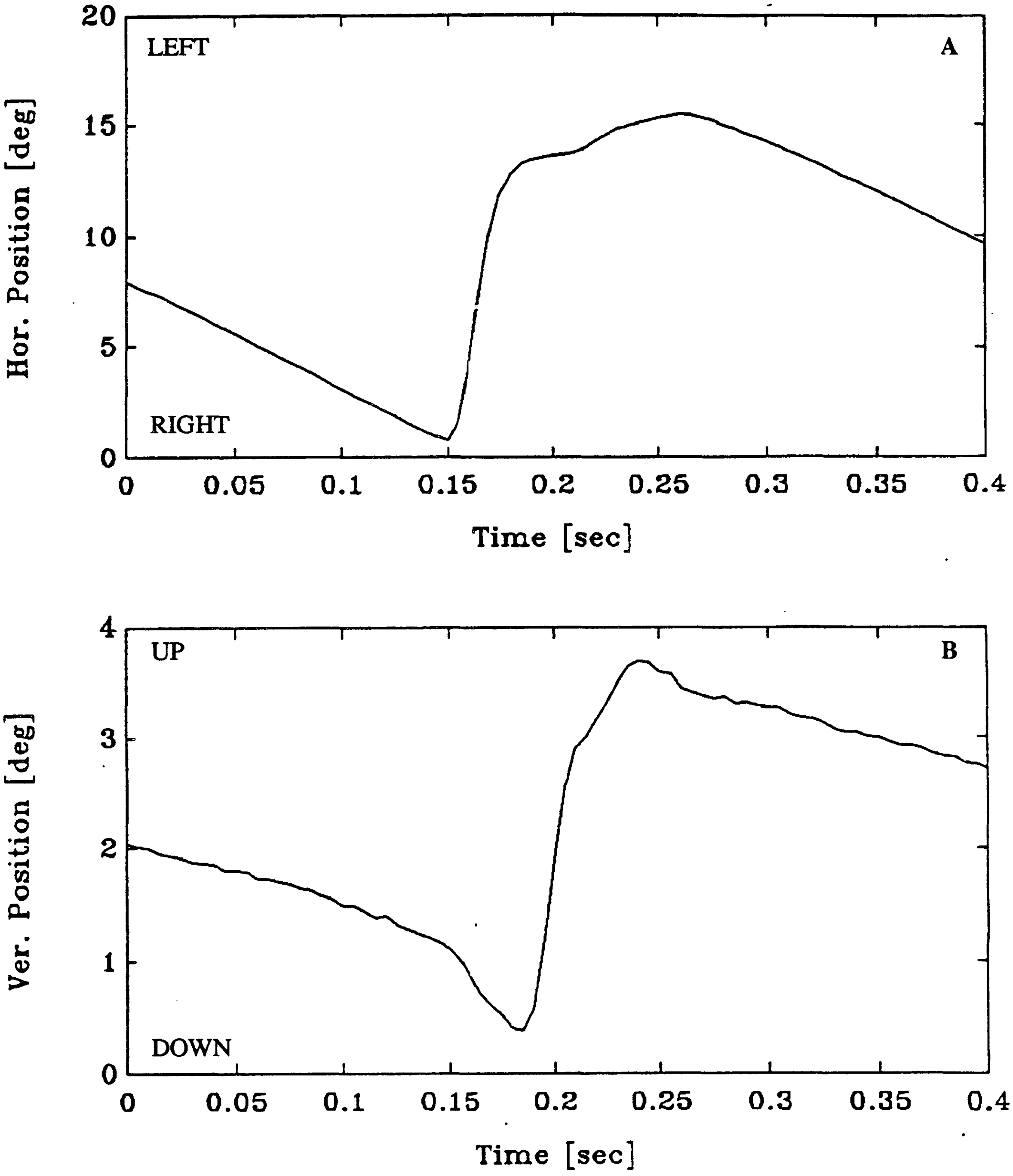


Figure 2.7

SLOW HORIZONTAL SACCADE (with Blink)

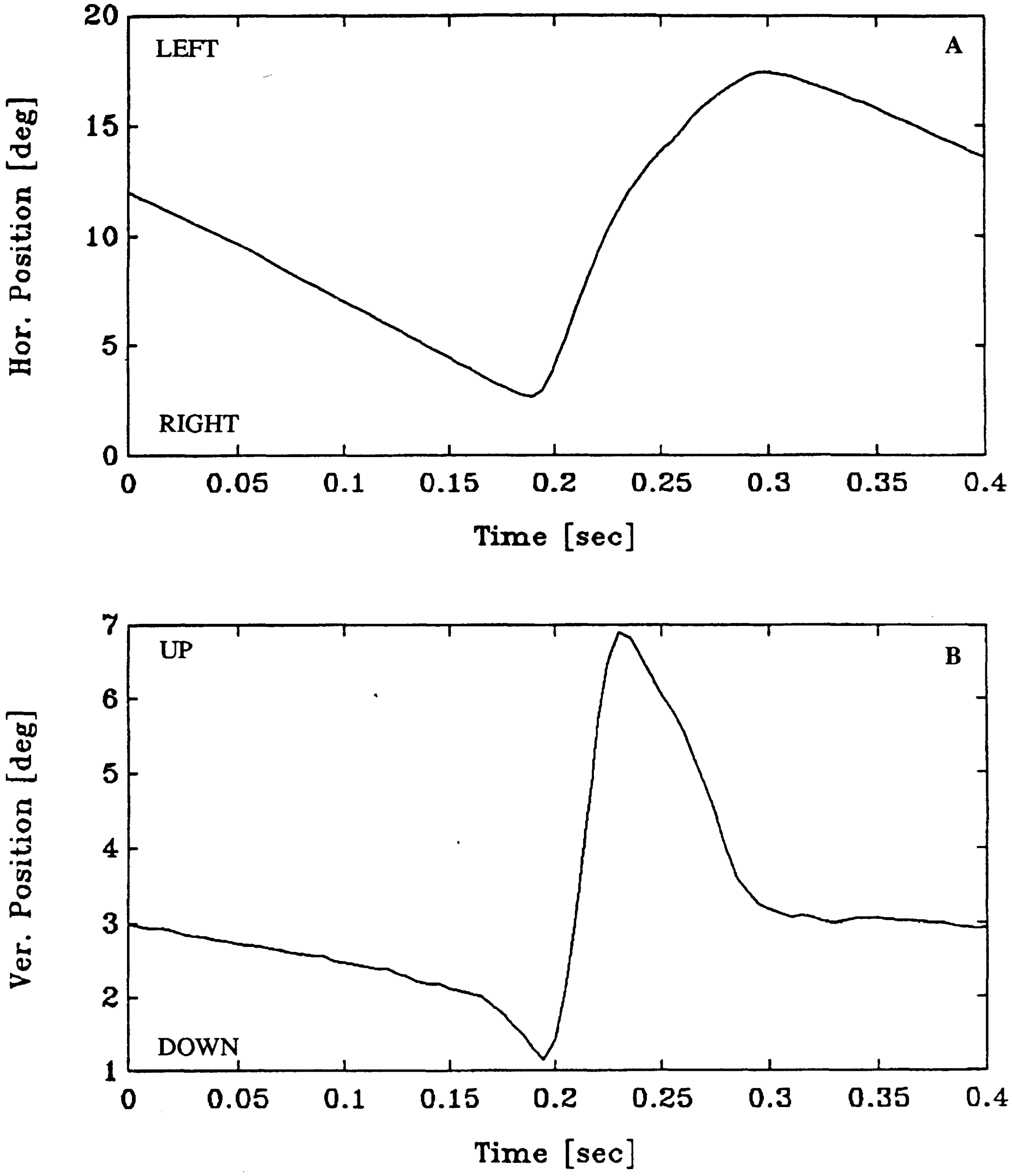


Figure 2.8

the identical algorithm more than once. The additional preprocessing time required to square and sum the components is much less than that required for event detection.

Therefore, a multi-dimensional approach to nystagmus analysis: 1) is unbiased with respect to direction. 2) finds beginnings and ends of events more easily. 3) is faster. For these reasons I have implemented an algorithm similar to Massoumnia's with a detection scheme based upon the total magnitude of eye acceleration. The programs which implement this algorithm are called DETECT1.M. (These programs are implemented using M-files in Matlab and are listed in Section 3 of Appendix B.).

The eye velocity is calculated as discussed in Section 2.1. The three orthogonal components of eye velocity are digitally differentiated to yield eye acceleration. Finite impulse response (FIR) filters are used to calculate both eye velocity and eye acceleration. The differentiation was implemented via a direct convolution of the filter coefficients with the input signal. A five element filter was used to calculate eye velocity and a nine element filter was used to calculate eye acceleration. The filter coefficients for the five element filter are:

$$B = [\tfrac{1}{8}\ \tfrac{1}{4}\ 0\ \tfrac{-1}{4}\ \tfrac{-1}{8}]\, f_s = [25\ \ 50\ \ 0\ \ {-50}\ \ {-25}],$$

where f_s is the sampling rate (200 Hz). The filter coefficients for the nine element filter are:

$$B = [\tfrac{1}{8}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \tfrac{-1}{8}]\, f_s = [12.5\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ {-12.5}].$$

The frequency response of each of these filters is shown compared to an ideal differentiator in Figures 2.9a and Figures 2.9c. The actual frequency response can be modelled as an ideal differentiator in series with a low-pass filter. The equivalent low-pass filters are shown in Figures 2.9b and 2.9d.

The magnitude of the acceleration is calculated by finding the square root of the sum of the squares:

$$|\vec{\alpha}| = \sqrt{\alpha_x^2 + \alpha_y^2 + \alpha_z^2}.$$

FREQUENCY RESPONSE OF DIFFERENTIATORS

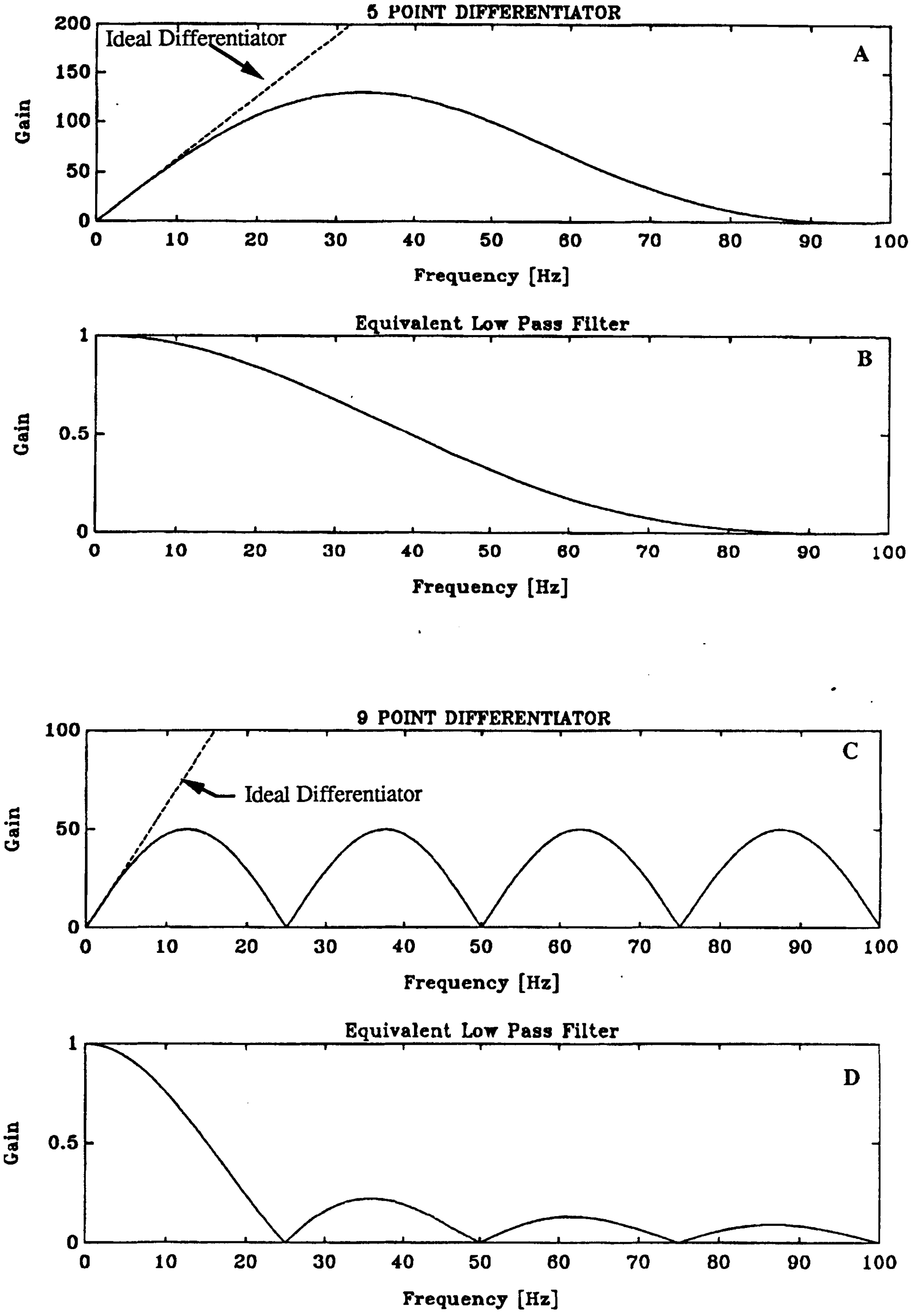


FIGURE 2.9

If the magnitude of the eye acceleration exceeds a threshold set by the user (The acceleration threshold was typically set between 800 and 2000 deg/s^2) an "event" is detected. The end of the saccade is the point where the magnitude of the eye acceleration is less than a second user set threshold (The end threshold was typically set between 500 and 1000 deg/s^2) for at least three points in a row.

The median value of eye velocity for the five points preceding the start of the saccade is determined. This value is substituted for the true eye velocity during the saccade. This method was selected over an interpolation approach because the end of the saccade is much harder to detect the the beginning. The slow phase velocity is resampled at 20 Hz and sent to the second stage of processing.

2.2.2.2 Velocity Detection (2nd Pass)

The second pass at the data is performed by a set of programs called DETECT2.M. (These programs are implemented using M-files in Matlab and are listed in Section 3 of Appendix B.) First each of the three components of eye velocity are divided into 10 signals each sample at 2 Hz. This is accomplished by choosing every tenth point with a starting point of one through ten, respectively. Each of these thirty signals (3 components of velocity times 10 signals each) is fit by a least squares model of the form:

$$y[n+1] = a\, y[n] + b\, u[n] + w[n+1] \quad ,$$

where,

y[n] is the slow phase eye velocity (yaw, pitch, or roll),
u[n] is the angular acceleration stimulus,
w[n+1] is white noise,
and a and b are the model parameters determined by the model fit.

(See Ljung, 1987 for an exhaustive discussion of this type of parameter estimation.)

The residuals for each of the three components are summed and squared. If the magnitude of the residuals exceeds a user set threshold (The velocity threshold was typically between 20 and 50 deg/s) each of the components is replaced by the value predicted by the best fit model.

This very simple ad hoc algorithm worked very well at removing nearly all of the saccades which were missed by the acceleration detection algorithm. As mentioned previously, the acceleration algorithm is not very good at detecting the end of the saccades. The end sometimes is declared to be a point or two before the end an intelligent user would define. This simple algorithm is also very good at finding and removing these points.

2.2.3 Additional Processing (Low Pass Filtering)

After all of the saccades are removed, the slow phase eye velocity is passed through a low pass filter. The FIR filter is implemented using direct convolution such that there is no phase shift at any frequency (zero-phase filter). The filter has 45 coefficients and was designed to have a frequency cut-off of 2 Hz. Figure 2.10 shows the frequency response of the filter. The filter coefficients were calculated using the Remez implementation of the Parks-McClellan algorithm. (See Oppenheim and Schafer, 1975 for further discussion.)

2.2.4 Conclusion

Figures 2.11, 2.12, 2.13 show the slow phase eye velocity during the various stages of processing for a typical run. Figure 2.11 shows the yaw response, figure 2.12 shows the pitch response, and Figure 2.13 shows the roll response. Figures 2.14, 2.15, and 2.16 show a potion of the same data with a dramatically different time scale.

The overall performance of the algorithm is acceptable for this particular data set. This performance is similar to that obtained for all of the coil data analyzed. Therefore, this algorithm appears adequate for analysis of the coil data discussed in Chapter 3.

A few thoughts for future work became very evident while working on this portion of the data analysis. I feel it is inaccurate to interpolate over the course of a saccade. Instead, I feel that further analysis should simply ignore the regions of the data stream which are detected as events.

I think that an iterative least squares approach should replace the least square parameter estimation discussed in Section 2.2.2.2. The iterative least square approach would require more computation, but it would increase the algorithm accuracy.

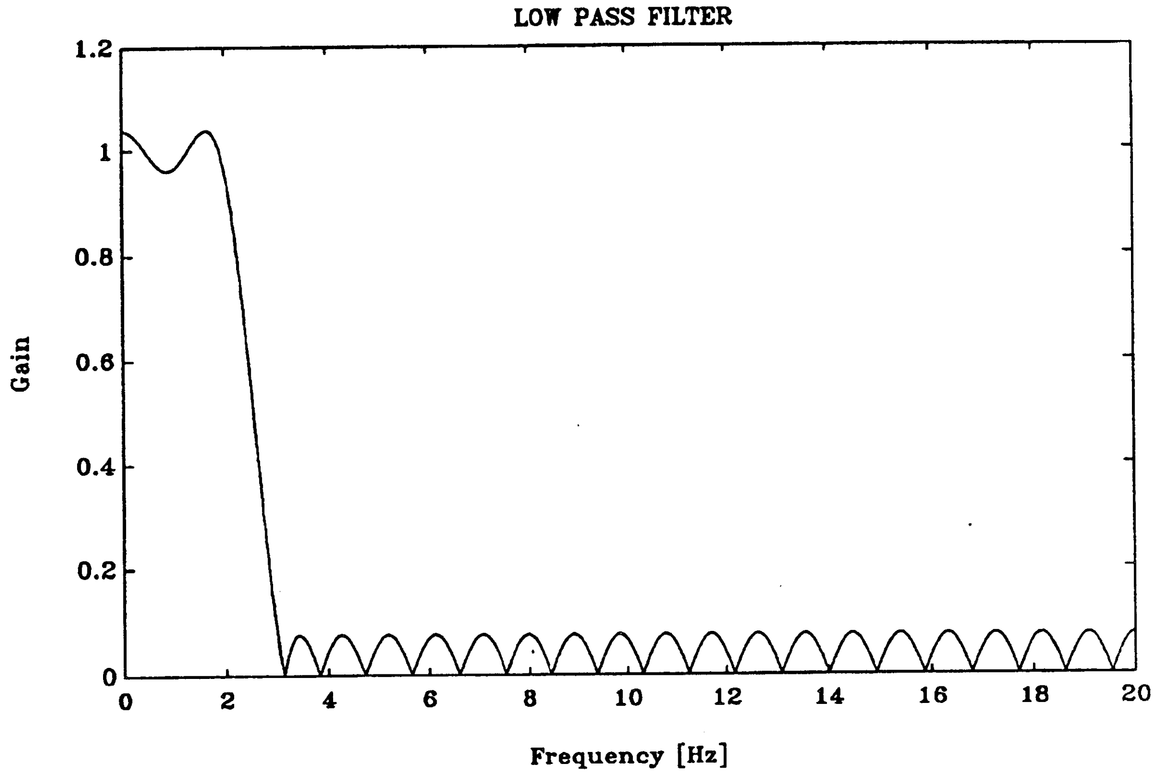

LOW PASS FILTER
1.2
1
0.8
0.6
0.4
0.2
0
Gain
0
2
4
6
8
10
12
14
16
18
20
Frequency [Hz]

Figure 2.10

HORIZONTAL SPV PROCESSING (Large Time Scale)

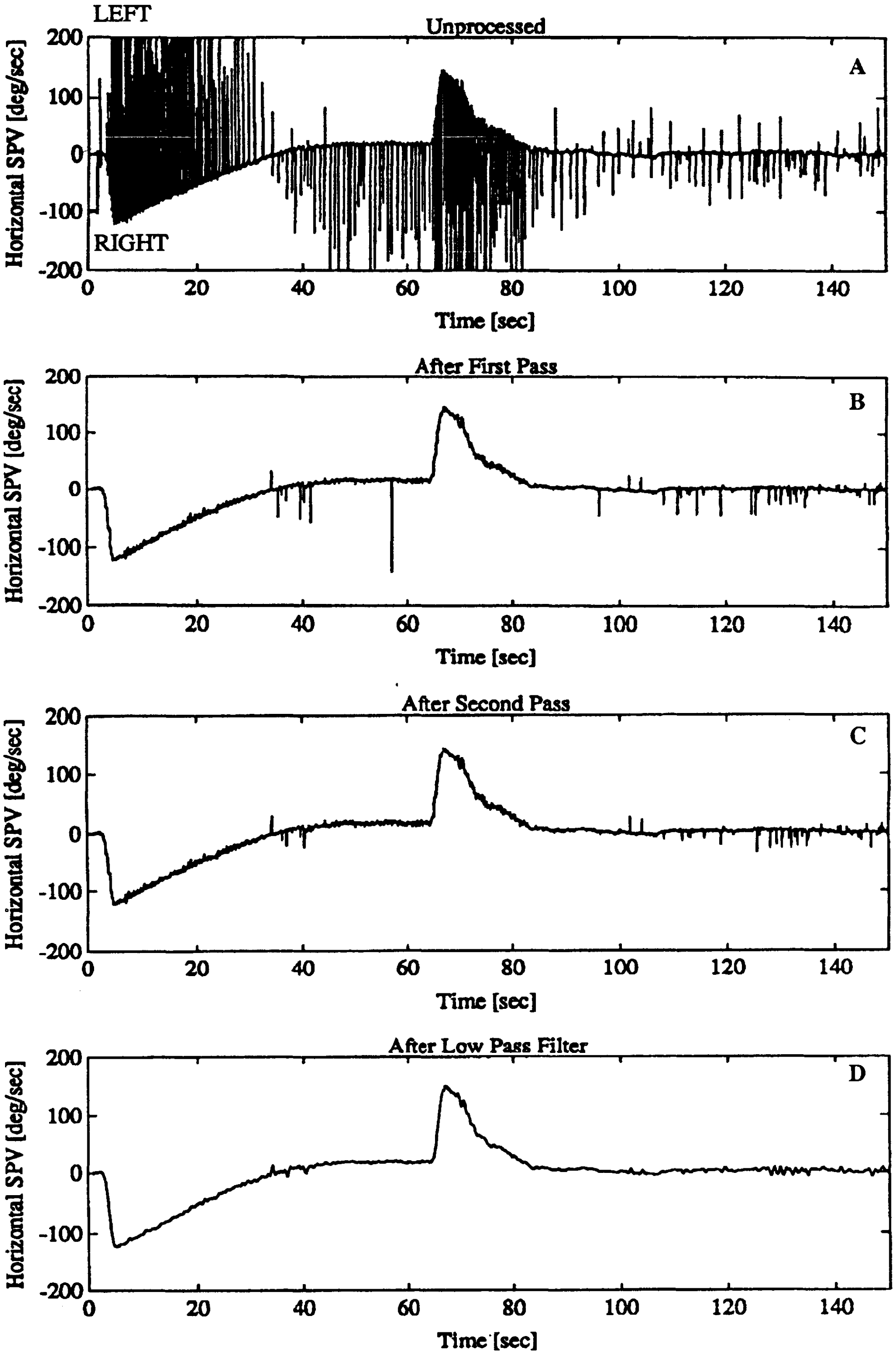


Figure 2.11

VERTICAL SPV PROCESSING (Large Time Scale)

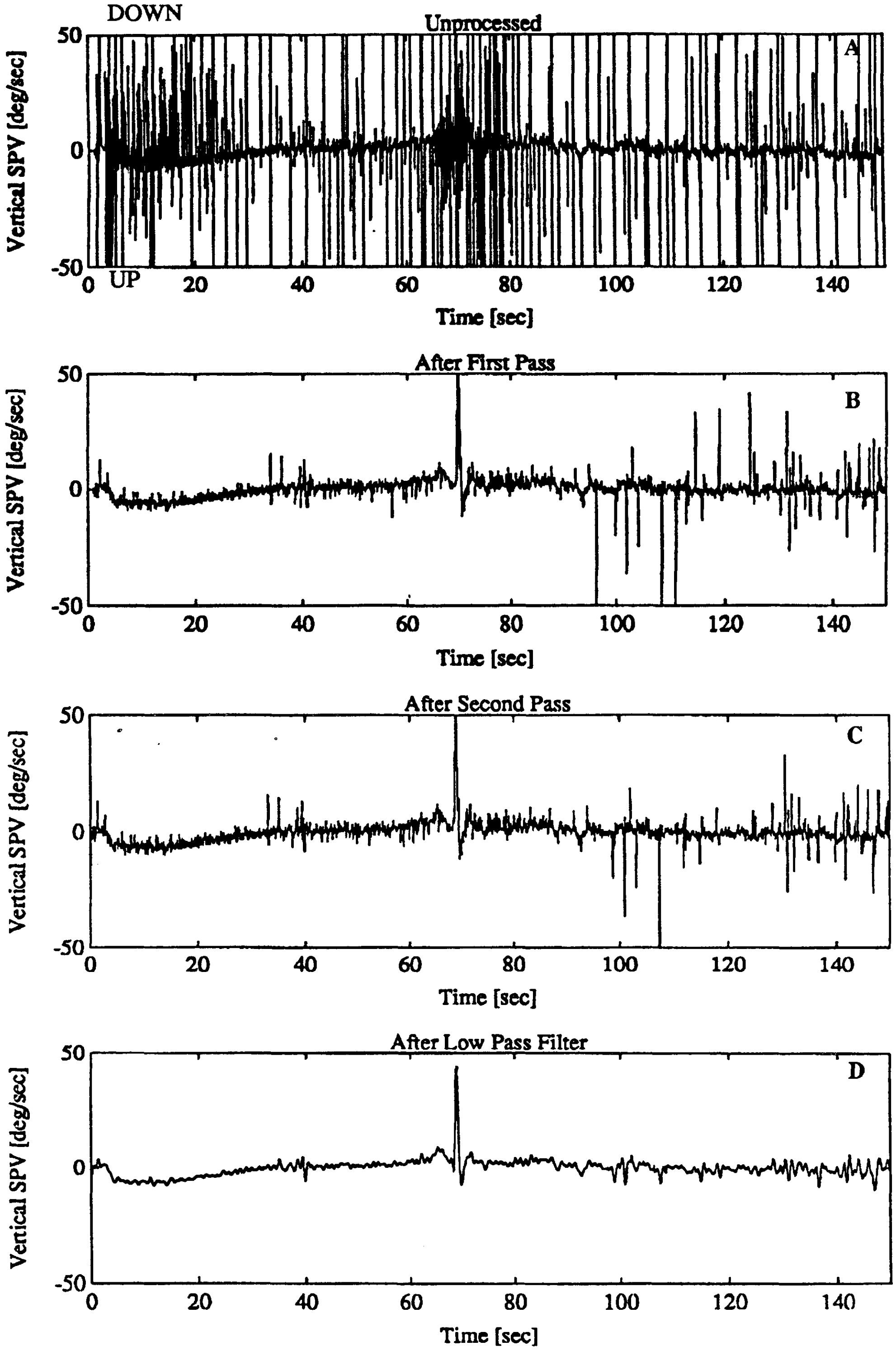


Figure 2.12

TORSIONAL SPV PROCESSING (Large Time Scale)

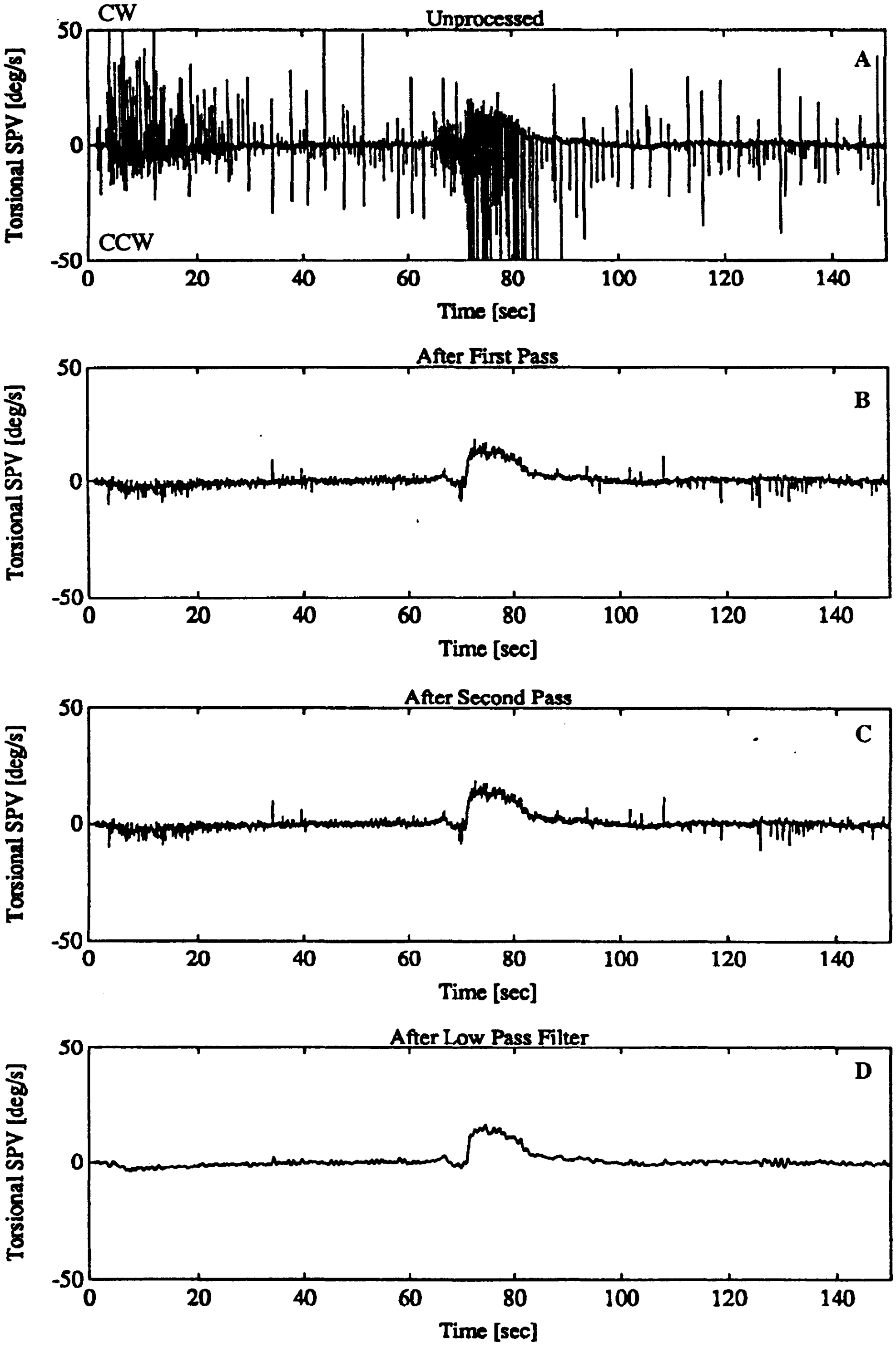


Figure 2.13

HORIZONTAL SPV PROCESSING (Small Time Scale)

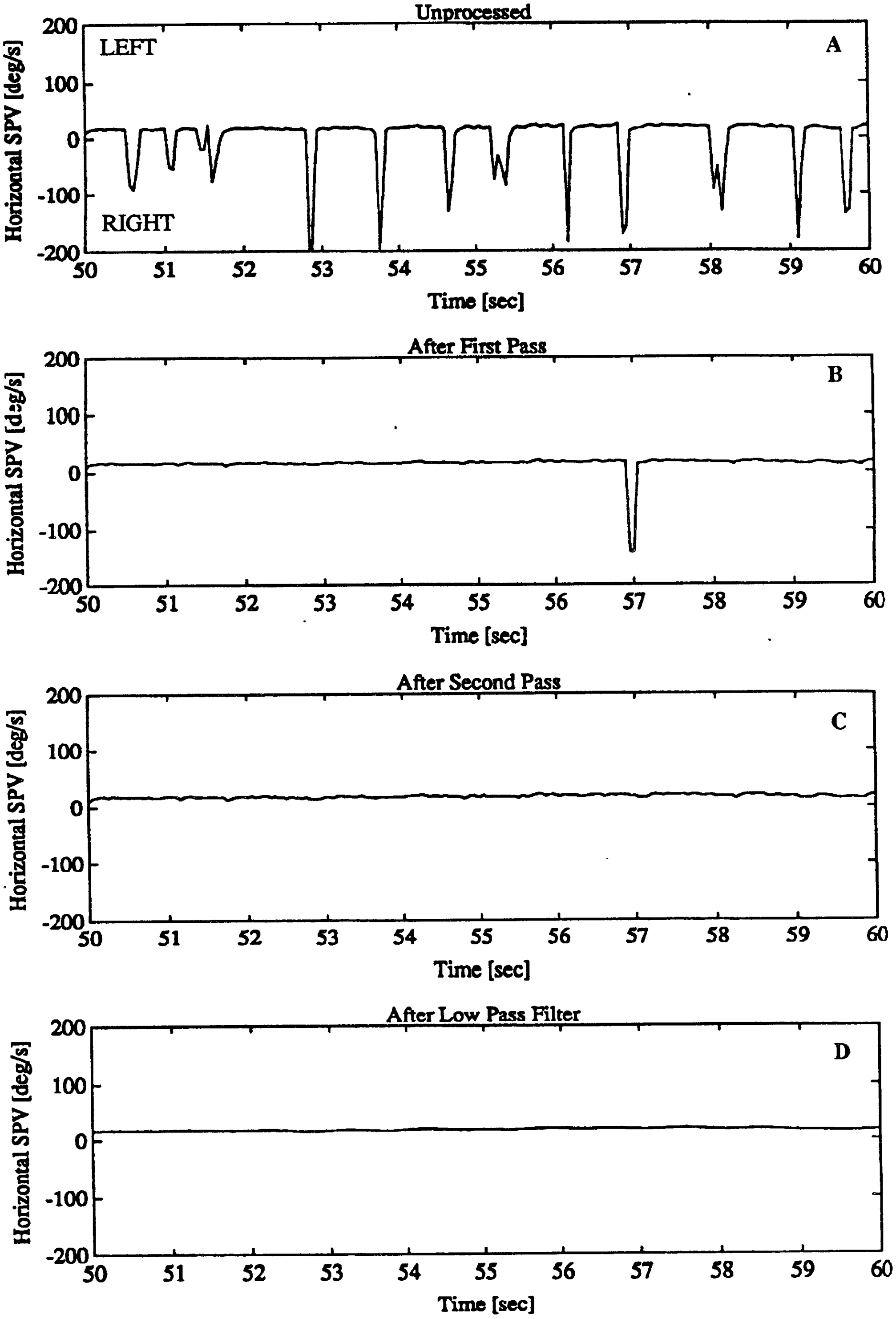


Figure 2.14

VERTICAL SPV PROCESSING (Small Time Scale)

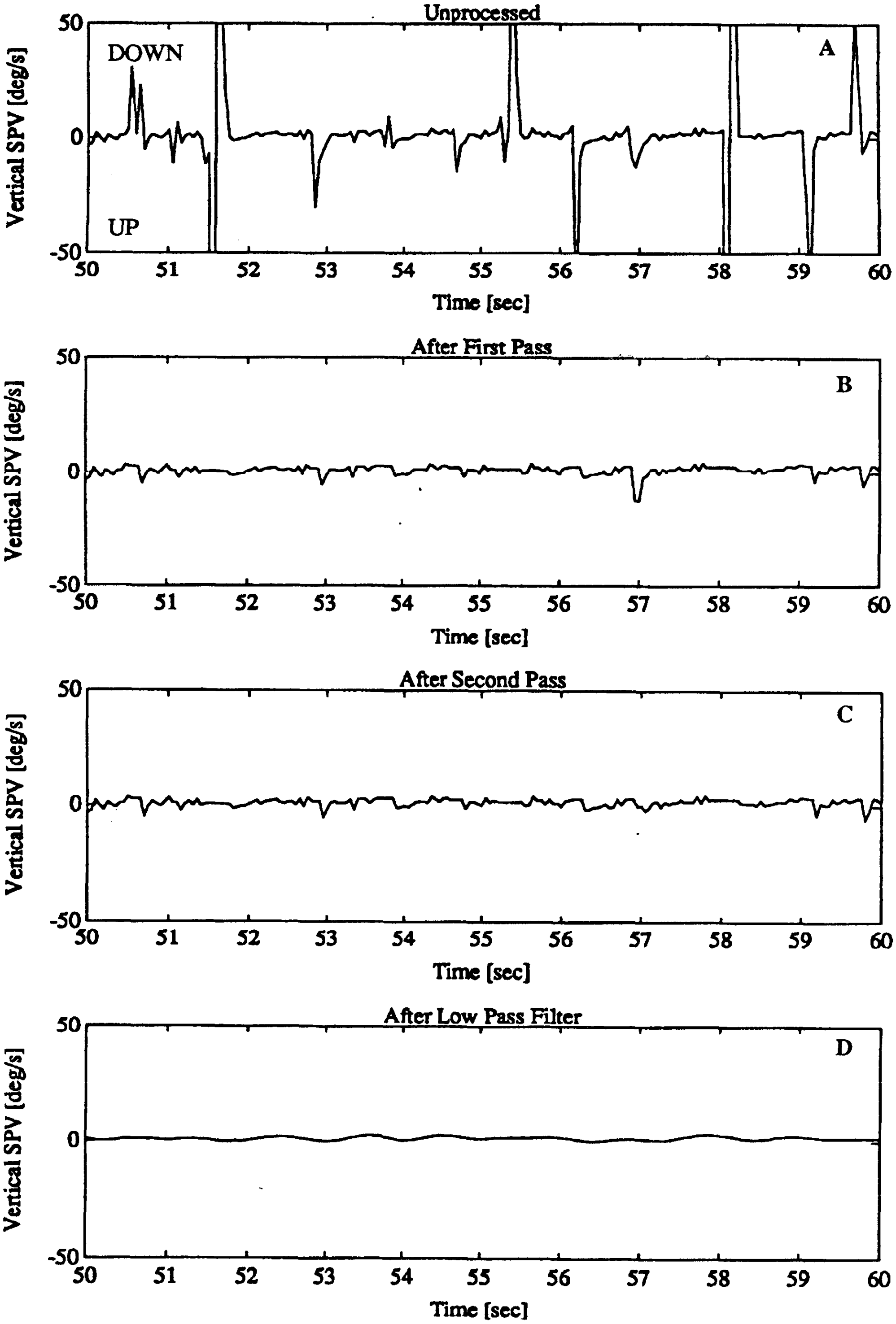


Figure 2.15

TORSIONAL SPV PROCESSING (Small Time Scale)

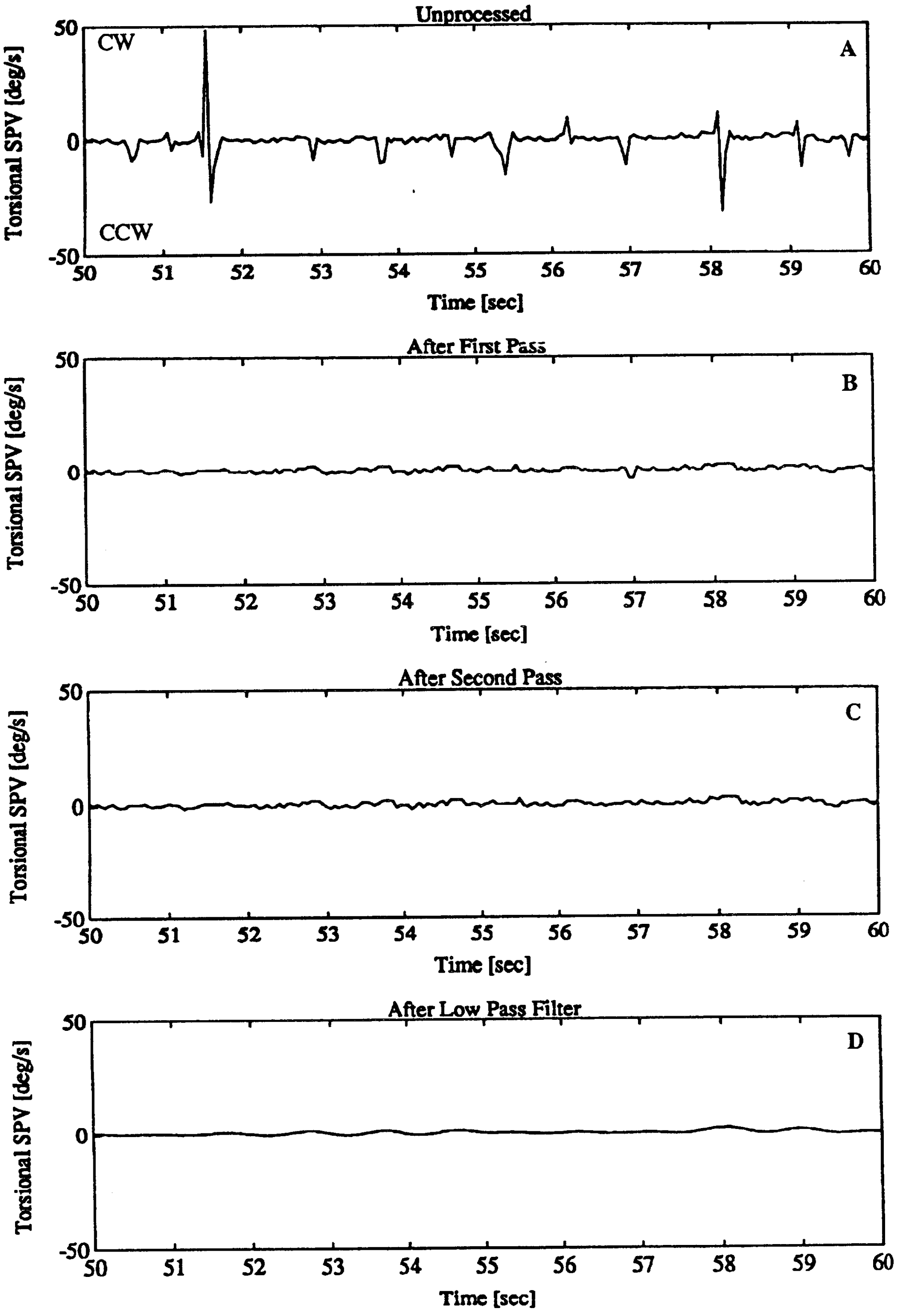


Figure 2.16

The three dimensional approach discussed in this chapter is an improvement over one dimensional analysis when more than one signal is available. It is slightly more accurate than the one dimensional approach and the amount of computation is dramatically reduced compared to analyzing two or three components separately.

2.3 Parameter Estimation

Previous research (Oman and Young, 1970) has determined that the transfer function representing the rotational response to angular velocity has the form:

$$\frac{spv(s)}{\omega(s)} = \frac{K\, s^2}{(\tau_l\, s + 1)\,(\tau_a\, s + 1)}\ .$$

The time response of this system to a step in angular velocity can be shown to have the form:

$$spv(t) = A\, e^{-t/\tau_l} + B\, e^{-t/\tau_a}\ .$$

In order to solve for the unknown parameters (A, B, τ_l, and τ_a), we need to solve a nonlinear estimation problem. This must be solved iteratively.

A number of methods have been developed to perform this type of nonlinear estimation. Almost all of these methods calculate a parameter which represents "goodness of fit" and attempt to minimize this parameter. Generally the merit function used is the sum of squared errors. Minimizing this χ^2 merit function determines the best fit in the least squares sense. Since our model is nonlinear, the minimization process proceeds until the χ^2 merit function is effectively minimized.

One algorithm which performs this type of iterative fit is the Levenberg-Marquardt method, also known as the Marquardt algorithm (Press et al., 1986; Marquardt, 1963; Levenberg, 1944). This algorithm has become accepted as the standard method used to perform least squares nonlinear parameter estimation. This algorithm has been implemented as part of an interactive data analysis package called IGOR (copyright WaveMetrics, 1989).

Figure 2.17 shows the curvefit obtained to synthetic data of the form:

$$y(t) = 100\, e^{-t/15} - 20\, e^{-t/80} + n(t) \,,$$

where n(t) is gaussian noise with a variance of 1. The algorithm calculates a curvefit of:

$$\hat{y}(t) = 99.67\, e^{-t/14.94} - 19.66\, e^{-t/81.2} \,.$$

This performance is acceptable.

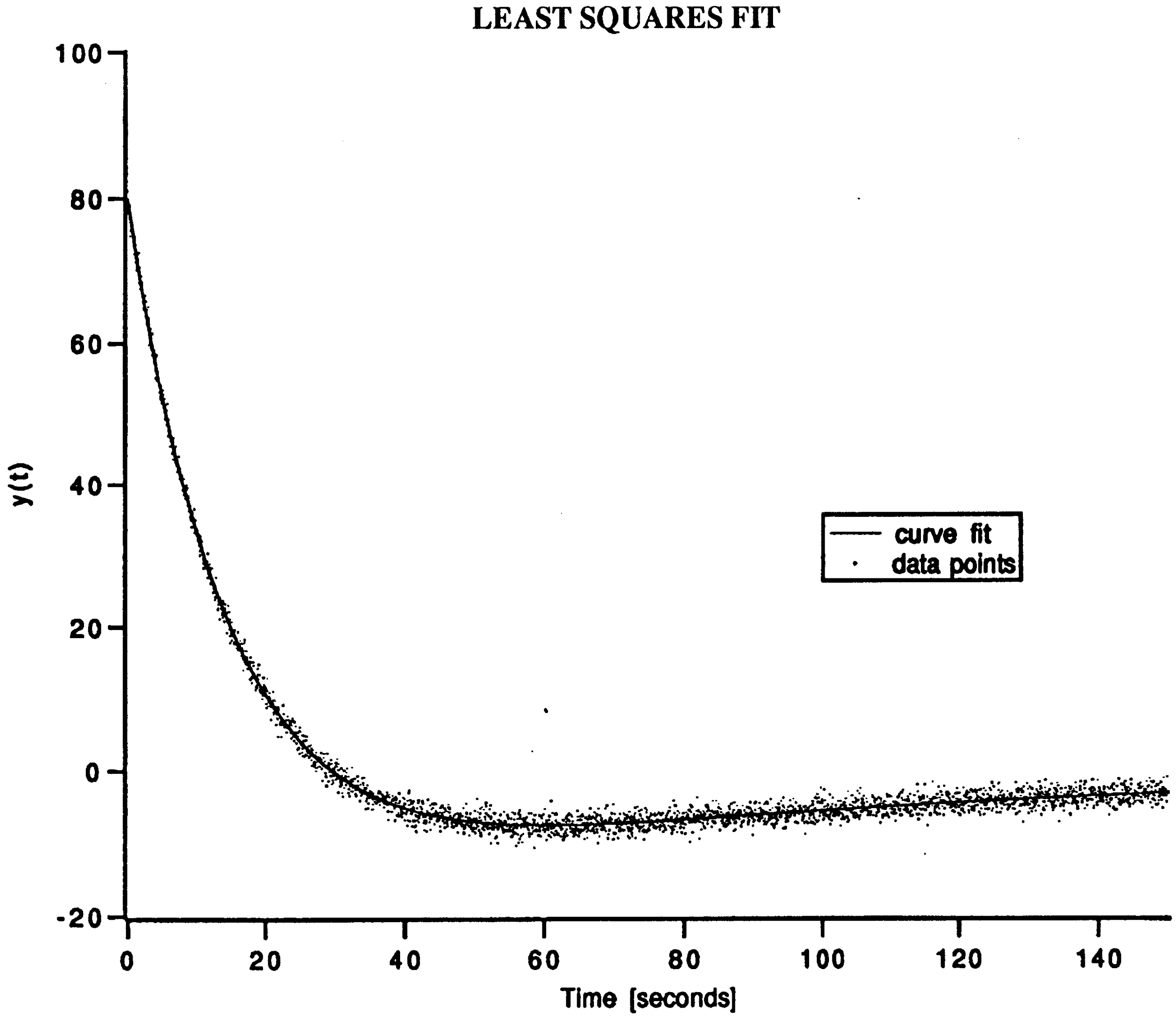
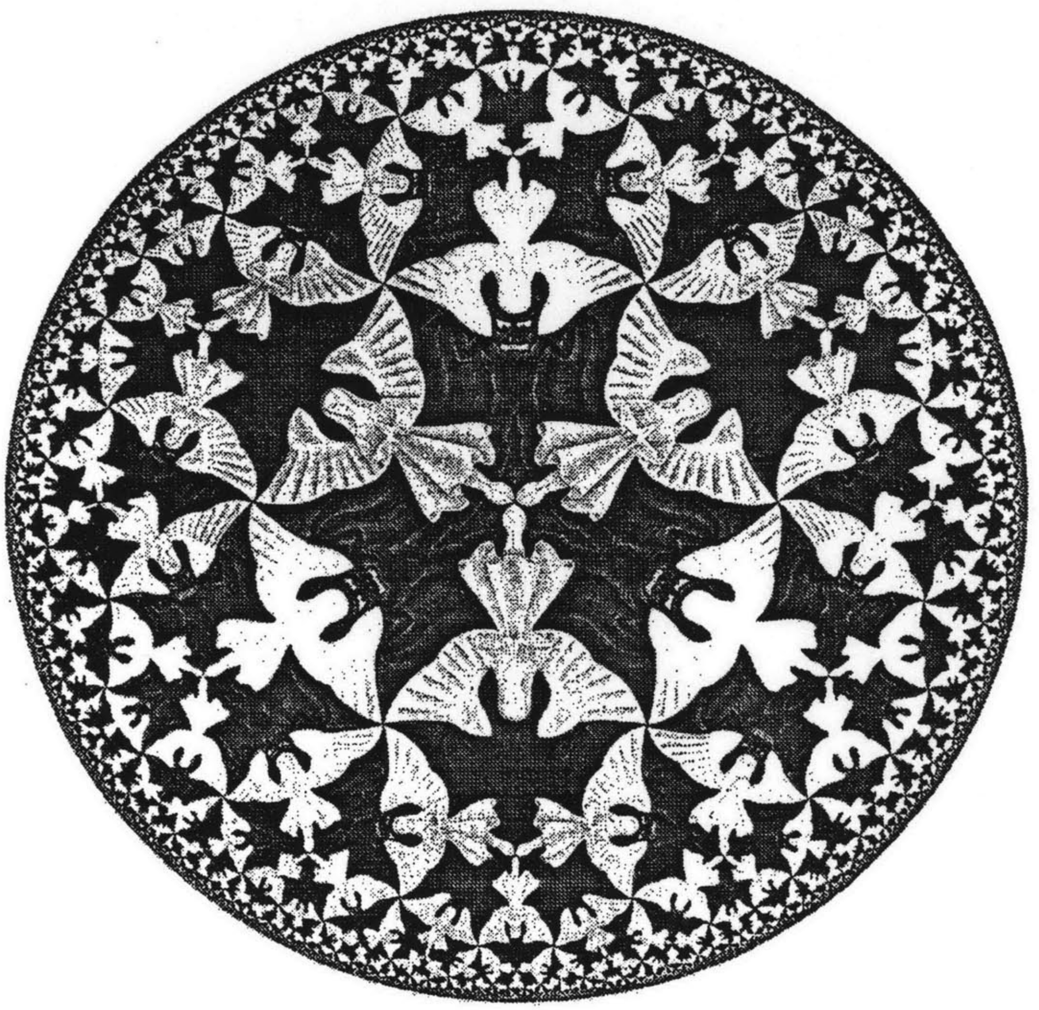

LEAST SQUARES FIT
100
80
60
40
20
0
-20
y(t)
0
20
40
60
80
100
120
140
Time [seconds]
curve fit
data points

Figure 2.17

III

VOR DURING CENTRIFUGATION



M. C. Escher

"There are three kinds of lies: lies, damned lies, and statistics."
Benjamin Disraeli

"Some circumstantial evidence is very strong, as when you find a trout in the milk."
Henry David Thoreau

CHAPTER THREE
VOR DURING CENTRIFUGATION

3.0 Introduction

In this chapter I present a set of experiments which investigate spatial orientation in the squirrel monkey. Specifically, I investigate how squirrel monkeys resolve gravito-inertial force into components representing "down" and linear motion. The hypothesis being tested claims that gravito-inertial force as sensed by the otoliths is resolved into two components. One component represents linear acceleration, while the second component represents gravity. This hypothesis will be supported by the results that follow and by a number of studies from the literature.

To investigate this hypothesis, three types of experiments were performed on each of six squirrel monkeys. Two of these tests were used as control studies to investigate the effects of gravity on the VOR: static tilt tests and "dumping" tests. The central test used centrifugation to introduce a gravito-inertial force which did not align with gravity.

The static tilt test is a very simple control designed to investigate spontaneous nystagmus induced by the force of gravity. With the lights off, the monkeys were manually positioned with respect to gravity and the eye movements were recorded.

The dumping test is a relatively simple test designed to investigate the effects of gravity on post rotary VOR. The monkeys were spun at a constant velocity till the responses were nearly extinguished (80 seconds) and then quickly decelerated to a stop. As quickly after stopping as possible, while the post rotary VOR was still strong, the monkeys were tilted with respect to gravity, and the VOR response recorded.

The central experiment of this thesis was designed to investigate how the CNS would resolve a gravito-inertial force (GIF) composed of gravity and centrifugal force into components representing gravity and linear acceleration. During centrifugation, the monkeys were always kept upright with respect to gravity. In this fixed orientation, they were spun about a non-centric vertical axis such that the centripetal acceleration was perpendicular to gravity. The orientation of the monkey with respect to the centripetal acceleration was varied for different trials.

This chapter is organized around these three experiments. In Section 3.1 a number of background studies from the literature are discussed. In Section 3.2, I present the methods used to perform the centrifugation tests, followed by the results in Section 3.3. Section 3.4 presents the control studies. Finally, sections 3.5 and 3.6, respectively, contain a discussion of the results and a review of the conclusions.

3.1 Background

The sense of spatial orientation is dependent upon the interaction of many sensory systems, including the senses of vision, proprioception, touch, audition, as well as the vestibular labyrinth. The dominant role.for most dynamic tasks is played by the labyrinth. This set of sense organs dominate the experimental studies conducted in this thesis almost to the exclusion of the other senses. Therefore a brief word about the importance of the other senses is necessary.

Vision dominates the remaining sensory systems and has therefore been studied most copiously. Vision has long been known to induce strong sensations of motion. Most of us in our own experience can recall the compelling sense of motion induced by the movement of a large visual field such as a stream flowing beneath a bridge or the movement of a train viewed through the window of a stationary train. The conditions necessary for this illusion have been thoroughly studied and are fairly well understood (Dichgans and Brandt, 1974; Held et al. 1975; Young, 1981).

The role of tactile cues was demonstrated when a stationary subject touching a rotating surface experienced a sense of self motion (Brandt et al., 1977). While less compelling than the sense of motion induced by visual stimulation, the sensation was strong enough to induce compensatory eye movements in some of the subjects.

The role of proprioceptive cues was demonstrated in a clever experiment performed by Bles (1979). He designed a simple room that appeared to allow the subjects to walk around in a small circle inside a drum. Actually, the subjects were stationary and the floor moved beneath them. Eye movements and sensations similar to those obtained during true circular walking were achieved in the absence of true motion.

Not surprisingly, auditory cues have also been shown to be important to spatial orientation. At least two studies (Lackner, 1977; Buizza et al., 1979) have demonstrated effects of auditory cues upon eye movements and upon motion sensation.

Despite the importance of each of these cues, a number of studies have indicated the important, if not dominant role, played by the vestibular system. Many of these studies have used subjects with partial or complete loss of vestibular function (Guedry and Harris, 1963; Graybiel and Clark, 1965; and Miller et al., 1966). Because of the important role played by the vestibular apparatus, the remainder of this chapter will emphasize the vestibular component of spatial orientation.

Scientists began to gain an understanding of the vestibular system in the mid 19th century when it was found that destruction of the vestibular labyrinth of an animal led to turning in one direction (Flourens, 1842). A later study suggested that the semicircular canals sense rotation of the head (Goltz, 1870).

Later in the 19th century, the physiology of the semicircular canals began to be understood. Three independent studies (Mach, 1873; Breuer, 1874; and Crum Brown, 1874) suggested that the parameter sensed during rotation was angular acceleration and that the vestibular system was the transducing organ.

In the early 20th century, Steinhausen demonstrated the presence of a cupula in each of the semicircular canals. After studying the characteristics of the vestibular responses, he developed a second order mechanical model (later developed as the "torsion pendulum" model) describing the motion of the cupula in response to angular acceleration (Steinhausen, 1933). Later, investigators (van Egmond et al., 1949; Hallpike and Hood, 1953) applied the model to human responses, including the reflexive eye movements called the vestibulo-ocular reflex (VOR).

The response of peripheral afferents of the semicircular canals were first recorded in an isolated preparation from a thornback ray (Lowenstein and Sand, 1940). This study and a later study (Groen, et al., 1952) qualitatively confirmed the theoretical predictions of the torsion pendulum model.

Later, the peripheral afferents of the semicircular canal were recorded in the intact squirrel monkey (Goldberg and Fernandez, 1971). These studies showed that intact afferents had a steady resting discharge, that the change in firing rate was dependent upon the direction of the angular acceleration, and that the dominant time constant was approximately six seconds. In addition, these studies showed that first order afferents could not be represented by a simple torsion pendulum model since rate sensitivity and neural adaptation were also observed.

A similar set of experiments investigated the afference from squirrel monkey otolith organs (Fernandez and Goldberg, 1976). Similar to the analogous study of the semicircular canal, these investigators observed that intact otolith afferents have a steady resting discharge, that the change in firing rate depended upon the direction of linear acceleration, and that there were two distinct afferent populations, one with regular firing rates and the other with irregular firing rates.

As mentioned earlier, angular accelerations are known to elicit a reflexive movement of the eye called the vestibulo-ocular reflex (VOR). This reflex helps stabilize the visual field on the retina when the head is rotating. A neural signal, transduced by the semicircular canals, drives the ocular motor neurons to move the eyes within the skull. The motion of the eyes in the skull is such that they tend to be inertially stabilized. This helps to compensate for motion of the head. Because of its consistency, the VOR is routinely used by clinicians to evaluate the vestibular system.

The most direct pathway of the VOR has been known for many years and involves a simple three neuron arc. The three cell bodies are found in the vestibular ganglion (Scarpa's ganglion), the vestibular nuclei, and the extra-ocular motor nuclei. There are a number of alternate pathways from the vestibular system to the ocular motor neurons. Among others, these include the cerebellum and the reticular formation.

Measurements obtained during yaw, pitch, and roll have shown that the responses to rotational stimuli depend upon the axis of rotation. One study (Melvill Jones et. al., 1964) measured responses to each of these rotational stimuli in normal human subjects and found that the time constants of nystagmus decay for pitch (6.6 s) and roll (4.0 s) were much less than that obtained for yaw (15.6 s). Another study (Guedry and Benson, 1970) found a similar difference in the time constants for constant velocity yaw and pitch rotations. Some studies (Matsuo et al., 1979; Matsuo and Cohen, 1984) have even indicated that the time constant for forward pitch (15 s) was greater than that for backward pitch (8 s). [The previous studies indicated no such asymmetry. The methods used in these three studies were remarkably similar. The major difference is that human subjects were investigated in the first two studies while monkeys were studied by Matsuo et al.]

Time-varying linear accelerations also elicit a reflexive response. This reflex, called the linear VOR (LVOR), was first observed in humans in tests with regular oscillations on a parallel swing (Jongkees, 1961). The observed eye movements were smooth and regular. A later study showed that the LVOR could have the nystagmic characteristic observed in the VOR (McCabe, 1964). Even though the LVOR was observed in humans over twenty years ago, it is not well understood nor universally accepted (Hain, 1986). A probable explanation for the confusion is that LVOR, as measured in the dark, is weak and irregular (Kitchen, 1982).

Researchers claim that, for clinically normal subjects, static tilt within the gravitational environment of earth does not induce a nystagmus dependent upon the position of the subject (Stahl, 1958; Jongkee, 1960). But Bergstedt (1961) has asserted that 27% of a sample of otherwise clinically normal subjects exhibited such positional nystagmus. Benson (1974) later examined these data and determined that the mean slow phase velocity was less than one degree per second. It can safely be claimed that positional nystagmus is absent in most normal humans, and that, when present, it is very weak.

While fixed and constant 1-g linear acceleration induces, at most, very weak nystagmus, it can have very dramatic effects on rotatory nystagmus and on optokinetic nystagmus.

It has been claimed (Markaryan's, 1969) that Khilov (1929, 1936) was the first to show that the duration of post-rotatory nystagmus was dependent upon the orientation of the head with respect to gravity. Similarly, it was shown that postrotational nystagmus had a smaller horizontal slow phase component when the axis of rotation was horizontal than when vertical (Correia and Guedry, 1964; Benson and Bodin, 1966a).

The same effect was observed (Benson and Bodin, 1966b) when the subject was rotated about a vertical axis, and shortly after stopping, moved to one of the four orthogonal horizontal positions (nose down - ND, nose up - NU, left ear down - LED, right ear down - RED).

At about the same time, it was pointed out that during the immediate post rotatory period graviceptors indicate constant orientation with respect to gravity while the semicircular canals indicate rotation. This sensory information is in conflict unless the apparent rotation occurs about an axis parallel to gravity (Guedry, 1966a, 1965b). If this sensory conflict is the cause of a response decrease, then the amount of the decline should be modulated by the intensity of the competing otolith signal. Benson (1966) later verified that the fastest response decay occurred when the the subjects were tilted 90 degrees after rotation. This orientation maximizes the competition between the semicircular canals and the otoliths by placing the axis of rotation indicated by the semicircular canals perpendicular to gravity as sensed by the otolith organs.

Perceptual studies also confirmed the effect of constant 1-g linear acceleration on the sense of rotation. These studies showed that the sensation of rotation was even more drastically affected than the nystagmic response. Some subjects reported no rotational sensations following constant velocity rotation about a horizontal axis (Guedry, 1965a; Benson and Bodin, 1966a).

Raphan et al. (1981) later verified with squirrel monkeys that the time course of post-rotatory eye movements was drastically reduced by tilt with respect to gravity. Furthermore, they demonstrated the presence of a strong vertical nystagmus which decayed with the horizontal nystagmus.

Harris (1987), who did experiments with cats, also found a strong vertical response that often built up following off-vertical rotations. When carefully measured, the axis of the post-rotational eye movements aligned with gravity.

Similar effects were found through observation of visual reflexes. It has long been known (Ter Braak, 1937) that motion of an entire visual field about an axis parallel to gravity in an upright subject induces horizontal eye movements called optokinetic nystagmus (OKN). When the lights are suddenly extinguished, horizontal optokinetic afternystagmus (OKAN) is often observed (Krieger and Bender, 1956). These responses have been quantitatively studied (Cohen et al., 1977) and modeled (Raphan et al., 1977). The OKAN response is generally much stronger in monkeys than in humans.

When a subject is tilted with respect to gravity and the visual stimulus continues about the yaw axis of the subject, horizontal OKN (in a head fixed reference frame) is induced. However, when the lights are extinguished, a strong vertical component of nystagmus is observed(in the monkey, Raphan and Cohen, 1983, 1988; in the cat, Harris, 1987). The direction of the vertical nystagmus was such that the axis of the eye movements tended to align with gravity.

Eye movements of humans have also been recorded during constant velocity rotation about an earth horizontal axis (also known as "barbecue spit" rotation). Benson and Bodin (1966) used rotation rates between 10 and 60 degrees per second. They discovered that the steady state response demonstrated a “bias” component and a “sinusoidal” component. Using rotation rates of 60 and 180 degrees per second, Correia and Guedry (1966) observed a similar response. Wall (1987) also verified these studies. Furthermore, he correlated the decay time constant of the horizontal nystagmus with the gain of the sinusoidal component. This correlation showed that subject who demonstrated long time constants showed less modulation than those who had short time constants.

Similar experiments with monkeys (Raphan et al., 1981) have indicated that monkeys demonstrate responses which are similar to those recorded in humans. Both bias and sinusoidal components are observable, but, unlike humans, the bias component tends to dominate the response.

Experiments with cats (Harris, 1987) also a show a two component response. The response is similar to that observed in monkeys with the bias component larger than the modulation component.

The eye movements of human subjects have also been studied using centrifugation to provide a change in gravito-inertial force (Benson, 1962; Lansberg et al., 1965; Crampton, 1966; Steer, 1967). Lansberg (1965) showed that the horizontal nystagmus generated by vestibular stimulation in an upright subject depended upon the orientation of the subject with respect to the centripetal acceleration. A later study (Crampton, 1966) confirmed these results and reported the presence of vertical nystagmus during centrifugation. A careful review of these data indicates that the vertical component of the eye movement tends to align the axis of the eye motion with the total gravito-inertial force vector. (Note: Lansberg's data indicate a slow phase vertical velocity which sometimes has a direction opposite that required to align the eye movement with gravity.) Using counter-rotation to eliminate angular acceration stimuli during centrifugation, Steer (1967) showed the presence of bias and modulation components similar to those observed during "barbecue spit" rotation.

These data were later analyzed by Young (1967) who found a component of the horizontal nystagmus which could be correlated with the linear acceleration resulting from centrifugation. This component was found by assuming that the differences in the responses measured with different subject orientations were solely attributable to the centripetal acceleration. Then, by subtracting the responses obtained in one orientation from the other, an estimate of the linear component was found.

The slow phase velocity of linear nystagmus rose rapidly with increasing linear (centripetal) acceleration, but then decayed with a time constant of at least 20 seconds during the constant velocity phase of the centrifugation.

Subjective orientation was measured under similar conditions (Graybiel and Brown, 1951). In this study, human subjects were requested to remotely orient a light rod parallel to their subjective horizontal during centrifugation. Typical finding from this study showed that the subjective indication of vertical lags behind the actual orientation of the gravito-inertial force and that the difference between the actual GIF and the subjective “down” decayed like an exponential with a time constant of approximately twenty to thirty seconds.

Since gravito-inertial force is a stimulus in the eye movement experiments and the orientation study, a joint interpretation of these centrifugation studies is illuminating. In the Graybiel and Brown study, the difference between the actual GIF and subjective orientation could represent that component of gravito-inertial force interpreted as acceleration. In Young's analysis of the Lansberg data, the linear eye movements shown are hypothesized to be related to the acceleration component of the gravito-inertial force. The similarity of the shape and time course of the two curves provides evidence that the CNS is resolving the sensory conflict by separating the gravito-inertial force into components representing linear acceleration and gravity. The component representing horizontal linear acceleration is indicated by the presence of horizontal linear nystagmus, while the component representing gravity is indicated by the subjective perception of "down".

These background studies also seem to indicate a number of consistent differences between human and animal responses. During "barbecue spit" rotation humans generally have a modulation component which is large and a bias component which is small compared to those responses obtained in monkeys or cats. Monkeys and cats also demonstrate a very robust axis transformation during tilted OKAN or during dumping. To my knowledge, no-one has ever reported these responses in humans, but, as previously discussed, weak vertical eye movements have been reported during centrifugation.

3.2 Methods

3.2.1 General Procedures

3.2.1.1 Facility

All of the experimental work presented herein was performed using a centrifuge located at the Vestibular Research Facility (VRF) at the NASA Ames Research Center. The VRF centrifuge is designed to provide high fidelity motion stimuli to small animals. This short arm centrifuge has two specimen test containers (STC) located at a distance of approximately 31 inches from the main spin axis. The STCs are mounted on a set of gimbals located at the end of the centrifuge arms. The computer controlled centrifuge can provide a maximum centripetal acceleration of 1.4 g (240 deg/sec). The main spin axis can provide a steady maximum angular acceleration of approximately 10 degrees per second. (See the NASA Ames Brochure entitled Vestibular Research Facility for a more complete description of the centrifuge capabilities.) Figure 3.1 shows a photograph of the VRF centrifuge.

3.2.1.2 Subjects

Six young male squirrel monkeys formed the subject pool for these experiments. Each of the monkeys was tested on at least four separate occasions, with one of the monkeys tested more than ten times. (Each test session involved approximately 10 to 15 three minute trials.)

3.2.1.3 Surgical Procedure

Two devices were surgically implanted on each of the squirrel monkeys. A head restraint device, a small stainless steel bolt, was attached on the occiput with dental cement, and a pair of small search coils were surgically attached to the right sclera. The head

VRF CENTRIFUGE

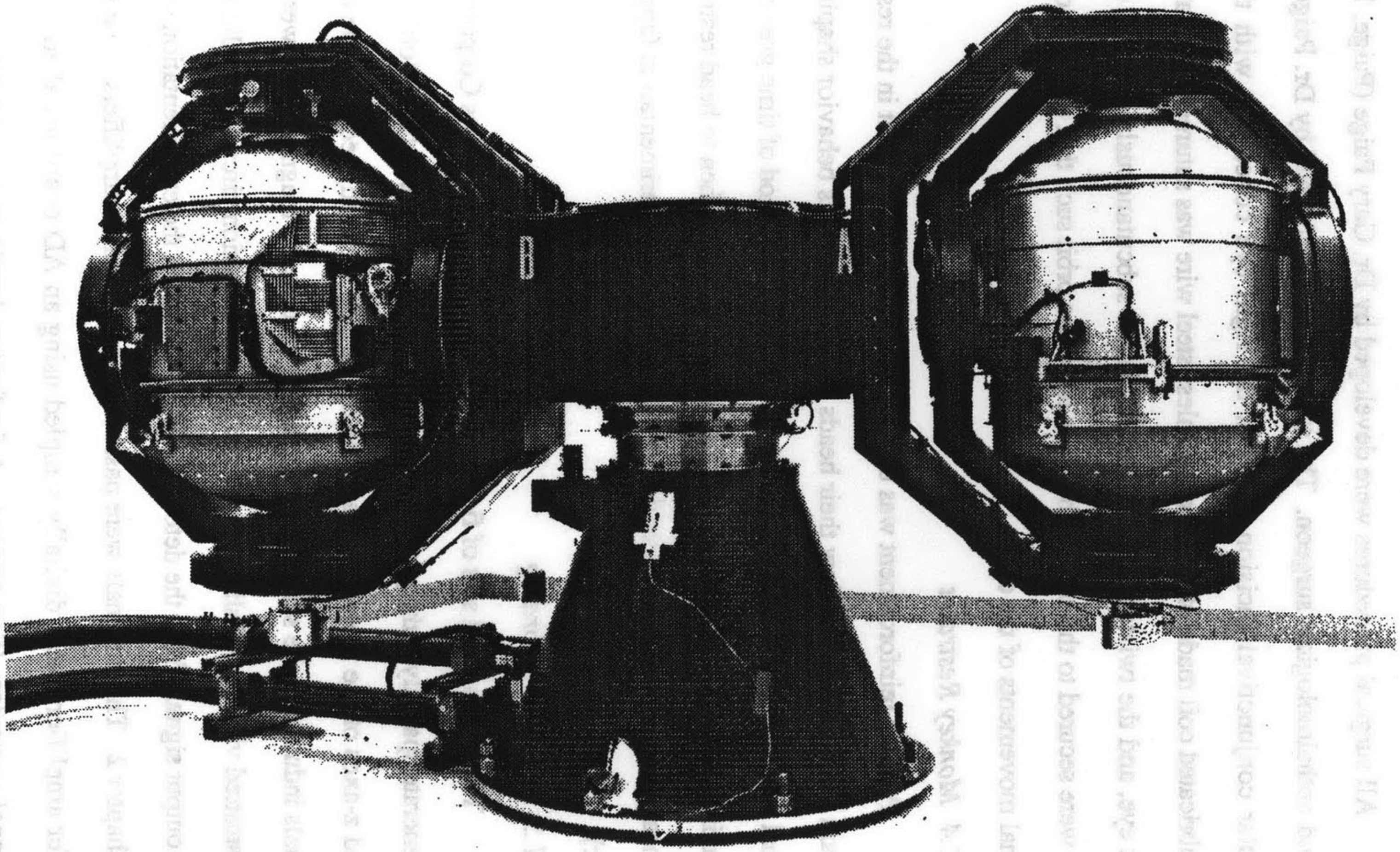


Figure 3.1

restraint device stabilized the head of the monkey, while the surgically implanted coils measured the orientation of the eye using a search coil technique (Robinson, 1963).

All surgical procedures were developed by Dr. Gary Paige (Paige, 1983a and b), a trained opthalmological surgeon. The surgery, also performed by Dr. Paige, began with a circular conjunctival incision made in the right eye concentric with the limbus. A prefabricated coil made of insulated stainless steel wire was sutured to the lateral side of the right eye, and the twisted coil leads were passed subcutaneously over the cranium where they were secured to the skull. The leads leave the orbit such that they do not interfere with normal movements of the eye.

3.2.1.4 Monkey Restraint

Positive reinforcement was used to train the monkeys to sit in the restraint chair and to train the monkeys to allow their heads to be fixed. This behavior shaping began in the animal facility and continued in the laboratory. Over a period of time greater than a month, the animals were introduced first to the restraint chair and then to head restraint. Gradually the head restraint time was extended to that needed for experimentation (approximately one hour).

3.2.1.5 Eye Coil Recording

Two orthogonal sets of field coils (Neuro Data Instruments Corporation) were used to generate spatially orthogonal magnetic fields in phase quadrature (Robinson, 1963). The y and z-axes of the monkeys head (see figure 1.2) were aligned with the magnetic fields. Signals induced in the scleral search coils by changes in magnetic flux were amplified and transmitted via a set of slip ring assemblies to a detection circuit (McElligott, et al., 1979). The output signals from the detection circuit represent the eye orientation signals discussed in Chapter 2. These signals were passed to a set of buffer amplifiers. The signals from the buffer amplifiers were digitally sampled using an AD board installed in a Compaq 386 personal computer. (See Appendix A for further details concerning data acquisition.)

An elaborate calibration procedure was developed by previous researchers (Haddad et al., 1988) to accurately calibrate the eye coil system. This procedure is listed in Appendix C of this thesis.

Eye movements were always recorded in complete darkness. No fixation points nor visual cues were ever available during any of the experimental trials. To maintain alertness and cooperation, the lights were always turned on between trials and extinguished just before the start of the next trial.

3.2.2 Centrifugation

The centrifugation experiment formed the backbone of this investigation. In the dark, the upright monkey was spun about a vertical axis displaced approximately 31 inches from its spine. The orientation of the monkey was controlled such that four primary positions were used [facing motion, back to motion, facing center, and back toward center See figure 3.2.]. Each of these four position reoriented the centrifugal force with respect to the monkey (e.g. centrifugal force out the back while facing the center). The centrifuge was accelerated as quickly as possible ($\alpha = 10$ deg/s^2) to a constant angular velocity of 200 degrees per second. This constant velocity was maintained for fifty seconds, until a symmetric deceleration brought the centrifuge to a stop. After one minute the lights were turned on, and the monkey was positioned for the next trial. Horizontal, vertical, and torsional eye movements were recorded throughout the trials.

The number of controlled parameters did not allow a complete Latin Square experiment design to be implemented. Nevertheless, the sessions were organized such that rotation direction and subject orientation were well controlled and balanced. The order in which the conditions were presented was alternated both within a given session and across sessions. Table 3.1 shows the number of trials with each set of conditions.

3.3 Results

Figure 3.3 shows three representative sets of data. The first column shows data obtained with the monkey's back toward the direction of motion. The second column

CENTRIFUGE ORIENTATIONS

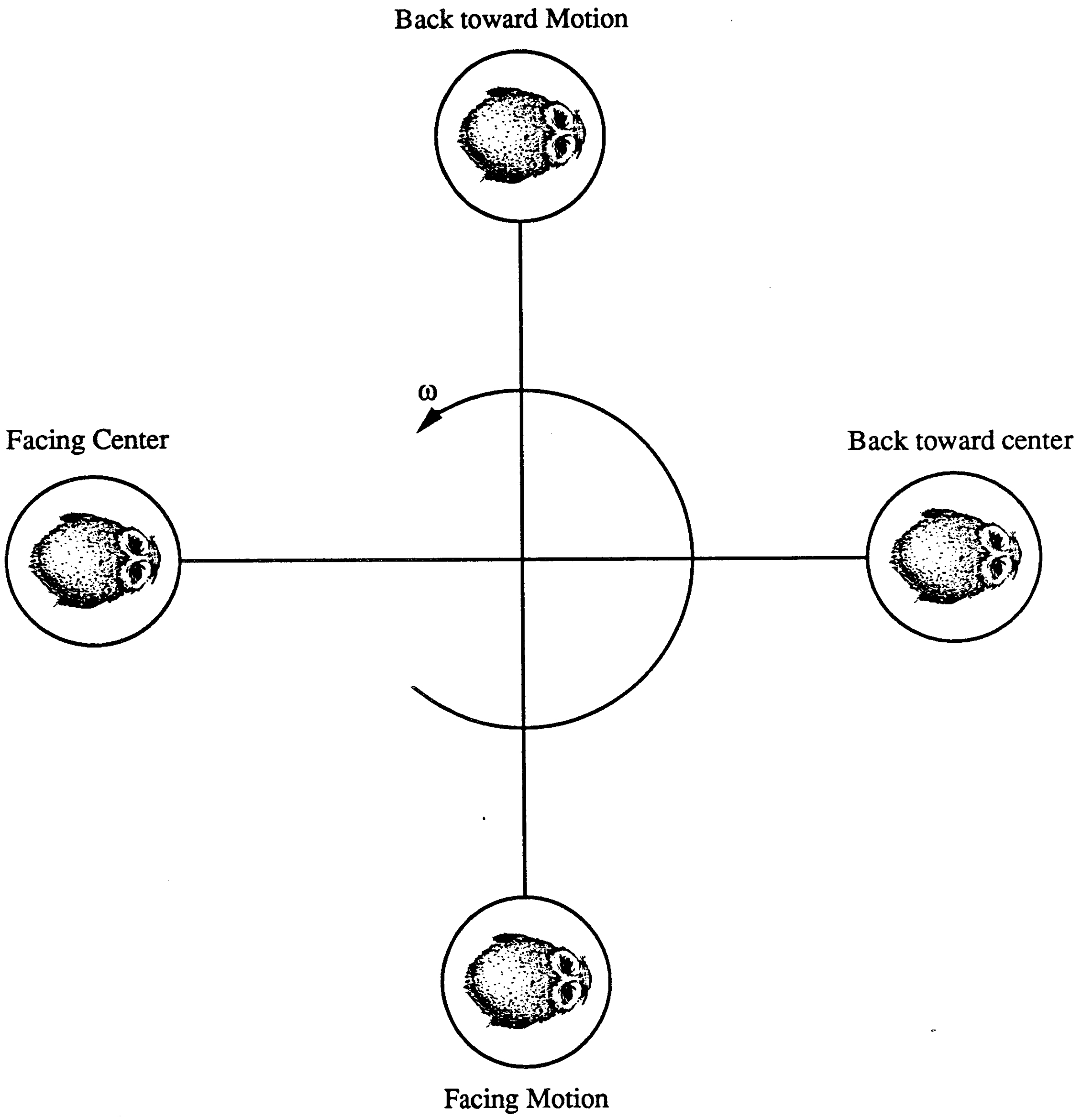


Figure 3.2

CENTRIFUGE DATA (Facing Motion and Back to Motion)

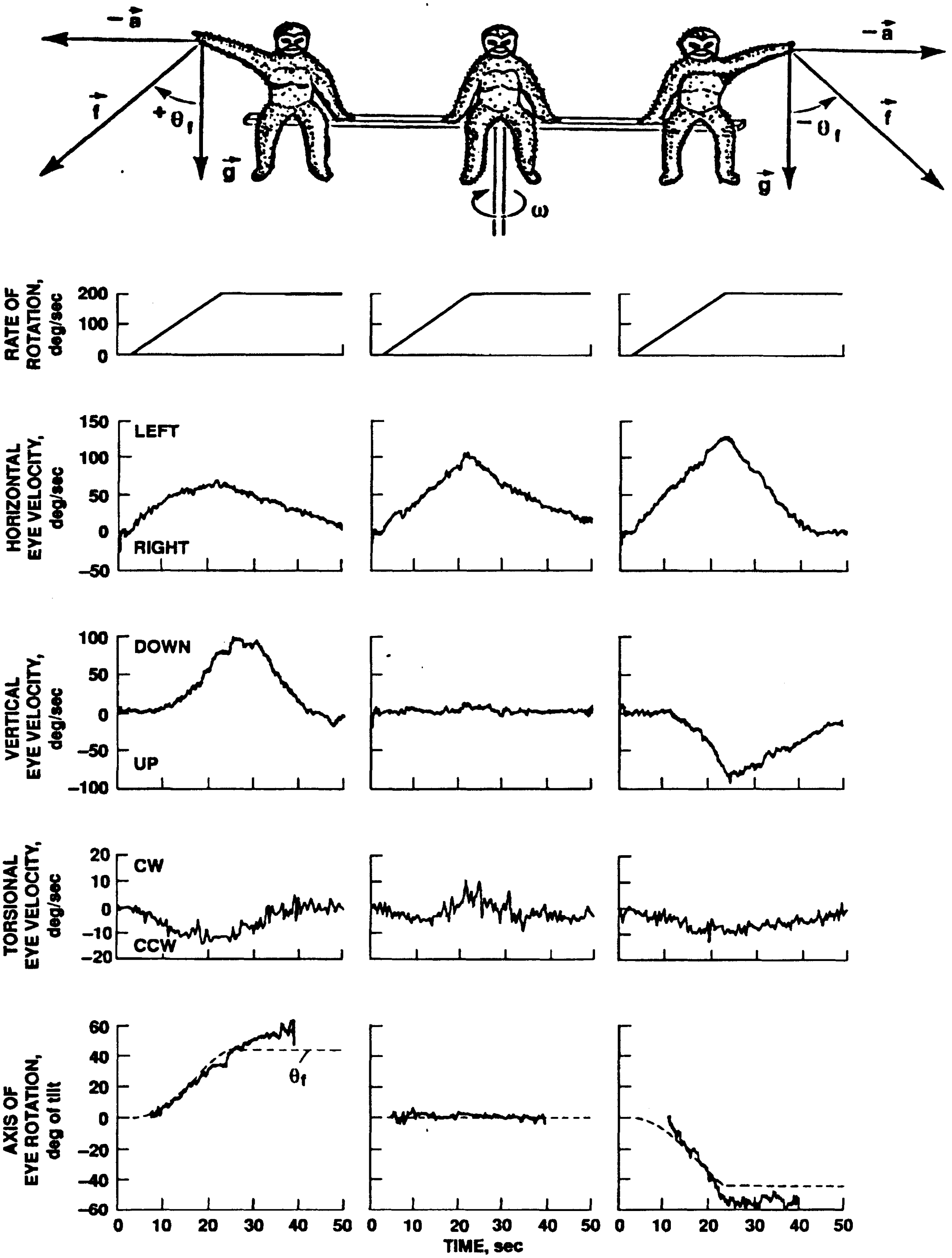


Figure 3.3

TEST MATRIX

CENTRIFUGE STUDY

SUBJECT	CW				CCW			
	F_B	B_F	I	O	F_B	B_F	I	O
A	1	3	3	3	2	1	2	2
B	2	3	6	6	3	1	4	4
D	2	1	2	2	2	1	2	2
E	1	3	2	2	2	2	2	2
F	5	3	0	0	4	4	0	0
G	2	2	0	0	2	2	0	0

"DUMPING" STUDY

SUBJECT	CW				CCW			
	RED	LED	NU	ND	RED	LED	NU	ND
A	2	2	2	2	1	1	1	1
B	2	2	2	2	1	1	1	1
D	2	2	0	0	1	1	1	1
E	1	1	1	1	0	0	0	0
F	1	1	1	1	0	0	0	0
G	0	0	0	0	0	0	0	0

Notation:

CW is clockwise.
CCW is counterclockwise.
F is facing the direction of motion.
B is back to the direction of motion.
I is facing toward the center of rotation ("in").
O is facing away from the center of rotation ("out").
RED is right ear down.
LED is left ear down.
NU is nose up.
ND is nose down.

"B_F" and "F_B" represent the order in which a pair of conditions were run.
Table entries are the number of trials with that condition.

Table 3.1

shows the "pure" rotational response in the absence of centrifugal force. The third column shows data obtained with the monkey facing the direction of motion. These orientations of the monkey are shown in the cartoon at the top of the figure.

The columns of data are organized with the centrifuge angular velocity shown as the top plot. Horizontal eye velocity is shown in the second row, followed by vertical eye velocity, and torsional eye velocity. The axis of eye rotation is shown across the bottom row. (Figure 3.4 shows a cartoon demonstrating how this angle is calculated.)

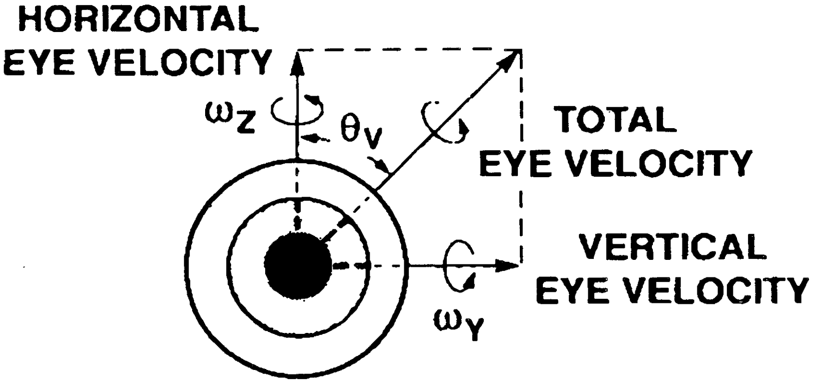
I believe this figure shows at least three important results. First, a large vertical eye velocity is generated whenever centrifugal force is present (first and third columns). Second, the magnitude of the peak horizontal eye velocity is much larger when the monkey is facing the direction of motion than when he has his back to the motion. Finally, the dominant time constant of the horizontal response is longer when the monkey has his back to the motion than when he is facing the motion.

I observed the strong vertical nystagmus on 100% of the trials when a monkey was either facing the direction of motion or had his back toward the motion. The vertical nystagmus sometimes reached a peak velocity of greater than 100 degrees per second and always exceeded 20 degrees per second.

Furthermore, the axis of eye rotation always moved toward that axis which was aligned with gravito-inertial force. This result was so dramatic and so repeatable that statistics are not really necessary to verify the significance. Nonetheless, a paired one-sided t-test was used to test the hypothesis that θ_v (Figure 3.4) was greater than zero when θ_f (Figure 3.5) was equal to +45^o and less than zero when θ_f was equal to -45^o. These statistical tests indicated that for each of the six monkeys, the axis of eye rotation tended toward alignment with gravito-inertial force ($p < 0.01$).

As previously noted, the magnitude of the peak eye velocity was generally greater when the monkey faced the direction of motion compared to when his back was toward the motion. (See Table 3.2.) Since this was not always true, I decided to test this observation

CALCULATION OF θ_V



$$\theta_V = \tan^{-1}\left(\frac{\omega_Y}{\omega_Z}\right)$$

Figure 3.4

TABLE OF PEAK VALUES

A		B		D		E		F		G	
$\lvert\overline{p_f}\rvert$	$\lvert\overline{p_b}\rvert$	$\lvert\overline{p_f}\rvert$	$\lvert\overline{p_b}\rvert$	$\lvert\overline{p_f}\rvert$	$\lvert\overline{p_b}\rvert$	$\lvert\overline{p_f}\rvert$	$\lvert\overline{p_b}\rvert$	$\lvert\overline{p_f}\rvert$	$\lvert\overline{p_b}\rvert$	$\lvert\overline{p_f}\rvert$	$\lvert\overline{p_b}\rvert$
60.2	57.8	90.0	37.3	118.7	112.6	106.3	43.1	115.5	114.8	74.0	16.2
81.4	58.6	79.8	57.0	87.5	65.4	100.1	94.3	105.2	107.4	55.5	20.7
80.0	66.1	57.5	31.7	77.7	76.9	93.1	92.5	138.9	69.9	78.7	17.2
49.9	55.8	48.8	45.3	47.9	31.0	119.5	80.1	101.8	85.9	53.0	23.3
51.4	45.5	73.9	54.0	89.9	88.6	96.9	75.5	100.6	104.5	79.1	24.4
65.8	48.9	53.5	49.6	57.9	52.8	107.6	87.4	104.8	105.4	74.4	33.1
		57.6	54.6			106.8	87.8	120.1	63.3	78.0	35.8
		58.0	29.3					94.1	76.0	80.0	32.4
		25.1	17.6					123.8	81.7		
								126.6	101.2		
								105.2	103.4		
								105.3	82.7		
								130.9	99.1		
								135.9	105.8		
								107.4	93.4		
								96.9	97.3		

Calculations:

Subject	$\overline{\lvert\overline{p_f}\rvert - \lvert\overline{p_b}\rvert}$	σ	t	p
A	9.83	12.04	2.00	$p < 0.052$
B	24.9	19.92	2.80	$p < 0.050$
D	8.70	9.62	2.22	$p < 0.050$
E	25.7	24.50	2.77	$p < 0.025$
F	20.6	22.68	3.64	$p < 0.005$
G	46.2	12.00	10.89	$p < 0.001$

Notation:

$\lvert\overline{p_f}\rvert$ is the magnitude of the average peak value when facing the motion.
$\lvert\overline{p_b}\rvert$ is the magnitude of the average peak value with back to motion.
$\overline{\lvert\overline{p_f}\rvert - \lvert\overline{p_b}\rvert}$ is the average value of $\lvert\overline{p_f}\rvert$ minus $\lvert\overline{p_b}\rvert$.
σ is the standard deviation of the above average.
t is the calculated t-value.
p represents the statistical significance.

Table 3.2

using a paired t-test. (The test sessions were designed such that these conditions were naturally paired.)

To calculate the t-values, I determined the mean horizontal SPV for one second before through one second after the constant stimulus velocity was reached. For each pair, the mean difference was calculated and a one-sided t-test was used to evaluate whether the magnitude of horizontal SPV while facing the motion was indeed greater than that obtained with back to motion. Five of the six monkeys indicated ($p < 0.05$) that the magnitude of the peak value was greater when facing the motion. The sixth monkey (A) just missed significance at the .05 level ($p < .052$). This monkey had fewer trials ($N = 6$ pairs) than most of the other monkeys.

As previously observed, the decay time constant of the horizontal SPV was generally smaller with "face toward the motion" than with "back to motion". (See Table 3.3.) The dominant time constant was calculated using the Levenberg-Marquardt algorithm to perform a least-squares fit to an adaptation response of the form:

$$spv(t) = A\, e^{-t/\tau_d} + B\, e^{-t/\tau_a} ,$$

where A and B are the gains determined as two of the fit parameters, and τ_d and τ_a are the dominant time constant and the adaptation time constant, respectively.

Since the dominant time constant was not always smaller when facing the motion, I tested this observation statistically using a paired t-test. As before, the statistical significance was evaluated using a one-sided paired t-test. Five of the six monkeys indicated ($p < 0.05$) that the time constant was smaller when facing the motion. The sixth monkey (A), again, just missed statistical significance at the 0.05 level ($p < 0.055$).

Figure 3.5 shows data in a format similar to that discussed for figure 3.3. As indicated in the cartoon at the top of the figure, the only difference is the orientation of the subject. As before, the middle column shows the rotational response in the absence of centrifugal force. But the first column now shows data obtained when the monkey was facing toward the center of rotation, while the third column shows data obtained while the

TABLE OF TIME CONSTANTS

A		B		D		E		F		G	
τ_f	τ_b	τ_f	τ_b	τ_f	τ_b	τ_f	τ_b	τ_f	τ_b	τ_f	τ_b
6.16	21.7	3.41	7.02	3.88	21.9	8.16	17.00	11.22	15.79	4.01	11.50
7.11	13.69	4.03	11.40	10.82	16.96	10.30	18.73	9.49	12.39	4.93	9.72
10.74	14.14	11.84	14.25	11.02	25.4	8.09	18.31	9.80	25.00	4.65	8.26
6.89	12.79	7.54	10.93	9.36	12.95	6.37	17.42	8.66	25.00	5.25	5.15
13.74	15.34	5.16	14.74	7.77	21.1	8.52	22.08	5.61	18.06	7.63	13.06
16.20	14.41	6.36	12.51	12.25	15.66	8.01	17.61	8.38	10.67	7.09	5.96
		8.78	12.40			10.17	29.8	10.04	26.9	5.12	6.20
		10.16	9.16			10.11	17.04	10.06	20.7	6.70	9.01
		9.83	10.16					12.15	14.71		
								6.05	14.82		
								13.57	9.65		
								8.45	9.61		
								13.01	24.5		
								15.63	19.45		
								9.78	10.50		
								11.08	12.97		

Calculations:

Subject	$\overline{\tau_b - \tau_f}$	σ	t	p
A	5.21	6.47	1.97	$p < 0.055$
B	4.35	3.58	3.84	$p < 0.010$
D	9.36	6.38	3.59	$p < 0.010$
E	11.03	4.26	7.32	$p < 0.005$
F	6.73	6.59	4.09	$p < 0.005$
G	2.94	2.94	3.00	$p < 0.025$

Notation:

τ_f is the decay time constant when facing the motion.
τ_b is the decay time constant with back to motion.
$\overline{\tau_b - \tau_f}$ is the average value of τ_b minus τ_f .
σ is the standard deviation of the above average.
t is the calculated t-value.
p represents the statistical significance.

Table 3.3

CENTRIFUGE DATA (Facing Center and Back to Center)

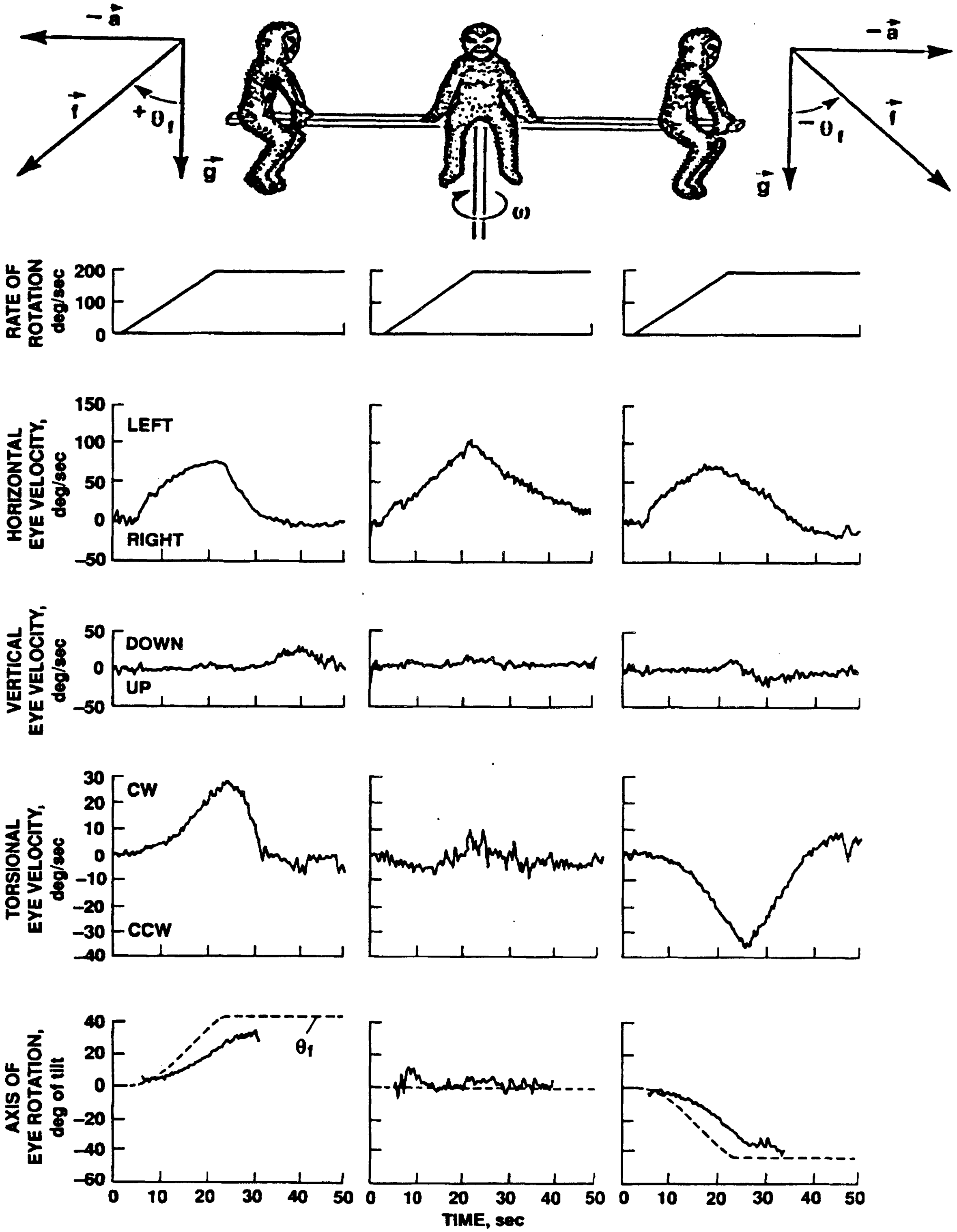


Figure 3.5

monkey had his back toward the center. The columns of data are the same as figure 3.3 except that the final row shows a different convention for the axis of eye rotation. Figure 3.6 shows how this angle of tilt is calculated.

I feel the most dramatic result shown by this graph is the large torsional eye velocity induced when centripetal force was influencing the eye movements (columns one and three). This strong torsional nystagmus was always present under these conditions. The torsional velocity sometimes reached peak velocities of greater than 50 degrees per second.

Furthermore, as before, I observed that the axis of eye rotation always moved toward alignment with gravito-inertial force. Again, this result was so dramatic and so repeatable that statistics were not necessary to verify significance. Nonetheless, a one-sided paired t-test indicates that for each of the four monkeys (A, B, D, E) tested under this set of conditions, the axis of eye rotation tended to align with gravito-inertial force ($p < 0.01$).

3.4 Controls

3.4.1 Static Tilt Tests

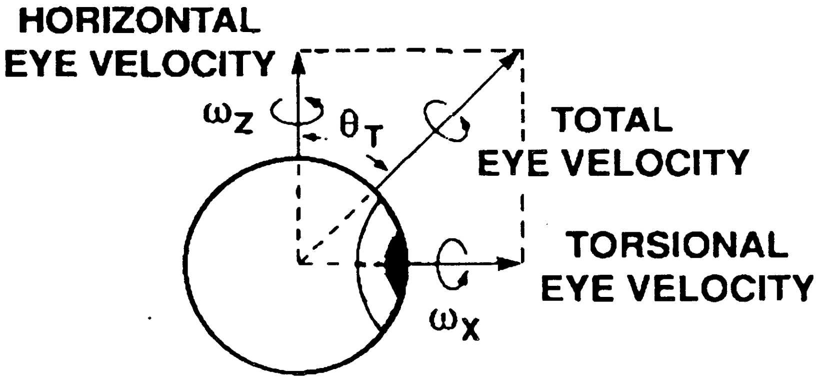
3.4.1.1 Methods and Results

Six monkeys were tested for positional nystagmus. Testing was performed by quickly tilting the monkeys to a desired orientation and holding them at that orientation for more than a minute.

Three of the monkeys were tested by tilting them 45 degrees or 90 degrees from the vertical in each of the four primary directions [Nose Down (ND), Nose Up (NU), Right Ear Down (RED), and Left Ear Down (LED). Each of these monkeys demonstrated some sustained positional nystagmus in all orientations, including the upright orientation. The nystagmus was judged to be very weak in ALL cases (slow phase eye velocity never exceeds 20 degrees/second). Figure 3.7 shows an example of this type of response.

The three remaining monkeys were tested similarly, but the orientations of the monkeys were extended to include tilts of 135 degrees and 180 degrees (to an inverted

CALCULATION OF θ_T



$$\theta_T = \tan^{-1}\left(\frac{\omega_X}{\omega_Z}\right)$$

Figure 3.6

WEAK STATIC RESPONSE

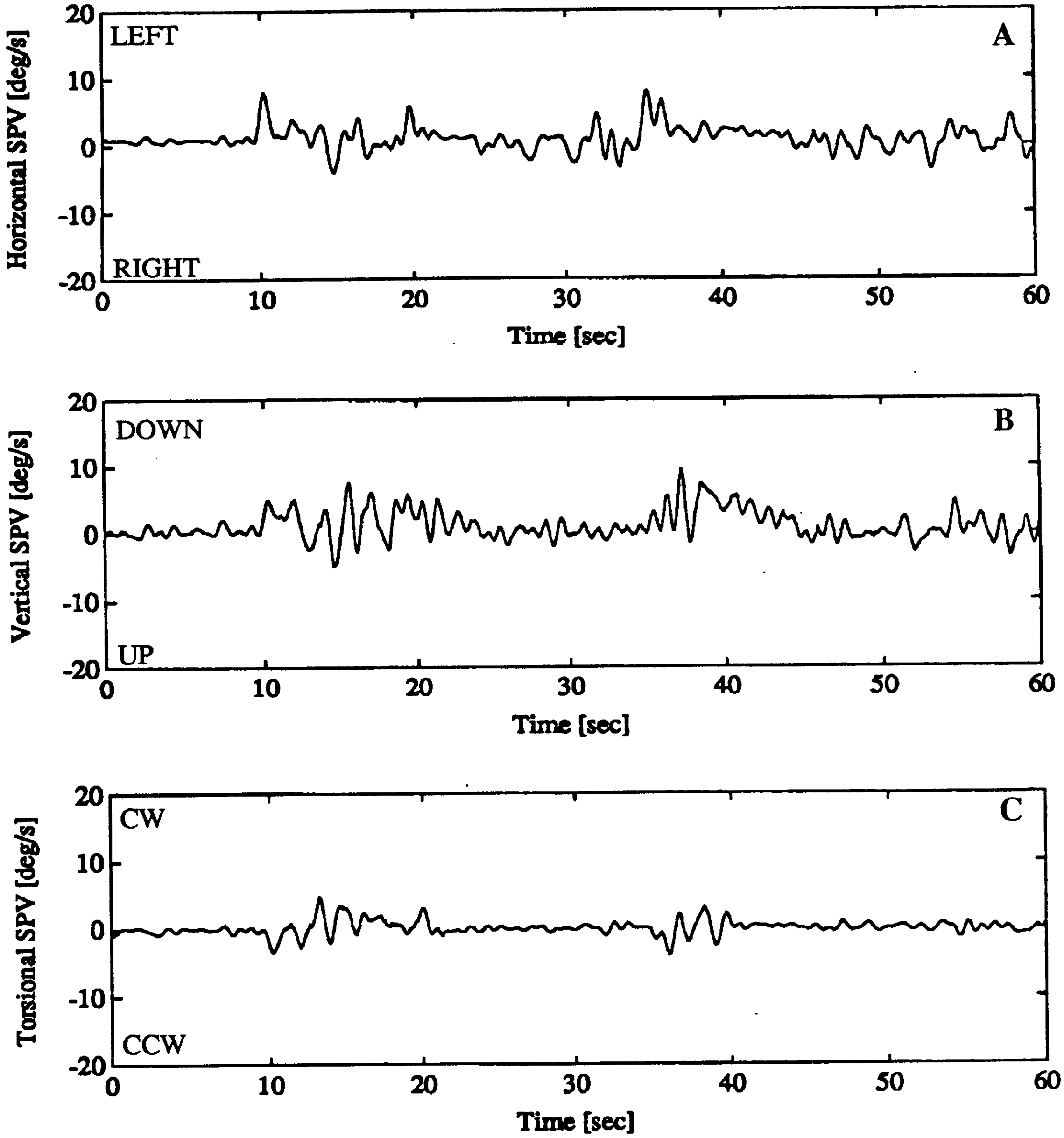


Figure 3.7

orientation) from vertical. As with the other experiments, all three monkeys, at most, exhibited very weak positional nystagmus in almost all orientations including the upright. In the inverted orientation, all of the monkeys exhibited very strong (> 50 deg/sec) nystagmus. Figure 3.8 shows a typical response during inversion. For two of the monkeys this was the only orientation which resulted in anything but weak nystagmus. The third monkey, in one out of two tests, exhibited strong nystagmus when tilted 135 degrees from the vertical.

Strong positional nystagmus was observed nine times (eight out of eight inversions and one out of six tilts at 135 degrees). Every time strong positional nystagmus occurred it was primarily vertical with the slow phase velocity pitching the eye downward.

3.4.1.2 Conclusions

The central conclusion drawn from this study is that monkeys do not demonstrate strong positional nystagmus except during inversion or near inversion. For the centrifuge studies discussed shortly the magnitude of the GIF vector never exceeded 1.4 g and never tilted more than 45 degrees. I, therefore, suggest that the strong vertical or torsional eye movements observed during centrifugation are not simply a positional nystagmus due to reorientation of the gravito-inertial force, but, rather, are induced by an interaction between the rotational responses and the tilted gravito-inertial force.

3.4.2 Dumping Tests

3.4.2.1 Methods

The second test is called dumping. In the dark, upright monkeys were spun about an axis parallel to their spine and aligned with gravity. The rotation of the monkeys was identical to that experienced by humans during a standard rotating chair test. A constant velocity rotation of 100 degrees per second was maintained for one minute and then quickly halted. Within three seconds, while the post-rotational response was very strong, the test container and monkey were manually tilted 45 degrees in one of the four primary directions (LED, RED, ND, and NU). This orientation was maintained for one minute. The animal

STRONG STATIC RESPONSE

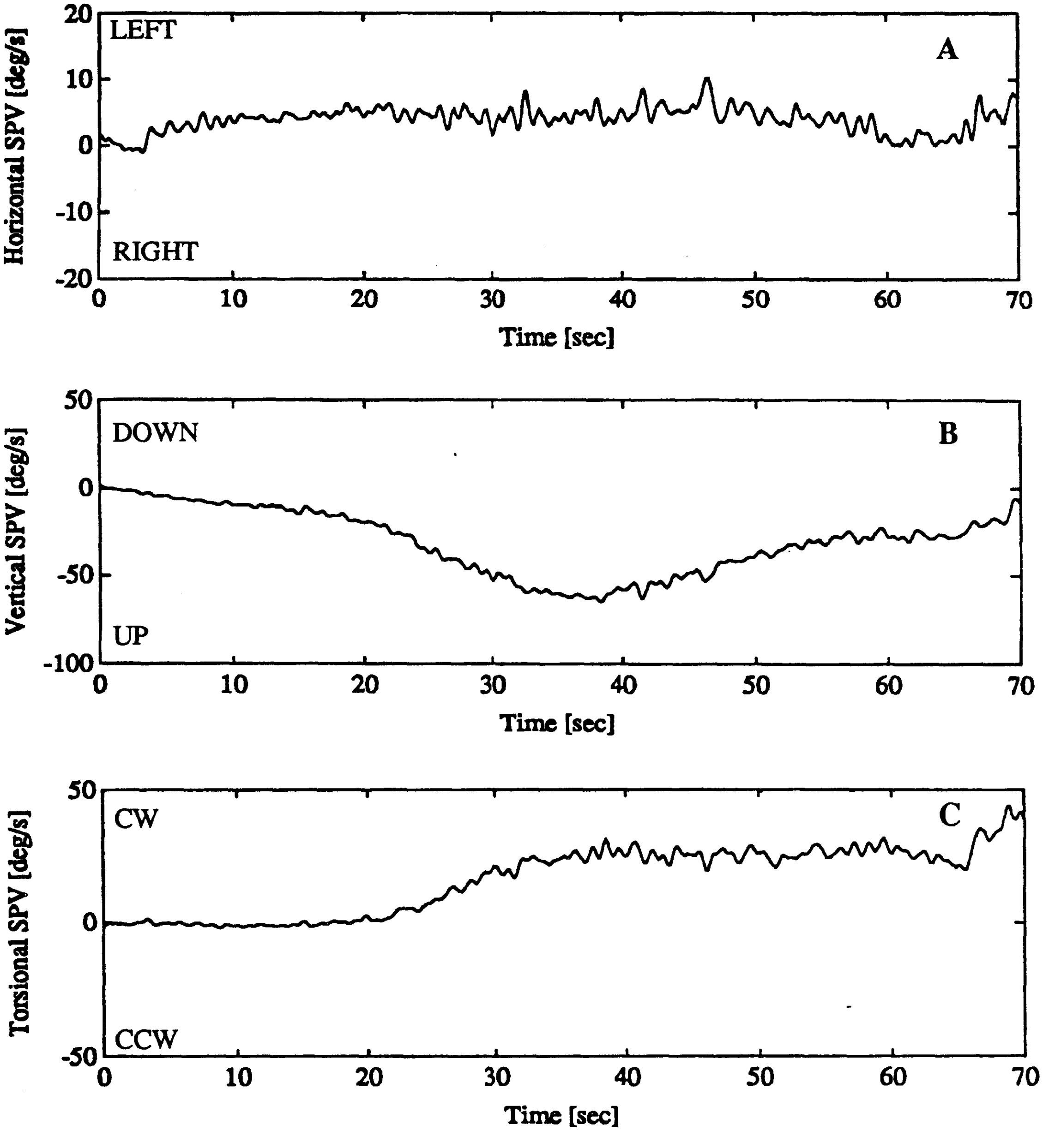


Figure 3.8

was then returned to the upright position and the lights were turned on for one minute. The test was repeated in each primary direction of tilt.

3.4.2.2 Results

Analysis of the data supports three conclusions. First, as demonstrated in figure 3.9, when the monkey was tilted to the left (LED) or to the right (RED) a very strong vertical nystagmus (in a head fixed reference frame) always built up and then decayed with the horizontal nystagmus. (Similar to results described previously, Raphan and Cohen, 1981; Harris, 1987) Similarly, as demonstrated in figure 3.10, when the monkey was tilted forward (ND) or backward (NU) a strong torsional nystagmus always built up and then decayed with the horizontal nystagmus. Finally, in all cases the axis of the eye movements tended to align with the gravitational field of the earth (Harris, 1987).

As seen in figures 3.9 and 3.10, the shift in the axis of eye rotation occurs very quickly. Time constants for this shift were estimated to be less than five seconds.

3.4.2.3 Conclusions

The central conclusion drawn from this study is that squirrel monkeys tend to align the axis of eye rotation with gravity. This alignment occurs very quickly. These facts are consistent with the notion that the axis of eye rotation provides a reliable measure of "down" in the squirrel monkey.

These tests confirm the earlier observation that the eye movements observed during centrifugation are the result of an interaction between the rotational response and the orientation of the gravito-inertial force. In the following discussion, I will assume that the axis of eye rotation provides a measure for the perception of gravitational force. This interpretation is supported by a number of other studies (Raphan et al 1981; Harris, 1987).

3.5 Discussion

I have statistically shown that the squirrel monkey's axis of eye rotation tends to align with gravito-inertial force during centrifugation. This axis transformation is robust

DUMPING (Right Ear Down)

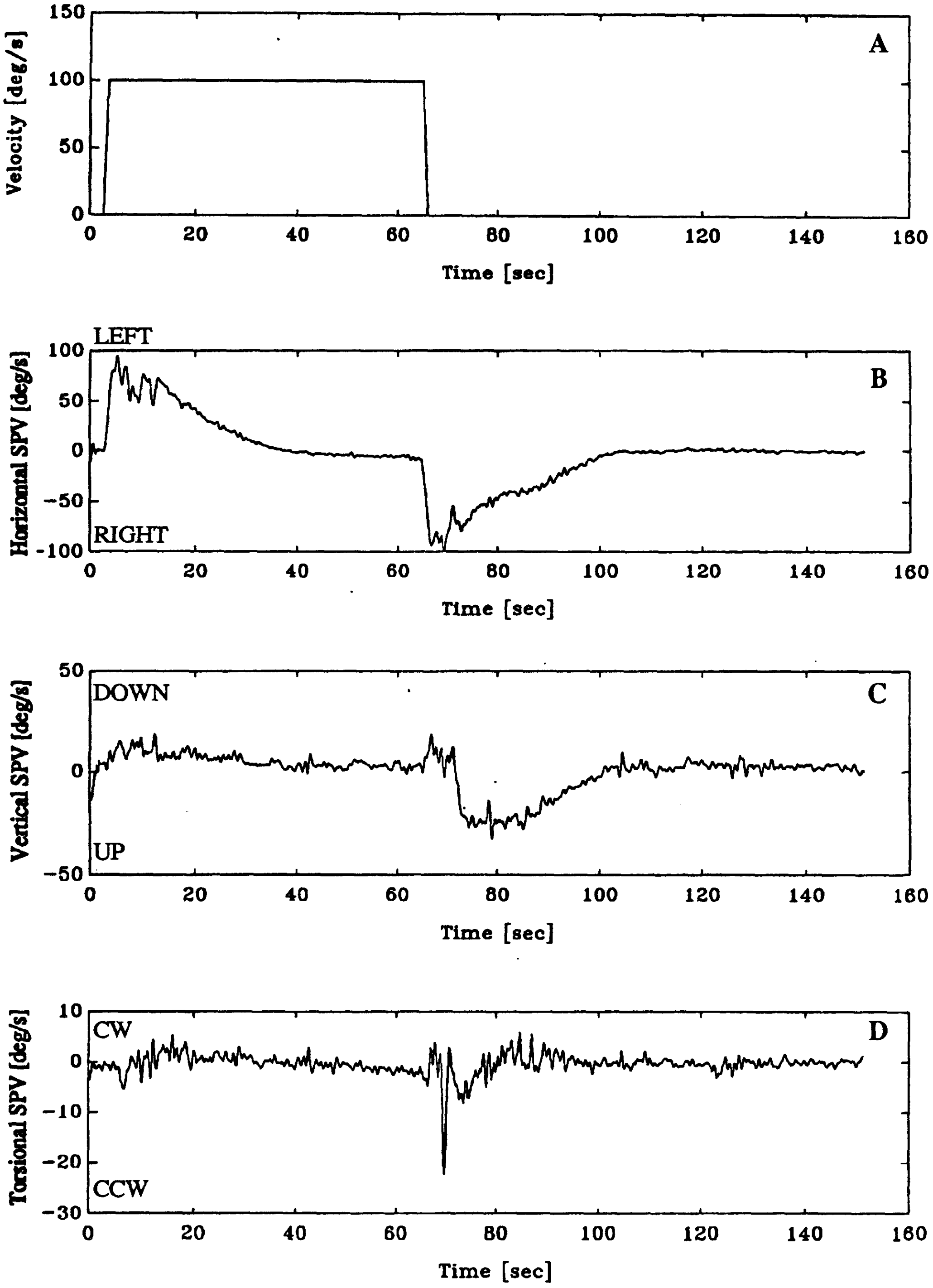


Figure 3.9

DUMPING (Nose Down)

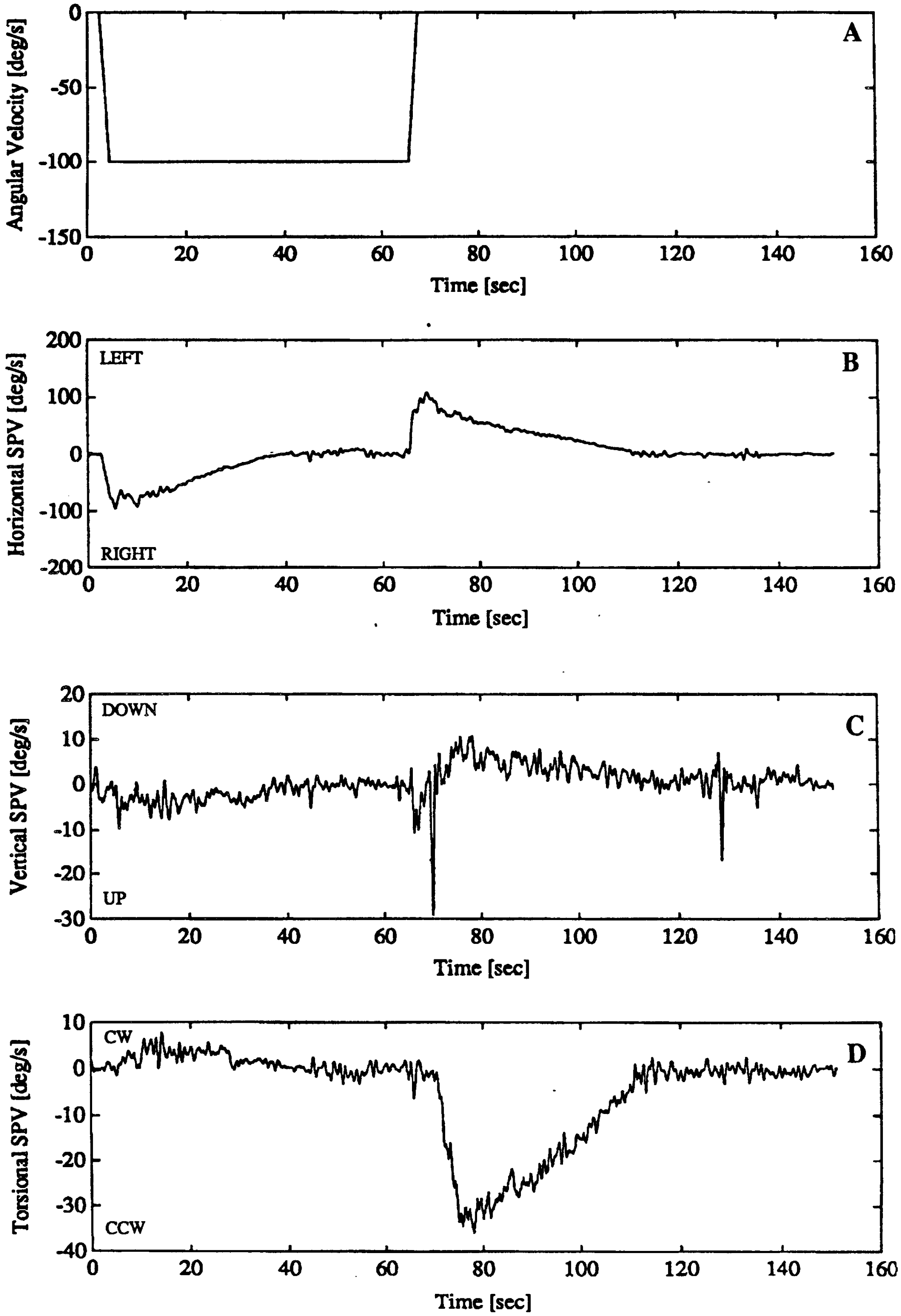


Figure 3.10

and repeatable. Furthermore, this response is a logical way to resolve the sensory conflict induced by centrifugation.

During centrifugation in the absence of visual information to the contrary, all sensory systems (otoliths, tactile sensors, proprioceptors) indicate a consistent direction for the total GIF. Since the only constant gravito-inertial force naturally experienced is gravitational, it seems logical that the CNS interpret a constant GIF as "down". At the same time, the semicircular canals respond to angular acceleration and indicate a rotation about the body's yaw axis. If gravity were truly tilted with respect to rotation, the otoliths would sense a sinusoidal rotation of gravity which would confirm the rotation sensed by the semicircular canals.

During centrifugation, however, the orientation of gravito-inertial force is constant. This results in a sensory conflict. Centrifugation provides a yaw rotation which is sensed by the semicircular canals, but the otoliths do not confirm that gravito-inertial force is also rotating. In fact, the otoliths provide a sensory signal which indicates that no rotation is taking place unless the rotation exactly aligns with gravito-inertial force.

One way to resolve the conflict is to transfer activity representing rotation to an axis which is aligned with gravito-inertial force. Then the presence of angular rotation and a constant GIF do not conflict. This is a logical way to resolve the sensory conflict induced by centrifugation, and this is the resolution process exhibited by squirrel monkeys.

I also have statistically shown that the peak value of SPV is dependent upon the orientation of the subject. This result can be explained by the LVOR hypothesis first suggested by Young (1967). If the rotational VOR and the linear VOR are summed to yield the total VOR (See Figure 3.11.), then the magnitude of the sum will be dependent upon the orientation of centripetal acceleration. As discussed by Young, the magnitude of this response should increase when the subject is facing the motion because the responses are additive. If the subject has his back to the motion the LVOR should decrease the magnitude of the total response.

SIMPLE VOR MODEL

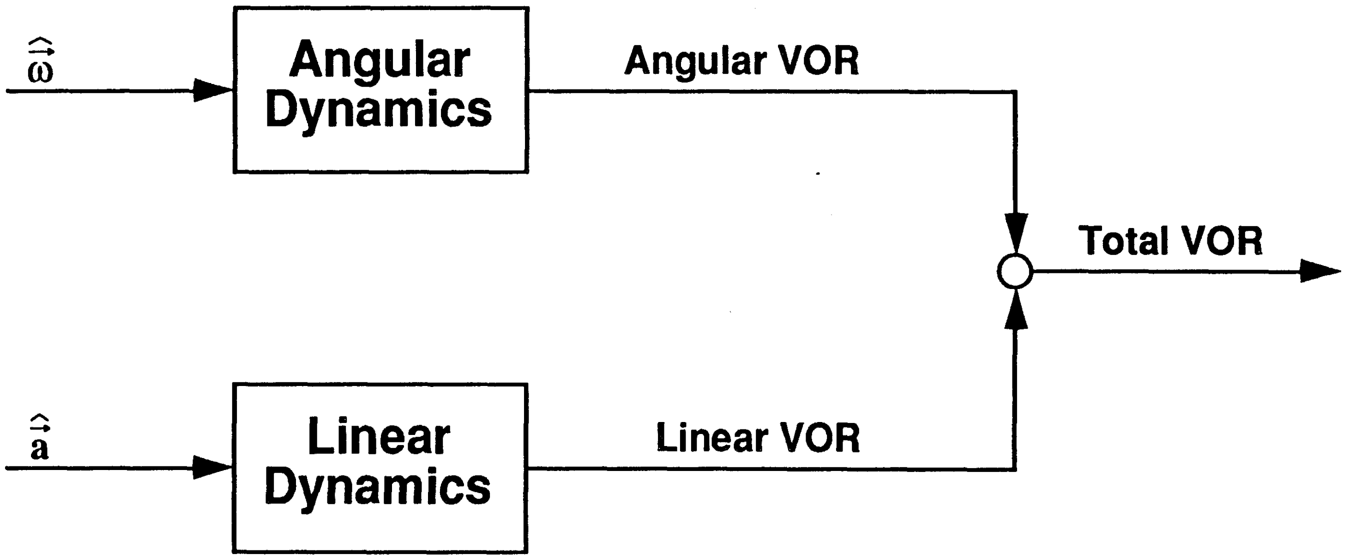


Figure 3.11

This statistical finding encourages further analysis of the data, and if we are willing to accept the simple model shown in Figure 3.11, then straightforward calculations yield the linear and rotational responses. This model states that the total VOR is simply a sum of the angular component and a linear component. The angular component is independent of whether the monkey is facing the motion or has his back to the motion, while the sign of the linear component will be dependent upon the orientation of the centripetal acceleration with respect to the subject. Therefore, by subtracting the eye movements obtained with the monkey's back to the motion from those obtained while facing the motion, we can estimate the linear component (Young 1967).

This type of analysis was performed for each pair of conditions. Figure 3.12c shows one typical type of response. The peak velocity of this response occurs very near the time at which a constant velocity is obtained. This response quickly decays to zero. This response is qualitatively similar to that observed in humans (Lansberg et al, 1965; Young, 1967).

A second type of response was also observed. Figure 3.13c shows an example of this type of response. As before, the peak linear velocity occurs very near the time at which a constant velocity is reached. Again the response decays toward zero, but then significantly overshoots and gradually settles back to zero.

Both types of response were well represented in the data. The simple decay occurred approximately 25% of the time, while the overshooting response accounted for most of the remaining responses. The dominant time constant for decay was estimated to be approximately 6.51 seconds. This time constant appears to be independent of the type of response.

In the previous section I used statistics to demonstrate that the decay time constant was dependent upon the orientation of centripetal acceleration. This result may also be explained by the simple VOR model shown in figure 3.11. Since the linear dynamics were observed to be fast relative to the rotational dynamics, the linear VOR will quickly decay

LINEAR VOR (Single Peak)

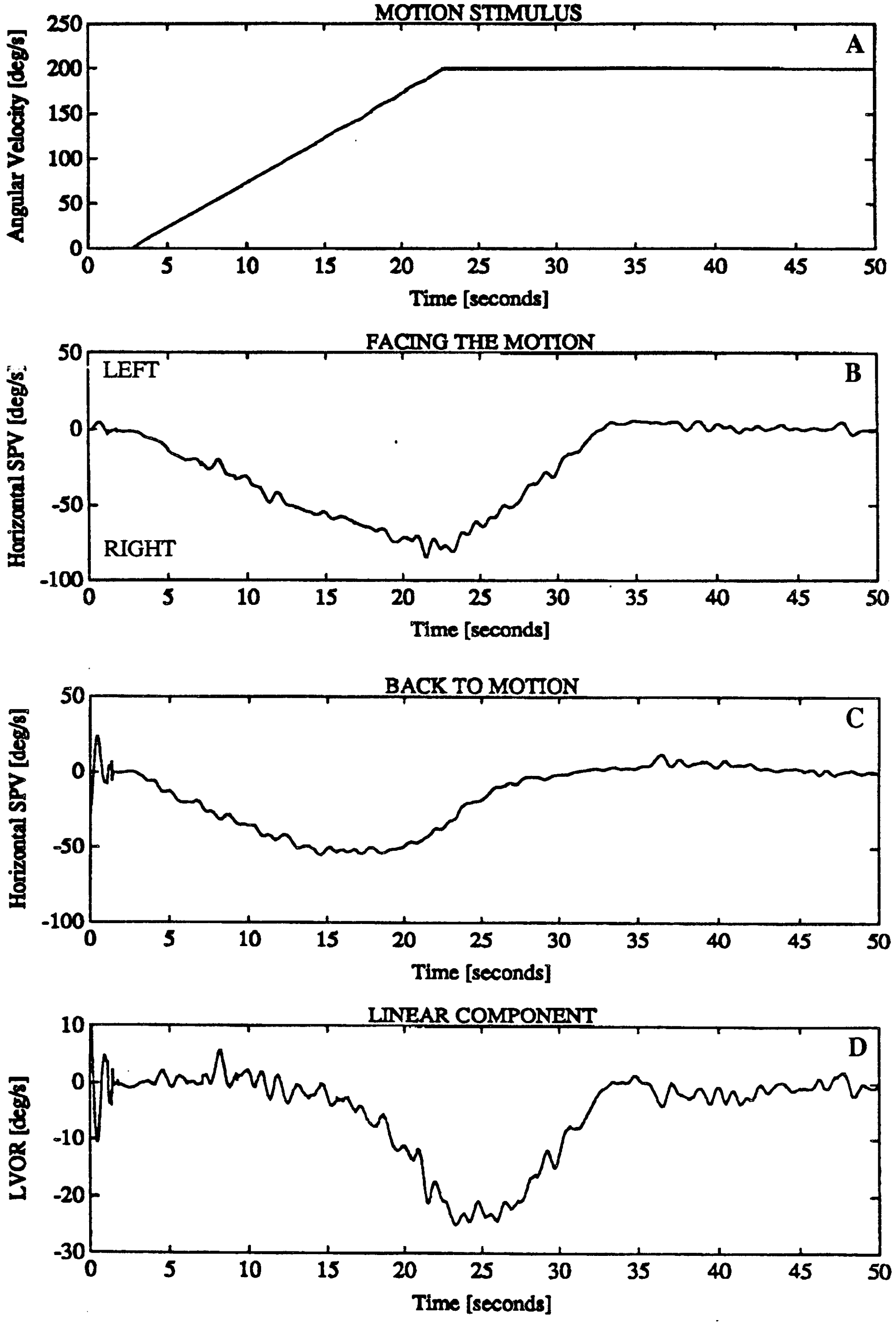


Figure 3.12

LINEAR VOR (Double Peak)

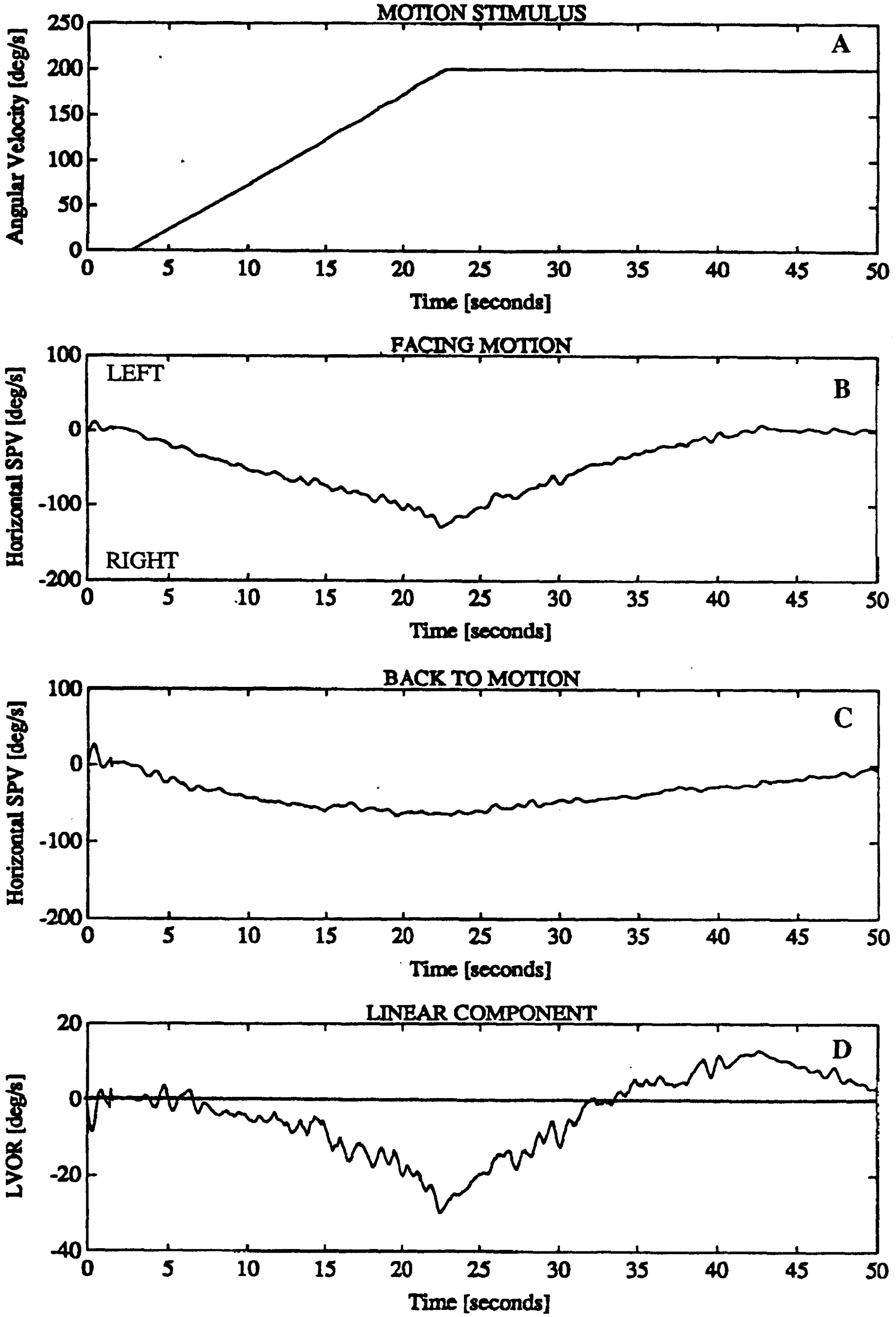


Figure 3.13

leaving only the rotational VOR. The decay will be faster when it starts from a higher value (rotational VOR + linear VOR) than when it starts from a lower value (rotational VOR - linear VOR).

The statistical results found for the magnitude of the peak SPV and the decay time constants strongly support the hypothesis that the total VOR is the sum of two components; the linear VOR and the rotational VOR. This in turn supports the calculations used to determine the linear VOR.

A reasonably simple model also explains the difference between the two types of linear responses. The internal estimate of gravity (or the perception of "down") always lags slightly behind the changing gravito-inertial force. The difference between the sensed gravito-inertial force and the internal estimate of gravity is interpreted as linear acceleration. Shortly after the rotation of the gravito-inertial force stops, the internal estimate of gravity aligns with gravito-inertial force. At this point, no linear acceleration is sensed. Therefore, the linear acceleration is only interpreted as such while the perception of down is not aligned with GIF.

A single peak decay will result when the internal representation of gravity lags behind GIF during the rotation of GIF and then slowly settles into alignment with GIF in the steady state. A double peak response will result when the rotation of GIF induces a sense of whole body rotation. When GIF and the internal gravity state are aligned, the inertia of the internal sense of rotation drives the internal state past the measured gravito-inertial force. The difference between the internal gravitational state and GIF now has a sign opposite to the previous sense of linear acceleration. Therefore, the reflexive response should also reverse. This will result in an overshoot which will then decay to a steady state.

Monkey OVAR data strongly supports this hypothesis. As previously discussed constant rotation of gravity can induce a large bias component in the compensatory response. Presumably, this bias component represents a sense of constant rotation which is induced by the otolith organs. This provides strong evidence that rotation of gravito-inertial force does induce a sense of rotation which could be compelling enough to the rotate the internal estimate of gravity past the measured GIF.

As previously discussed, humans do not show as pronounced a bias component as that demonstrated by monkeys. This is consistent with the fact that humans do not demonstrate an overshoot in their LVOR during centrifugation (Young, 1967). A weak bias component represents a weak sense of rotation induced by the otolith, and a weak sense of otolith mediated rotation would suggest little, if any, overshoot in the linear response.

The fast decay of linear VOR in the squirrel monkey indicates that the internal representation of gravity quickly aligns with gravito-inertial force. The axis of eye rotation quickly changes toward alignment with GIF.

Humans exhibit a much slower change in the axis of eye rotation. The data of Lansberg et al. (1965) and Crampton (1966) do not indicate vertical nystagmus until well into the motion profile. This is consistent with a slow change in the LVOR as estimated by Young (1967).

All of these differences between humans and monkeys support the original hypothesis: that GIF is resolved into a component representing gravity and a component representing linear acceleration.

The monkey data indicate that monkeys quickly align their perception of down with gravito-inertial force. This explains a large bias and a small modulation during OVAR, it explains the rapid axis transformations observed during centrifugation and dumping, and it explains the fast dynamics of the LVOR measured during centrifugation.

The human data indicate that humans are much less likely to align their perception of down with GIF. This explains the small bias and large modulation observed during OVAR, it explains the slow axis transformations observed during centrifugation, and it explains the relatively slow dynamics of the LVOR measured during centrifugation.

Experience provides a plausible explanation for these differences. Humans in the modern world are often exposed to somewhat unnatural constant accelerations (cars, trains, airplanes, elevators, etc.). This experience teaches us that constant gravito-inertial force does not always indicate gravitation. On the other hand, few animals are regularly exposed to such linear accelerations. Without this experience, monkeys and other animals, are easily tricked into interpreting any constant GIF as "down." This simple hypothesis could explain some of the previously discussed differences between humans and monkeys.

3.6 Summary and Conclusions

Statistical analysis of the centrifuge data supports three conclusions. First, the axis of eye rotation moved toward alignment with gravito-inertial force. Second, the magnitude of the slow phase eye velocity was larger when the monkey was facing the motion than when the monkey had his back toward the motion. Finally, the dominant time constant of decay was larger when the monkey was facing the direction of motion than when his back was toward the motion.

The control studies, static tilt tests and dumping test, suggest that the responses are due to an interaction of rotational and linear responses and that the axis of eye rotation provides, in the absence of any other information, an estimate of "down."

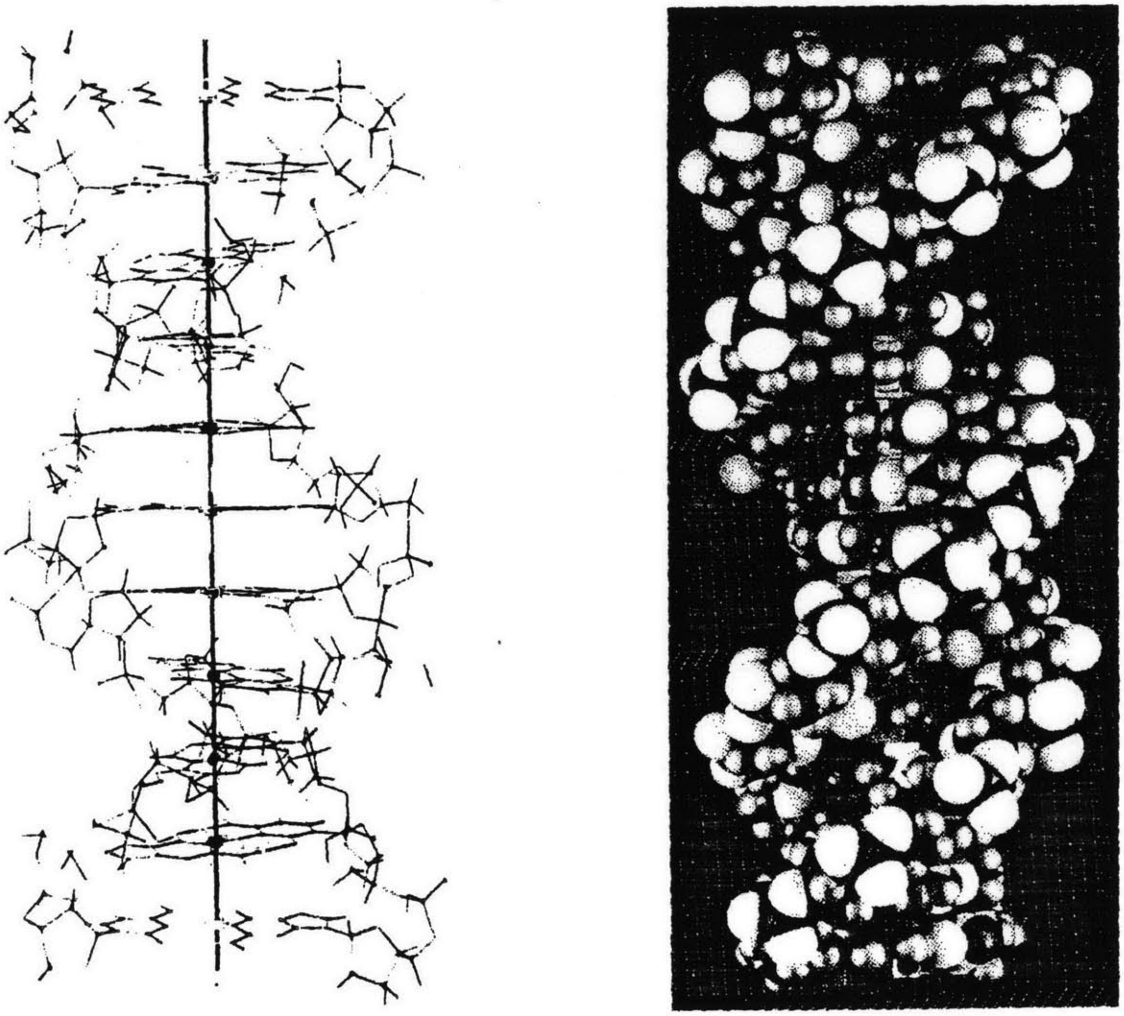
Further analysis of the centrifuge data indicates that a linear component of the response may be extracted. The time course of the linear component indicates a very fast dynamic process. Humans show a similar linear component, but the dynamics are much slower.

The data also indicate that monkeys quickly align their perception of down with the gravito-inertial force. Humans, also, eventually align their perception of down with GIF, but the alignment process is more gradual.

All of these results and observations are consistent with the GIF resolution hypothesis, and therefore, provide evidence supporting this hypothesis.

IV

SENSORY CONFLICT MODEL



DNA Models

"Rules and models destroy genius and art."
William Hazlitt

"A moments insight is sometimes worth a life's experience."
Oliver Wendell Holmes

CHAPTER 4

SENSORY CONFLICT MODEL

4.0 Introduction

In this chapter I present a model of spatial orientation. Specifically, the model predicts how the squirrel monkey estimates gravitational "down", linear motion, and rotational motion in the presence of conflicting sensory information. The model is called a "sensory conflict model" since explicit calculations are performed which compare sensory afference with expected sensory afference. "Sensory conflict" is defined to be the difference between these quantities. The sensory conflict vector is used in a feedback schema similar to that used in optimal observer theory (Kalman-Bucy filtering; Kalman and Bucy, 1961) to drive the internal estimate of spatial orientation toward the true state of orientation.

The model is presented in four sections. In section 4.1, I present historical perspectives which lead to the model formulation. In section 4.2, I theoretically develop the model and use some simple examples to illuminate some central concepts underlying the model. In section 4.3, I model a variety of experimental data from the literature while emphasizing the qualitative similarity between the model's predictions and the experimental data. Finally in section 4.4, I discuss some of the problems associated with the model and suggest a number of improvements and/or additions.

Since this model is formulated using engineering system theory and since the language of system theory is the block diagram, a large number of figures are distributed throughout the chapter. Simultaneously, in the text, the necessary equations and/or descriptions are presented. These approaches should complement one another, though an understanding of either approach is sufficient.

4.1 Background

4.1.1 Physiological and Psychological Models

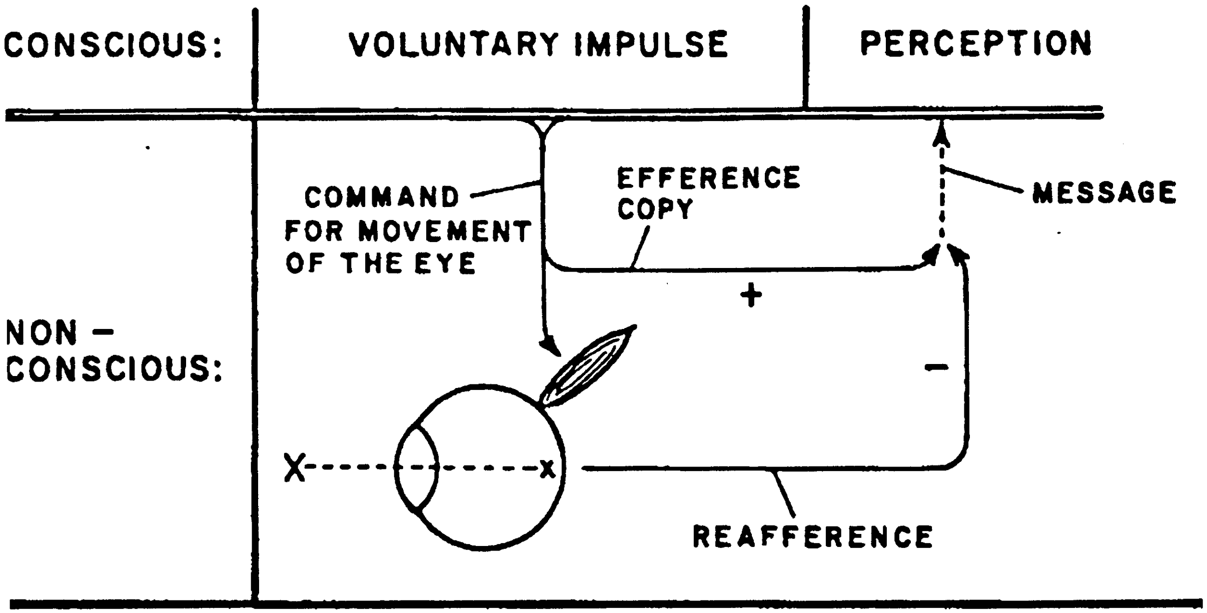
In 1954, von Holst proposed a model that has come to be known as the "efference copy" hypothesis. This hypothesis sprung from a set of experiments carried out in collaboration with Mittelstaedt (von Holst and Mittelstaedt; 1950). These experiments investigated how the CNS distinguishes sensory information due to self-generated motion from the sensory information due to external disturbances. Specifically, they investigated how the CNS distinguishes visual afference due to self-generated motion from visual afference generated by movements of the entire visual field. They concluded that sensory systems can be stimulated in two different ways, thus resulting in different types of afference. Muscular activity leading to self motion induces "re-afference" while all external motion produces "ex-afference."

Later von Holst proposed the model schematically shown in Figure 4.1. He proposed that motor commands leave an image of themselves somewhere in the CNS. This image is then compared to the reafference generated by the movement. He claimed that the "efference copy" and the reafference would cancel each other, and thus result in the perception of constancy of the external world. Von Holst stated that a motor command,

> "leaves an image of itself somewhere in the CNS, to which the re-afference of this movement compares as the negative of a photograph compares to the print; so that, when superimposed, the image disappears. A motor impulse, a 'command' from a higher centre causes a specific activation in a lower centre, which...(gives) rise to a specific efference to the effector (i.e. a muscle, a joint, or the whole organism). This central stimulus...'the image' of the efference, may be called 'efference copy'. The effector, activated by the efference, produces a re-afference which returns to the lower centre, nullifying the efference copy by superposition."

An adjustment was later made to this model when physiologists and psychologists learned that the CNS could adapt to consistent forms of sensory rearrangement. Held and some of his associates (Held, 1961; Hein and Held, 1961) showed the importance of active movements during the adaptation process. Held recognized that the efference copy and the re-afference generated by self motion could not simply be compared since one is a motor

VON HOLST'S MODEL



After: Von Holst (1957)
From: Oman (1982)

Figure 4.1

command and the other is sensory afference. He also found that the von Holst model didn't allow for adaptation. Therefore, he proposed a scheme whereby a copy of the efferent signal is sent to a hypothetical structure called the "correlation storage." A "comparator" was also proposed to compare the output of correlation storage and the re-afferent signal . Figure 4.2 shows a representation of Held's model. As Held said,

> "...we assume that the re-afferent signal is compared (in the Comparator) with a signal selected from the Correlation Storage by the monitored efferent signal. The Correlation Storage acts as a kind of memory which retains traces of previous combination of concurrent efferent and re-afferent signals. The currently monitored efferent signal is presumed to select the trace combination containing the identical efferent part, and to activate the re-afferent trace combined with it. The resulting revived re-afferent signal is sent to the Comparator for comparison with the current re-afferent signal. The outcome of this comparison determines further performance."

The correlation storage model solved two difficulties associated with the von Holst model. First, this model allowed for sensory adaptation to occur to consistent forms of sensory rearrangement. Second, this model allowed for a conversion of muscle commands to the expected re-afference.

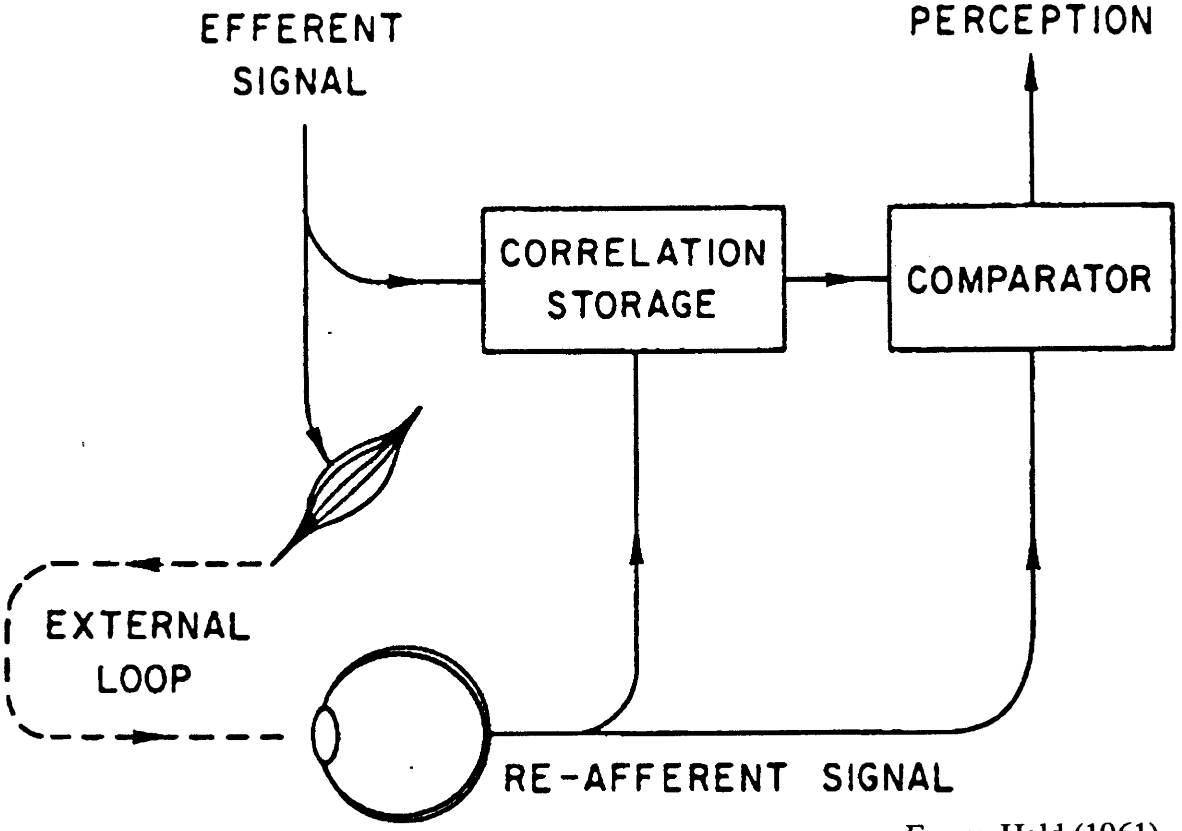
Reason (1977,1978) further developed Held's correlation storage model when he noted that earlier hypotheses had looked at sensory conflict as an incompatible difference which existed between sensory afferents. He emphasized that conflict is only defined when current sensory inputs differ from the sensory patterns which we expect based upon the previous exposure history. Figure 4.3 shows the basic structure of Reason's model. Reason cogently argued that the,

> "conflict is between the present sensory information and that retained from the immediate past, or what Held called 'exposure history'....It is this crucial temporal comparison between present and past patterns of spatial stimulation that provides the necessary explanatory link between the sensory rearrangement notion and protective adaptation."

4.1.2 Classical Engineering Systems Models

"Velocity storage" is a concept which is used as a central component in almost all of the current classical control models of spatial orientation. This element was developed when it was observed that per and post rotatory nystagmus last well beyond the activity of

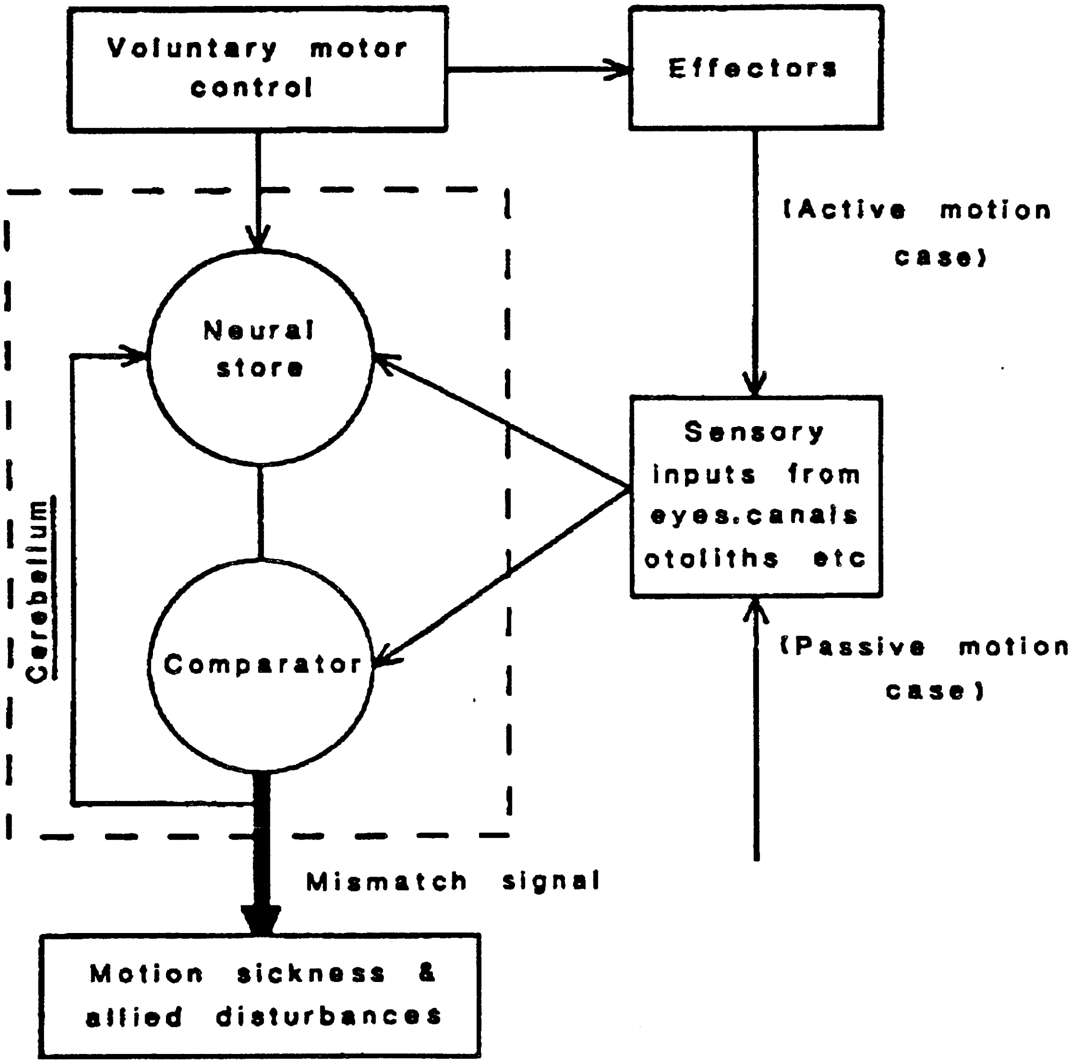
HELD'S MODEL



From: Held (1961)

Figure 4.2

REASON'S MODEL



From: Reason (1978)

Figure 4.3

the first order afferents from the semicircular canals. At nearly the same time OKAN was noted as another response which lasted beyond the visual stimulation. Furthermore, it seemed that the time course of these responses were similar. The velocity storage hypothesis proposed that a single neural element was responsible for the extension of vestibular nystagmus and for OKAN. Two somewhat different models achieve this effect. One model (Robinson, 1977) uses feedback, while another (Raphan et al., 1977) uses feedforward.

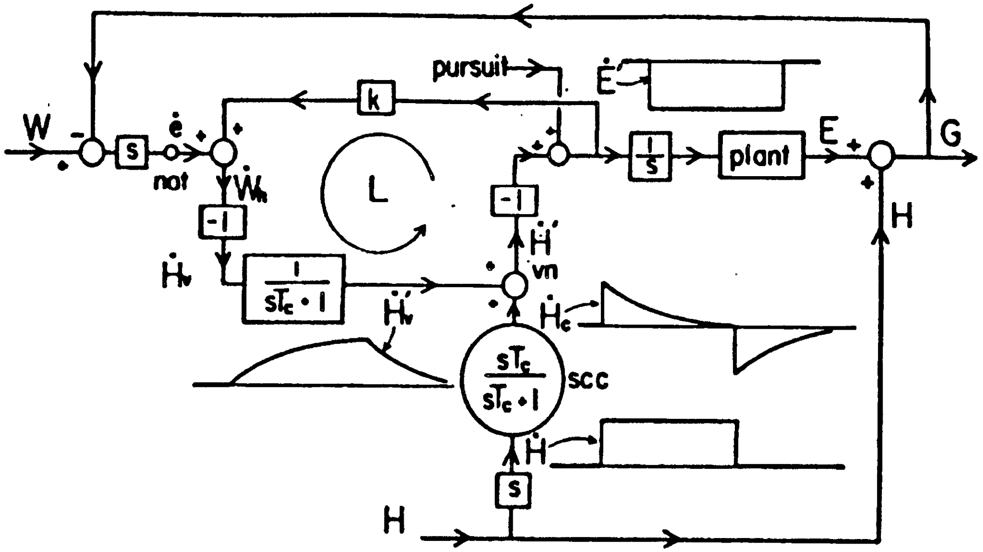
One form of Robinson's model is shown schematically in Figure 4.4. Robinson created this model in an attempt to explain OKAN and the time course of the VOR. But, more importantly, Robinson attempted to implicitly implement Newton's second law of motion. As discussed by Robinson,

> "The internal positive feedback loop L ... has an interesting general interpretation. ... its main feature is perseveration. That is, once a positive feedback loop is set into activity, it tends to feed itself by self-excitation and prolong that activity. Now this is just a restatement of Newton's second law of motion. Whenever a body (you) is set into motion, it is presumed that you will continue in that motion until you are acted upon by another force, at which time your canals will tell you. The perseveration, or holding, or integrating, or accumulating action of the loop L, is simply a demonstration of the recognition by the brain of this law of physics."

A schematic view of the model proposed by Raphan, Cohen, and their associates (Raphan et al., 1978; Raphan et al., 1979) is shown in figure 4.5. This model accomplishes velocity storage through parallel processing of the sensory afferents. The direct path mainly induces rapid changes in eye velocity. The indirect path includes a low-pass filter (called a "leaky integrator" by Raphan et al.) which stores activity and induces slow changes in eye velocity. These authors coined the term "velocity storage" to label the process by which neural activity is extended beyond the sensory afference.

Raphan and Cohen (1986) have generalized their model to include more than a single dimension of rotation. The model was extended by including three velocity storage elements: one for each direction in 3-space. The model incorporates feedback parameters which are modified by otolith afferents (and hence by the orientation of gravito-inertial

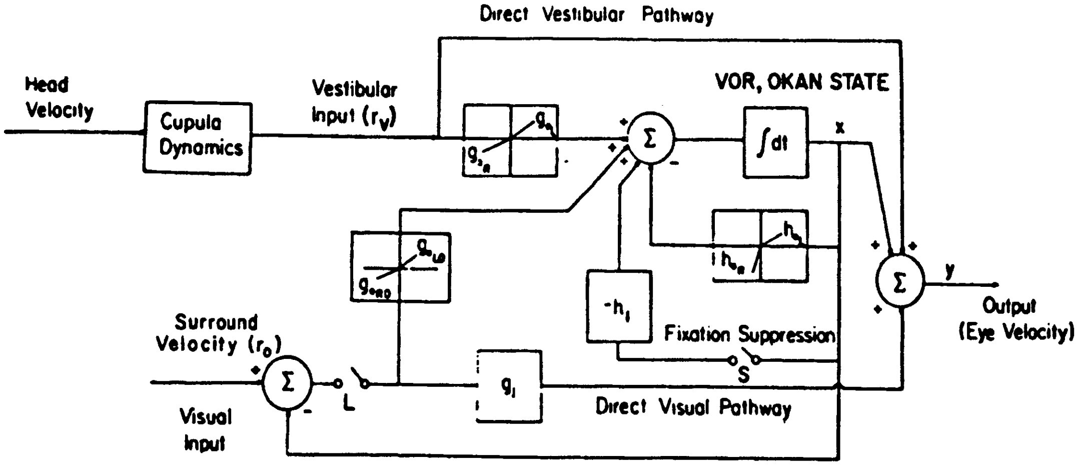
ROBINSON'S MODEL



From: Robinson (1977)

Figure 4.4

RAPHAN/COHEN MODEL



From: Raphan, Matsuo, and Cohen (1979)

Figure 4.5

force). By input-output standards, this model successfully models a number of experimental observations, but it doesn't provide insight into the mechanism by which the otoliths control the critical parameters or an explanation for the observations.

This multi-dimensional model has also been extended (Sturm and Raphan, 1988). The new model proposes that the eigenvalues and eigenvectors of the nystagmic response are dependent upon the gravitational field. This model mathematically captures some of the observed transformations, but, at this point, provides little insight into causes or mechanisms.

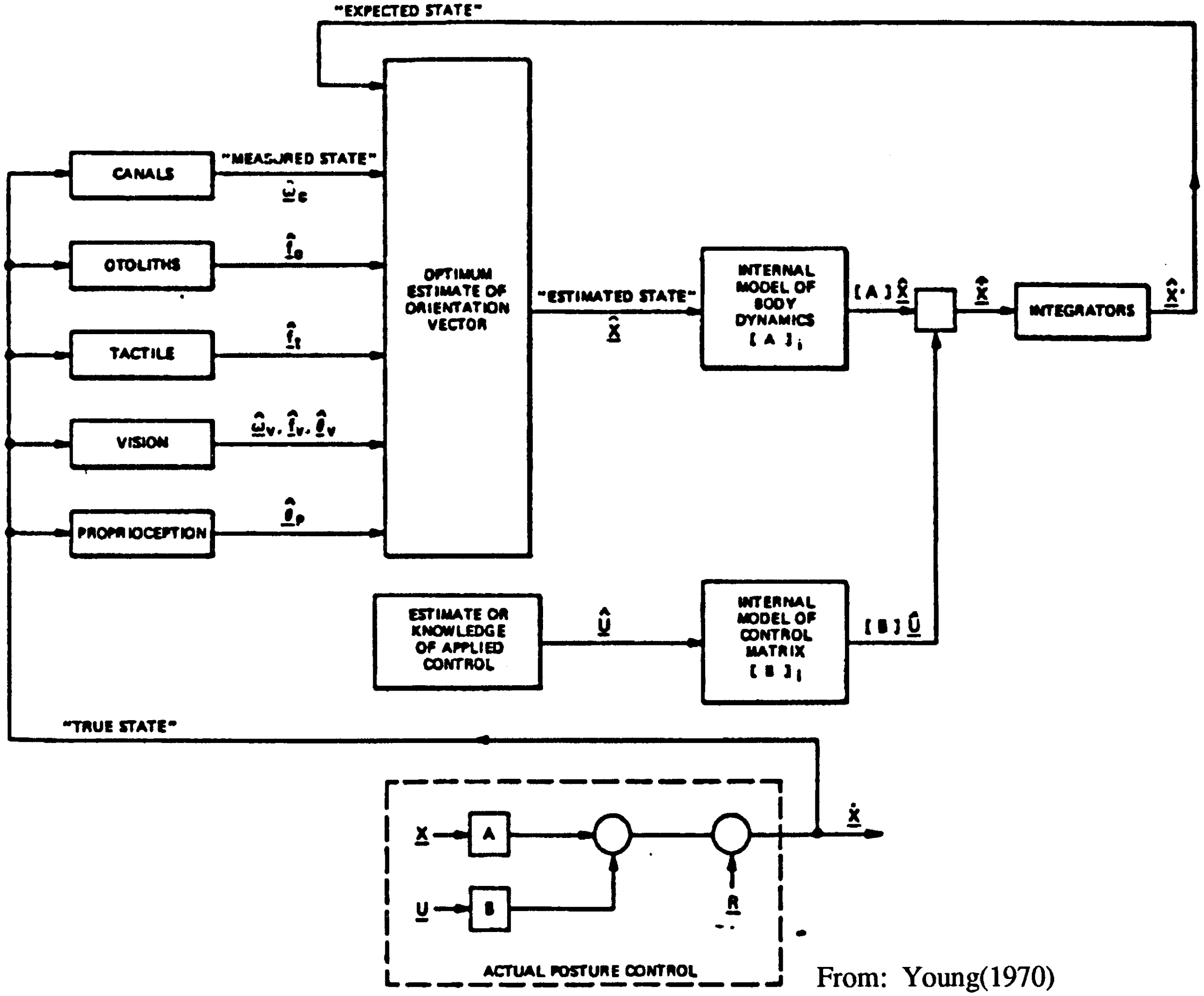
Robinson's feedback mechanism has also been included in a multidimensional model (Hain, 1986). In this model otolith signals modify particular feedback parameters such that the velocity storage time constants are reduced in the presence of tilt. In addition, the model passes otolith afference through the velocity storage mechanism. This model proposes a number of thought provoking hypotheses (e.g. LVOR is a residual effect), but it does not model most of the observed cross-axis transformations.

4.1.3 Modern Engineering System Models

Optimal control theory and optimal estimation theory have also been used in attempts to model spatial orientation. Young (1970) proposed that spatial orientation might be solved as an optimal mixing problem. (Figure 4.6.) In this model the estimated state is found by minimizing some unknown cost function. As Young described the model,

> "Consider first the sensors that are stimulated by the "true state" or actual motion of the subject. Each of the sensors measures one or more of the state variables with characteristic uncertainty and dynamic performance. For example, the semicircular canals measure the angular velocity of the subject with respect to inertial space, and their dynamic response may be described by the adaptation model. ... Similarly, the otoliths measure specific force; tactile sensation also measures specific force; and the visual channel measures angular velocity, direction of net specific force, and angular orientation.
>
> The outputs of all these sensors may be combined to make an optimum estimate of the true state."

YOUNG'S MODEL


"EXPECTED STATE"
CANALS
"MEASURED STATE"
OTOLITHS
TACTILE
VISION
PROPRIOCEPTION
OPTIMUM ESTIMATE OF ORIENTATION VECTOR
"ESTIMATED STATE"
INTERNAL MODEL OF BODY DYNAMICS [A]ᵢ
[A]
INTEGRATORS
ESTIMATE OR KNOWLEDGE OF APPLIED CONTROL
INTERNAL MODEL OF CONTROL MATRIX [B]ᵢ
[B]
"TRUE STATE"
X
A
U
B
R
ACTUAL POSTURE CONTROL

From: Young(1970)

Figure 4.6

In addition, Young recognized that expectations could also be included as part of the optimal mixing model. It had long been recognized by experimenters that a subject's expectations were very important to perceptions and reflexes. Young's model provided a natural mathematical method to model "expectations".

Zacharias (1977) and Zacharias and Young (1981) extended this model when they proposed a nonlinear cue blending model. This model proposed that each sensory signal is weighted dependent upon the perceived conflict between the senses.

Borah, Young, and Curry (1978) also explored optimal estimation when they implemented a model which could mimic a number of experimental findings. For example, by the proper choice of noise parameters, they were able to model the sensations associated with vection, vestibular velocity storage, and the gradual tilt perceived during sustained linear acceleration.

At approximately the same time as Young was proposing the optimal mixing problem, Kleinman, Boran, and Levinson (1970) developed an optimal control model which characterized the manual control capabilities of humans. Their quantitative model predicted human performance for a few simple compensatory tracking tasks.

Curry, Hoffman, and Young (1976) extended this manual control model by including representations of the dynamics of the semicircular canals and the otolith organs. The extended model was used to describe the performance of a pilot "flying" a light simulator and was validated by a number of experiments.

In yet another approach applying optimal estimation theory to spatial orientation, Ormsby (1974) and Ormsby and Young (1977) used estimation theory to calculate "optimal" estimates for each of the individual canal and otolith signals. These optimal estimates were then fed to a somewhat ad hoc schema to determine a state representing spatial orientation.

Oman (1982, 1988) formalized the optimal estimation approach toward spatial orientation while attempting to retain the spirit of the physiological models of Reason, Held, and Von Holst. Oman (1988) clearly attempted to mate the two different viewpoints by stating,

> "There is a direct analogy between the 'expected' feedback sensor measurement and 'internal dynamic model' concepts in control engineering Observer Theory, and the 'efference copy' and 'neural store' concepts which have emerged in physiology and psychology. From the perspective of control engineering, the 'orientation' brain must 'know' the natural behavior of the body, i.e. have an 'internal model' of the dynamics of the body, and maintain a continuous estimate of the spatial orientation of all of its parts. Incoming sensory inputs would be evaluated by subtraction of an "efference copy" signal, and the resulting 'sensory conflict' signal used to maintain a correct spatial orientation estimate."

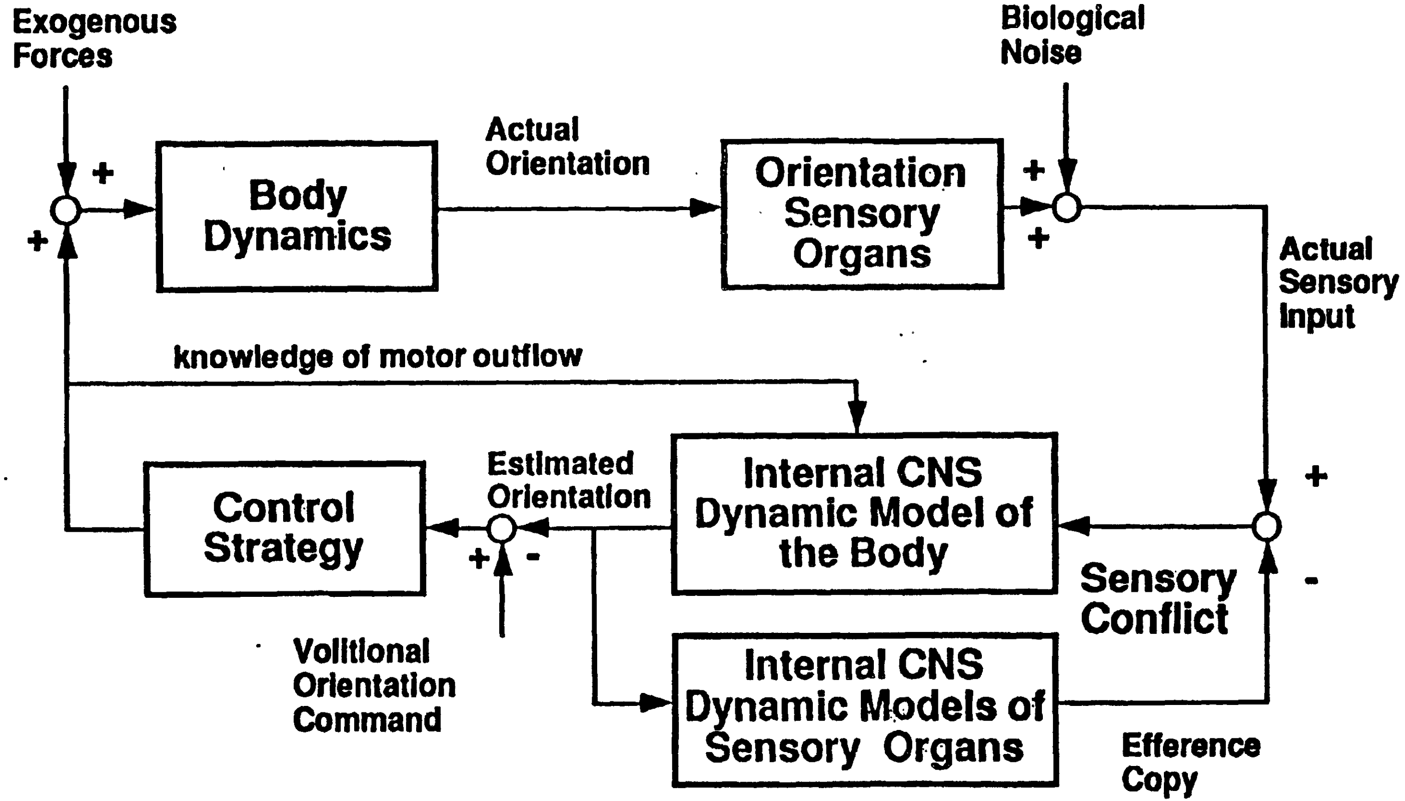
Oman posed the solution in terms of observer theory. In this formulation the observer contains a dynamic model of the controlled system and of the sensors (See Figure 4.7). These internal models calculate what the sensor measurements should be. Errors between the expected sensory afference and the actual sensory afference are used to drive the internal state towards reality.

By formulating the problem in terms of observer theory, Oman implicitly proposed a specific mathematical structure which underlies the block diagram shown in Figure 4.7. Explicitly, he attempted to represent the estimation process by a linearized set of differential equations. Figure 4.8 shows a block diagram of the linearized system represented in terms of state space notation.

4.2 Development

The approach which I present on the following pages is derived from the discussions in the background section. Specifically, I develop the internal model approach suggested by Young using the observer theory approach championed by Oman. I will begin developing the model using a very general non-mathematical description. I will then make a series of simplifying assumptions that will reduce the model to a 1-dimensional linear model. This simple version of the sensory conflict model will be investigated using two examples. I will then generalize until the model again includes 3 dimensions and some inherent nonlinearities.

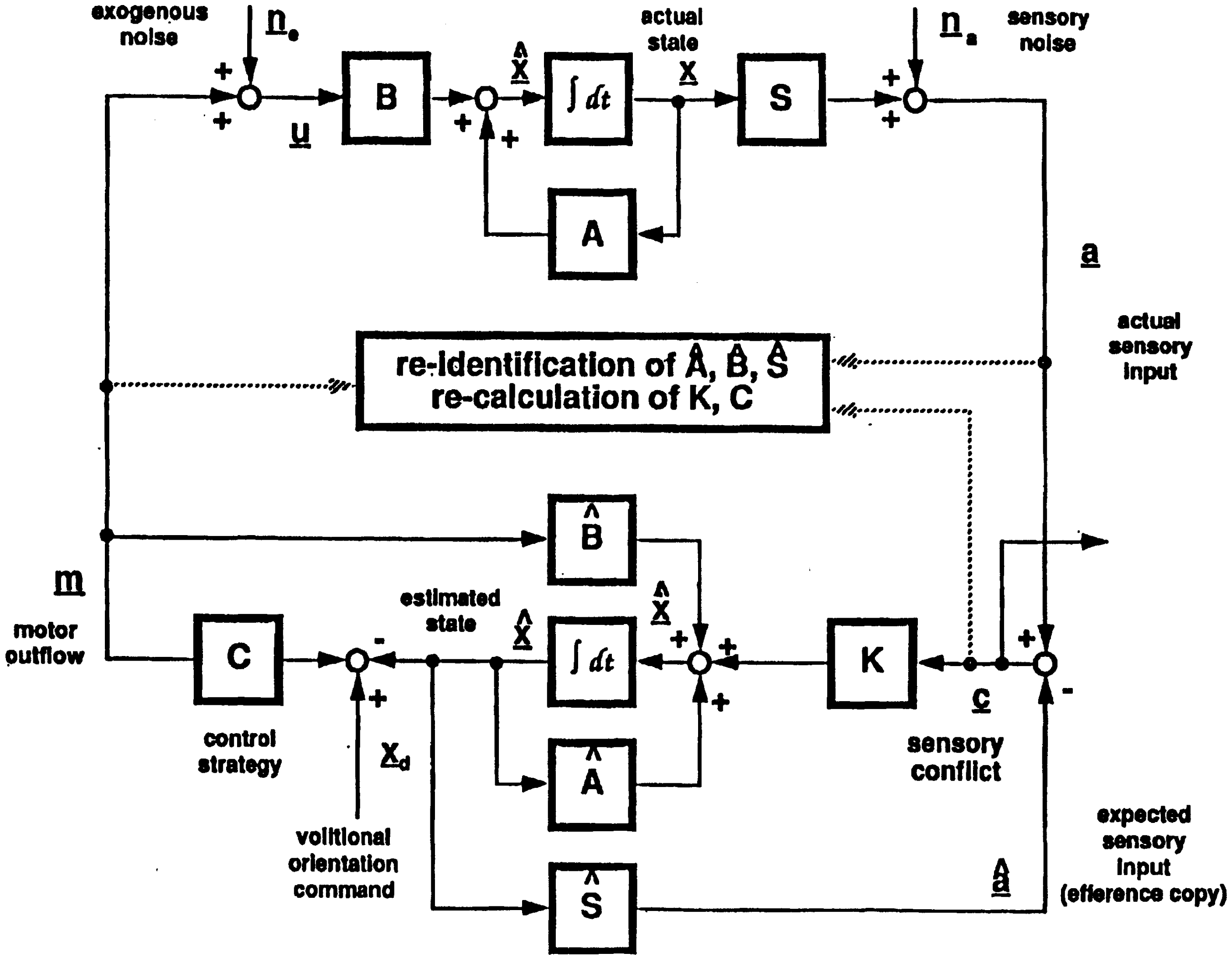
OMAN'S MODEL


Exogenous Forces
+
+
Body Dynamics
Actual Orientation
Orientation Sensory Organs
Biological Noise
+
+
Actual Sensory Input
knowledge of motor outflow
Control Strategy
Estimated Orientation
+
-
Internal CNS Dynamic Model of the Body
+
-
Sensory Conflict
Volitional Orientation Command
Internal CNS Dynamic Models of Sensory Organs
Efference Copy

From: Oman (1982)

Figure 4.7

OMAN'S LINEARIZED MODEL



From: Oman (1982)

Figure 4.8

I will use small letters to represent scalars ($\mathbf{k}$), small letters with arrows ($\vec{\mathbf{g}}$) to represent vectors, and capital letters ($\mathbf{K}$) to represent two dimensional matrices.

4.2.1 General Sensory Conflict Model

Figure 4.9 shows a very general block diagram which presents the philosophy underlying the development of the entire model. "Desired orientation", the primary system input, is compared to the "estimate of orientation" to yield an orientation error. A "control strategy" is applied to the orientation error to yield a "motor command." The motor command is relayed to the muscles which are represented as part of "body dynamics" to yield the "actual orientation." The actual orientation is measured by the sensory organs with the physiological output being sensory afference. A copy of the efferent signal ("efferent copy") is sent to an internal model of body dynamics (including muscle dynamics) to yield "estimated orientation." This estimate of orientation is sent to a model of sensory dynamics to give an "expected sensory afference." A difference between sensory afference and expected sensory afference indicates "sensory conflict." The sensory conflict returns to the internal model of body dynamics to drive the estimated orientation toward the true orientation.

Figure 4.10 shows the same model, but the model has been linearized, and state space notation is used. In this model, the desired orientation is represented by a vector whose elements represent the "states" of orientation. (Examples of possible states include linear velocity, angular velocity, tilt, etc.) The desired state vector ($\vec{\mathbf{x}}_\mathbf{d}$) is compared to the current state estimate ($\hat{\vec{\mathbf{x}}}$). The difference generates a motor command ($\vec{\mathbf{m}}$) via a control strategy ($\mathbf{M}$). The motor command is sent to the body dynamics ($\dot{\vec{\mathbf{x}}} = \mathbf{A}\,\vec{\mathbf{x}} + \mathbf{B}\,\vec{\mathbf{m}}$) to generate an actual state ($\vec{\mathbf{x}}$). This state is sensed via the sensory dynamics ($\dot{\vec{\mathbf{y}}}' = \mathbf{H}\,\vec{\mathbf{y}}' + \mathbf{C}\,\vec{\mathbf{x}}$) and with the addition of sensory noise ($\vec{\mathbf{v}}$) yields the sensory afference ($\vec{\mathbf{y}} = \vec{\mathbf{y}}' + \vec{\mathbf{v}}$).

A copy of the motor command is also sent to the model of body dynamics ($\dot{\hat{\vec{\mathbf{x}}}} = \hat{\mathbf{A}}\,\hat{\vec{\mathbf{x}}} + \hat{\mathbf{B}}\,\vec{\mathbf{m}}$). The current estimate of orientation ($\hat{\vec{\mathbf{x}}}$) is then sent to the model of the

SENSORY CONFLICT MODEL

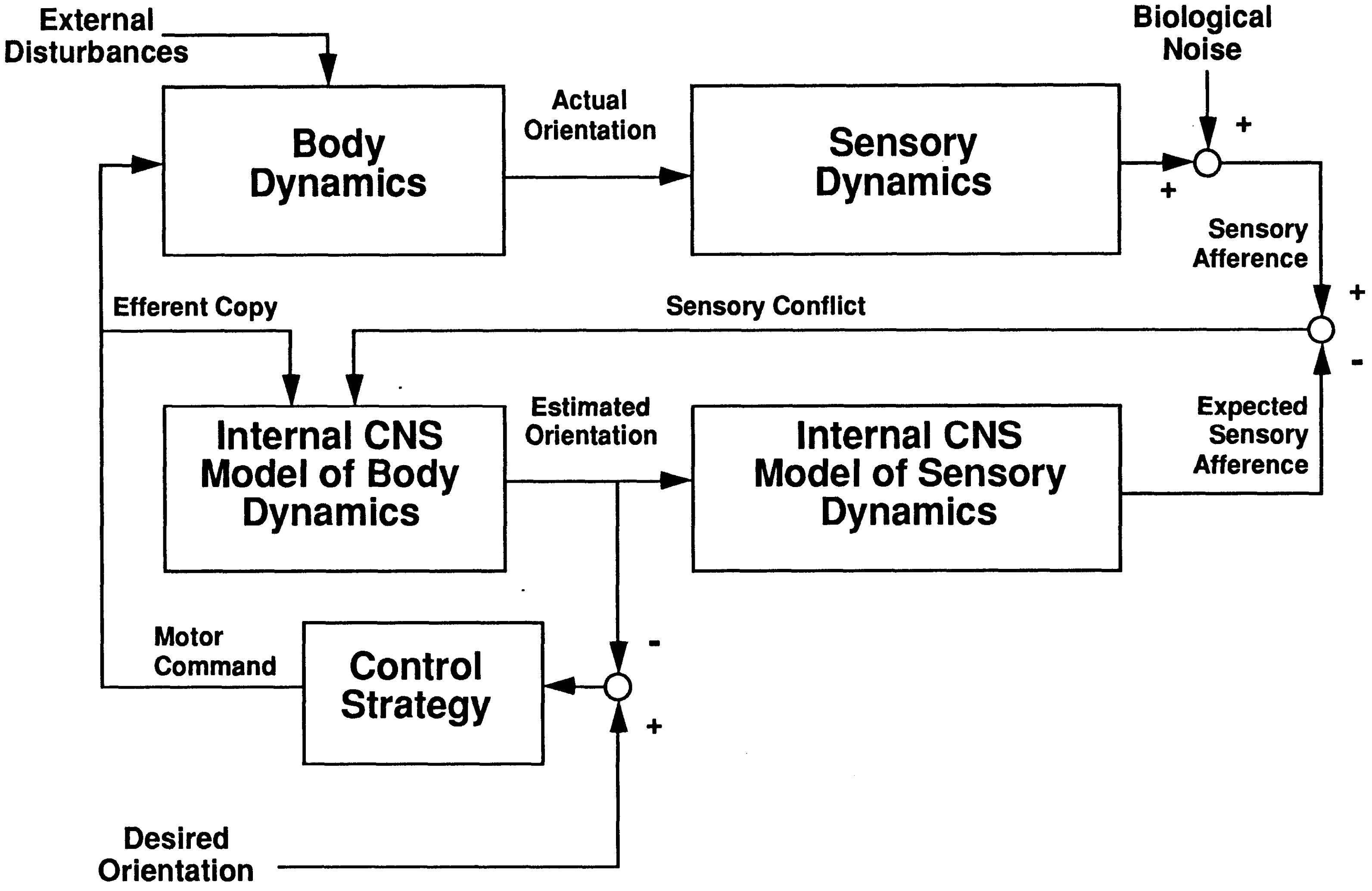


Figure 4.9

LINEARIZED SENSORY CONFLICT MODEL

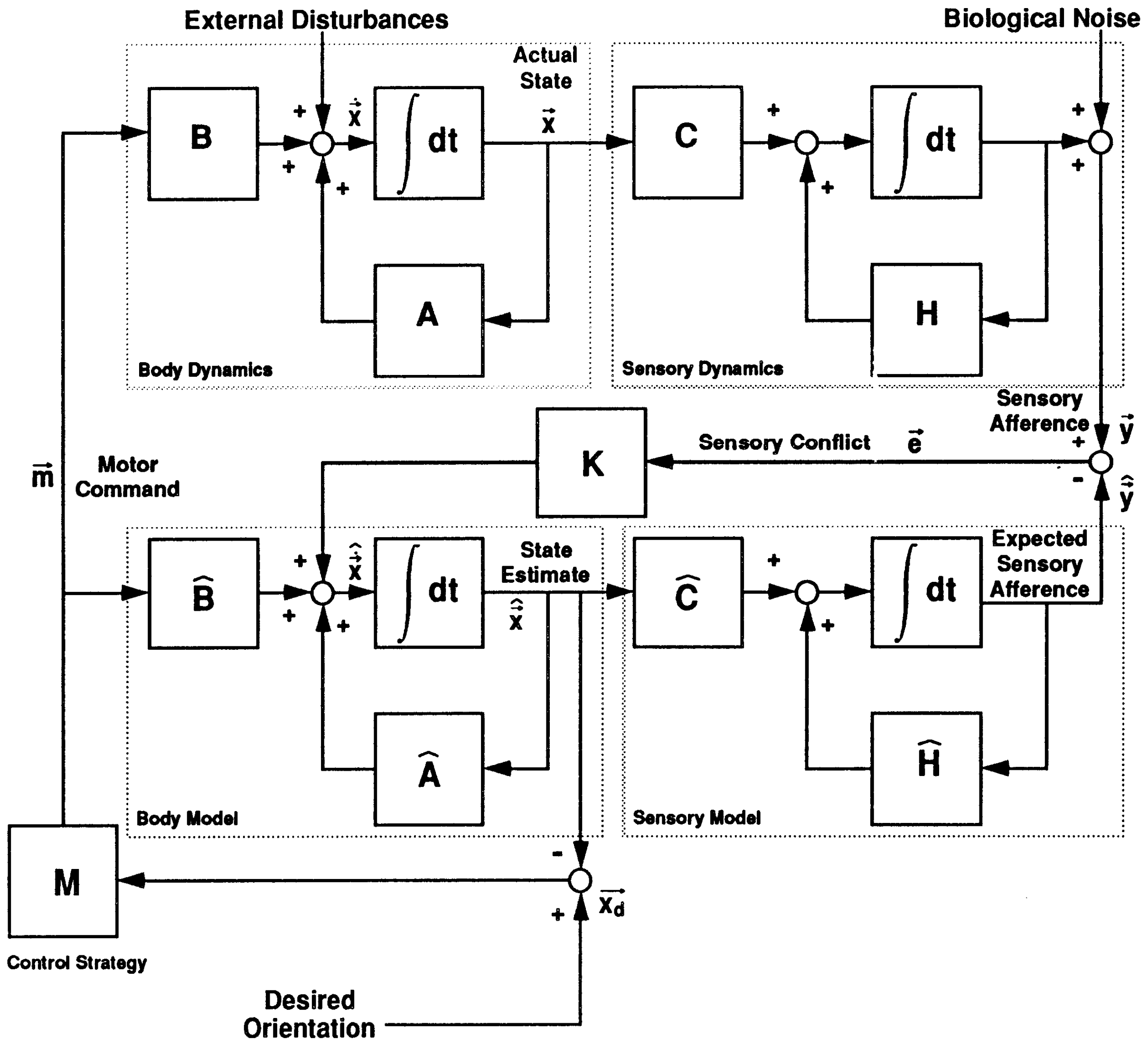


Figure 4.10

sensory dynamics ($\dot{\hat{\vec{y}}} = \hat{\mathbf{H}}\, \hat{\vec{y}} + \hat{\mathbf{C}}\, \hat{\vec{x}}$) to yield the expected afference ($\hat{\vec{y}}$). Sensory conflict ($\vec{e}$) is represented by the difference between the sensory afference and the expected sensory afference. The sensory conflict is returned as an input to the internal model of body dynamics through the Kalman-Bucy gain matrix (K) to drive the estimated state toward the true system state.

[Figures 4.9 and 4.10 are nearly identical to those presented by Oman. The major difference is merely cosmetic. The blocks have been rearranged to emphasize that the internal dynamic models, shown on the bottom half of the figure, are similar in form to the actual dynamics, shown on the top half.]

If the external disturbances are represented by $\vec{w}$, the entire model may be represented by the following set of equations:

$$\vec{m} = M\,(\vec{x}_d - \hat{\vec{x}})\ ,$$

$$\dot{\vec{x}} = A\,\vec{x} + B\,\vec{m} + \vec{w}\ ,$$

$$\dot{\hat{\vec{x}}} = \hat{A}\,\hat{\vec{x}} + \hat{B}\,\hat{\vec{m}} + K\,(\vec{y} - \hat{\vec{y}})\ ,$$

$$\dot{\vec{y}}' = H\,\vec{y}' + C\,\vec{x} \qquad (\vec{y} = \vec{y}' + \vec{v})\ ,$$

$$\dot{\hat{\vec{y}}}' = \hat{H}\,\hat{\vec{y}}' + \hat{C}\,\hat{\vec{x}}\ .$$

To simplify the model I will only investigate passive motion with this structure. This simple assumption, which applies to a majority of the literature, greatly simplifies the estimation process by removing the feedback loop dealing with motor control and control of orientation. Figures 4.11 and 4.12 show the block diagram for this simplified model. This simplified model applies only to the passive condition and is similar to the model shown in figures 4.9 and 4.10 except for removal of the control feedback loop. The only remaining system input is the external disturbance. This input corresponds to the passively controlled orientation of the subject.

SENSORY CONFLICT MODEL (w/o Voluntary Control)

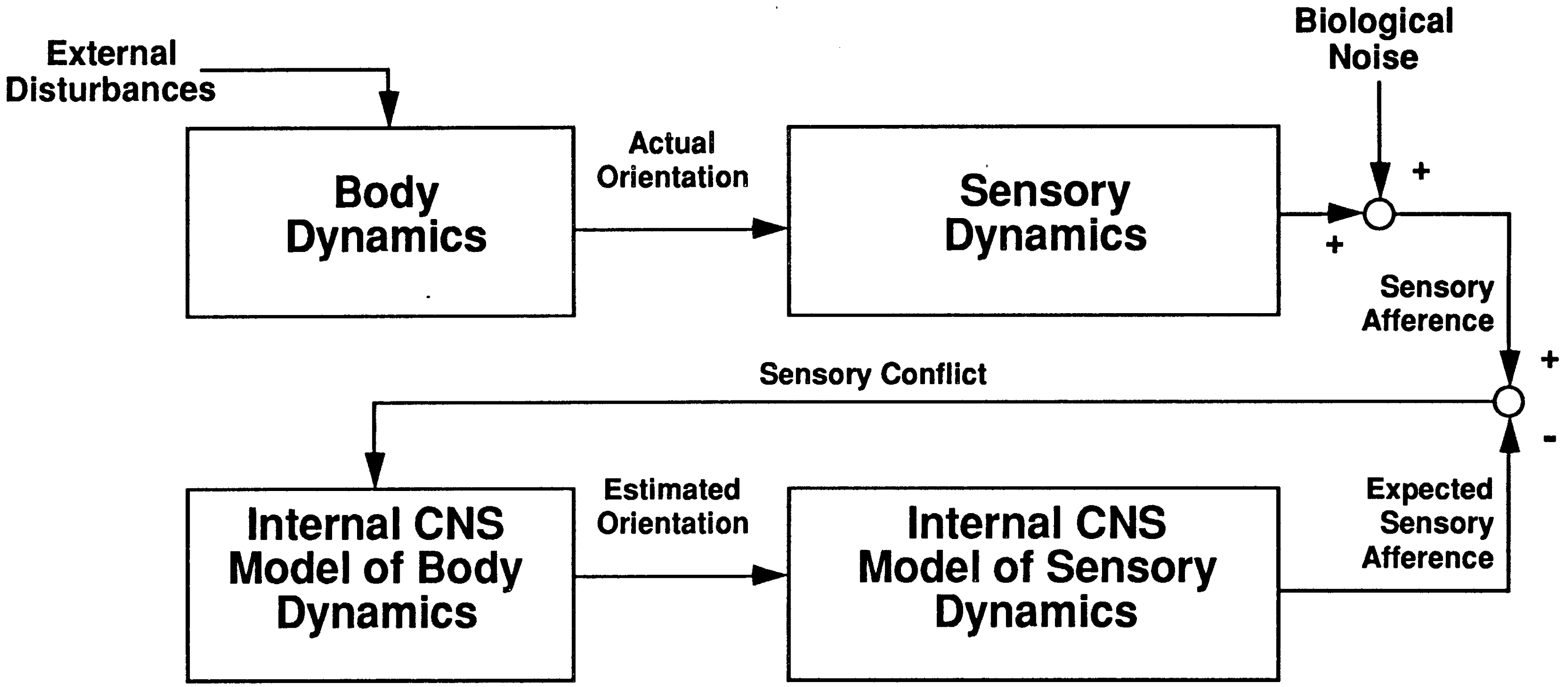


Figure 4.11

LINEARIZED SENSORY CONFLICT MODEL (w/o Voluntary Control)

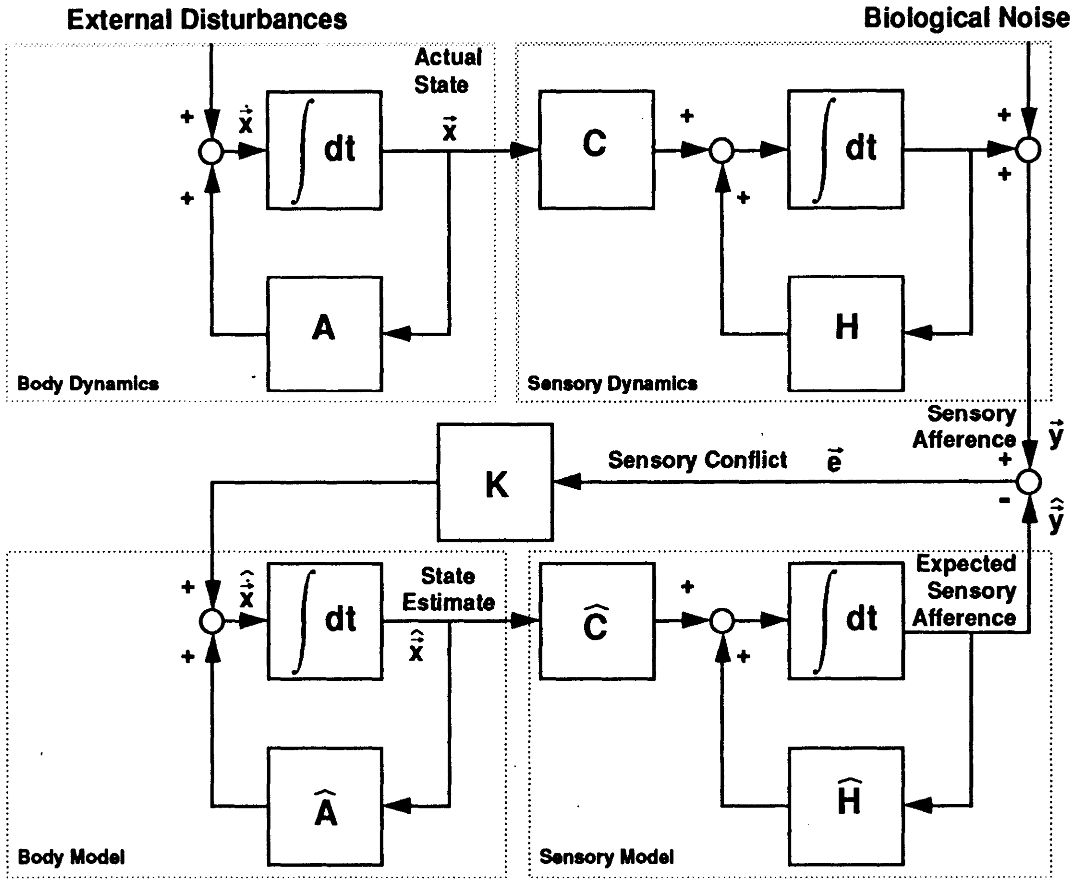


Figure 4.12

Most experiments which measure eye movements and/or the perception of orientation are designed to be passive. The subject or specimen is generally strapped into a chair or some other experimental device, and the orientation of the subject is controlled. Therefore, this simplified control model can be used to represent a majority of the experiments described in the literature, including all those which don't involve self control of spatial orientation.

With this assumption, the model can be represented by a simplified set of equations:

$$\dot{\vec{x}} = A\,\vec{x} + \vec{w}\ ,$$

$$\dot{\hat{\vec{x}}} = \hat{A}\,\hat{\vec{x}} + K\,(\vec{y} - \hat{\vec{y}})\ ,$$

$$\dot{\vec{y}}' = H\,\vec{y}' + C\,\vec{x} \qquad (\vec{y} = \vec{y}' + \vec{v})\ ,$$

$$\dot{\hat{\vec{y}}}' = \hat{H}\,\hat{\vec{y}}' + \hat{C}\,\hat{\vec{x}}.$$

The linear system shown in figure 4.12 may also be represented in terms of matrix transfer functions as shown in figure 4.13. The matrix transfer functions are defined by the equations:

$$D(s) = (s\,I - A)^{-1}\ ,$$

$$\hat{D}(s) = (s\,I - \hat{A})^{-1}\ ,$$

$$S(s) = C\,(s\,I - H)^{-1}\ ,$$

$$\hat{S}(s) = \hat{C}\,(s\,I - \hat{H})^{-1}\ .$$

The system equations may then be written as:

$$\vec{x}(s) = D(s)\,\vec{w}(s)\ ,$$

$$\hat{\vec{x}}(s) = \hat{D}(s)\,K\,(\vec{y}(s) - \hat{\vec{y}}(s))\ ,$$

$$\vec{y}'(s) = S(s)\,\vec{x}(s) \qquad \vec{y}(s) = \vec{y}'(s) + \vec{v}(s)$$

$$\hat{\vec{y}}'(s) = \hat{S}(s)\,\hat{\vec{x}}(s)\ .$$

MATRIX TRANSFER FUNCTION REPRESENTATION

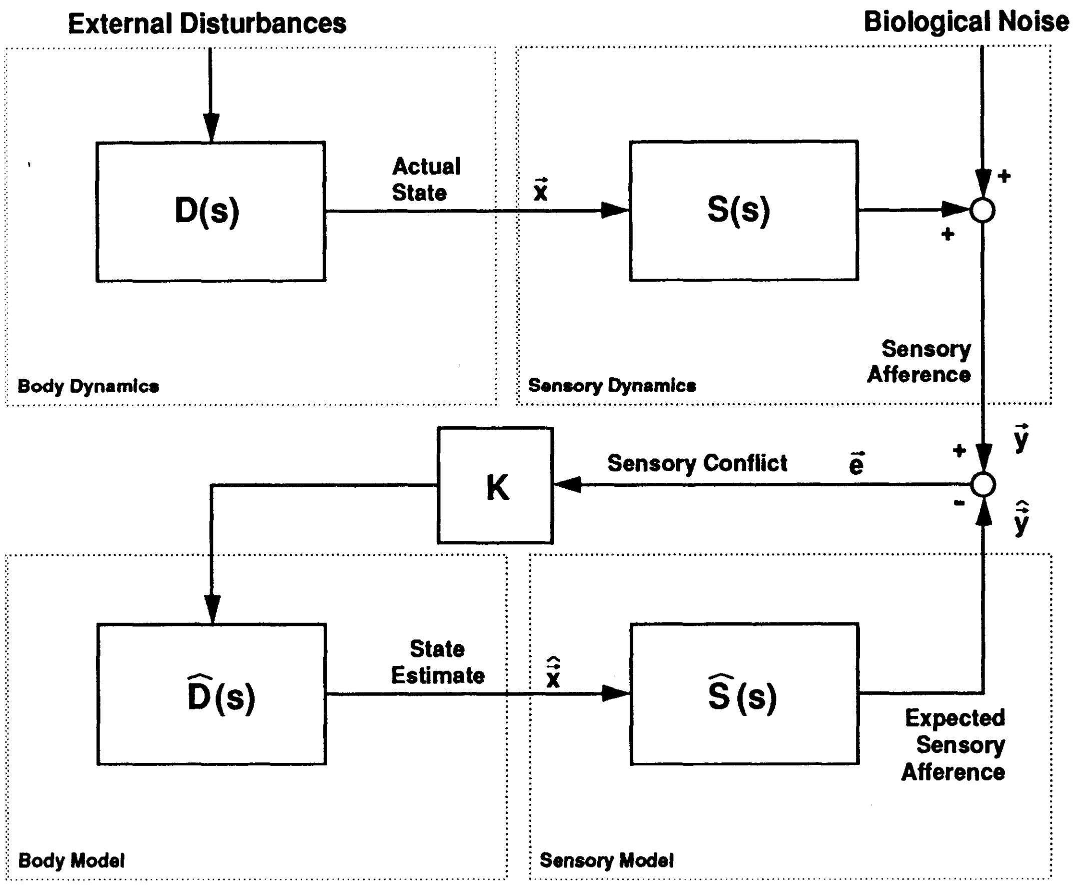


Figure 4.13

4.2.2 One-dimensional Sensory Conflict Model

In Figure 4.14 the model is simplified by developing a one dimensional linear observer. The matrix transfer functions of Figure 4.13 have been reduced to a single transfer function. Furthermore, by choosing the actual state (ω_z) to be identical to the disturbance (also ω_z), the body dynamics and the internal model of body dynamics simplify to unity.

As an example, assume that the sensory dynamics (scc(s)) may be represented by a high pass filter with a cut-off frequency equal to the inverse of the dominant time constant of the sensory afference:

$$\mathrm{scc}(s) = \frac{\tau s}{\tau s + 1} .$$

Also assume that the internal model of sensory dynamics has a similar form. Therefore:

$$\widehat{\mathrm{scc}}(s) = \frac{\hat{\tau} s}{\hat{\tau} s + 1} .$$

Figure 4.15 show a block diagram representing this model.

Using algebra, we can easily find the transfer function between the internal estimate of angular velocity ($\hat{\omega}_z$) and the actual angular velocity (ω_z). The transfer function has the form:

$$\frac{\hat{\omega}(s)}{\omega(s)} = \frac{k \tau s (\hat{\tau} s + 1)}{((k+1) \hat{\tau} s + 1)(\tau s + 1)} .$$

Goldberg and Fernandez (1971b) estimated that the dominant time constant of the semicircular canals was 5.7 seconds. If we model the afferent response with a time constant of 5.7 seconds and also model the internal model with a time constant of 5.7 seconds, a pole zero cancellation is obtained. This yields the transfer function:

$$\frac{\hat{\omega}(s)}{\omega(s)} = \frac{k \tau s}{((k+1)\tau s + 1)} .$$

ONE DIMENSIONAL LINEAR MODEL

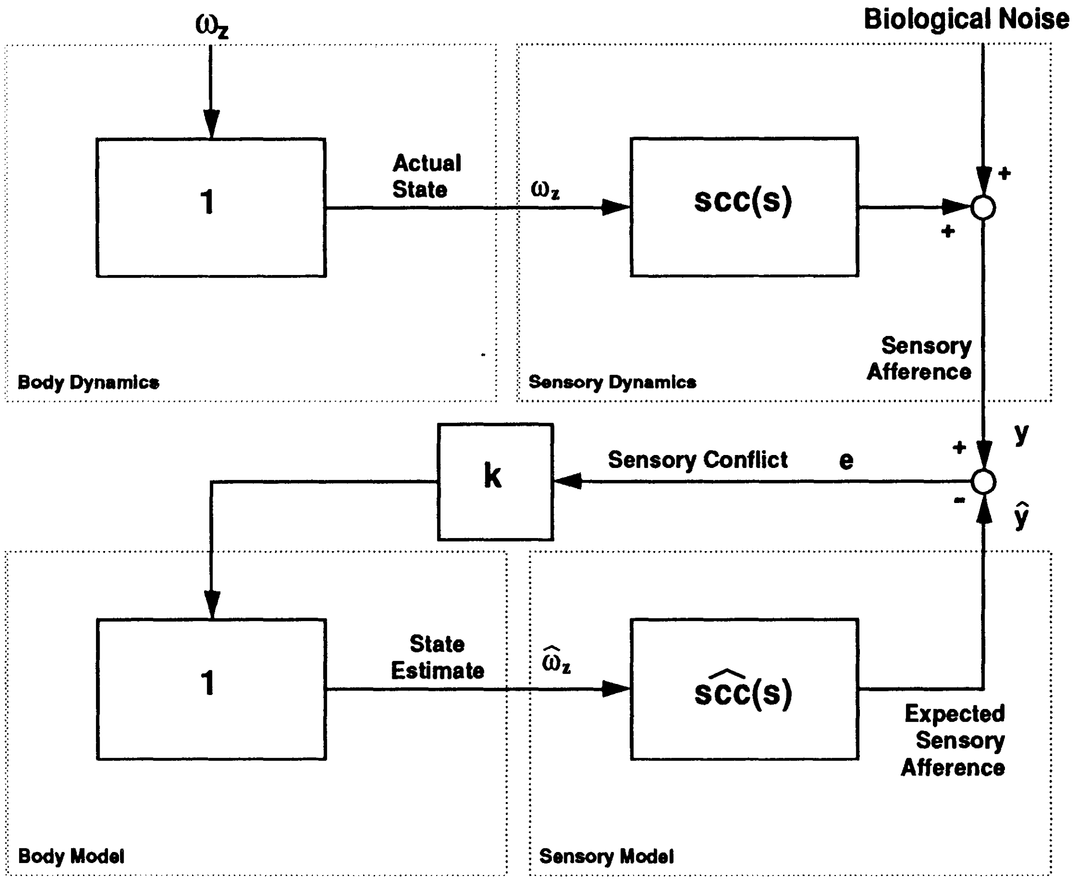


Figure 4.14

ONE DIMENSIONAL VELOCITY STORAGE MODEL

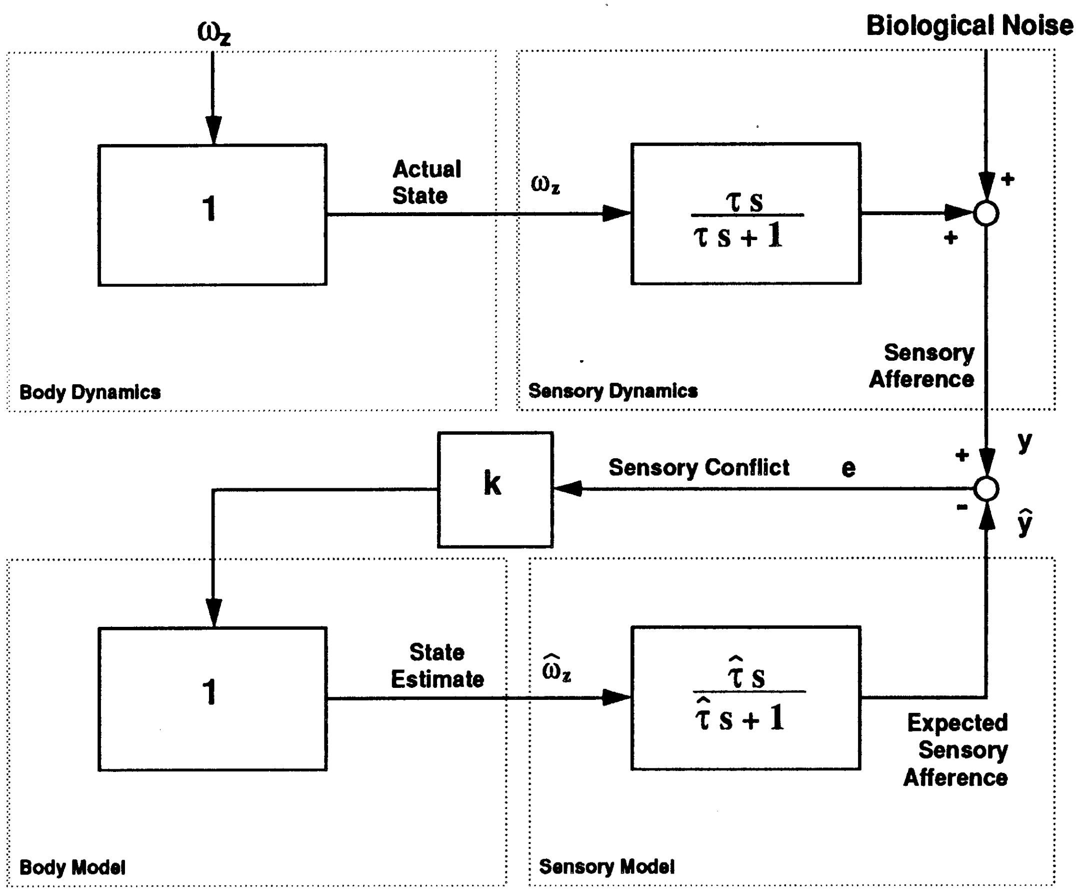


Figure 4.15

Figure 4.16a shows the response of this system to a trapezoidal step in angular velocity. The response is observed to have a dominant time constant of:

$$\tau' = (k+1)\tau \text{ s} .$$

This model exhibits the same velocity storage characteristic which has been exhibited by other models (Robinson, 1977, Raphan et al. 1977). In this case the velocity storage is accomplished by having an internal model of the sensory dynamics as part of a negative feedback loop.

This observer model of velocity storage is robust to relatively large errors in the sensory model parameters. Figures 4.16b and 4.16c show the model response for 20% errors in the model estimate of the dominant sensory time constant. The data illustrates that relatively large errors in model parameters have relatively small effects on the overall system response.

The data of Goldberg and Fernandez (1971b) will be used as an aid in choosing the sensory dynamics. They measured the first-order semicircular afferents in the squirrel monkey and deduced an average transfer function of:

$$\frac{y(s)}{\omega(s)} = \frac{80\,s^2}{1 + 80\text{ s}} \frac{1 + .049\text{ s}}{(1 + 5.7\text{ s})(1 + .003\text{ s})} .$$

This transfer function was estimated using data which spanned a frequency range between 0.0125 Hz and 8 Hz.

[Note: The transfer function originally deduced by Goldberg &Fernandez was with respect to acceleration whereas this transfer function is with respect to velocity. This difference results in an additional s in the numerator.]

There are some problems with this curve fit. First, the model predicts that the system response will increase as the frequency is increased from 3.25 Hz [$f = 1/(2\ \pi\ 0.049)$] to 50 Hz. Second, the model predicts that the system response will be constant for all frequencies greater than approximately 50 Hz [$f = 1/(2\ \pi\ 0.003)$]. Neither

"VELOCITY STORAGE" - MODEL PREDICTIONS

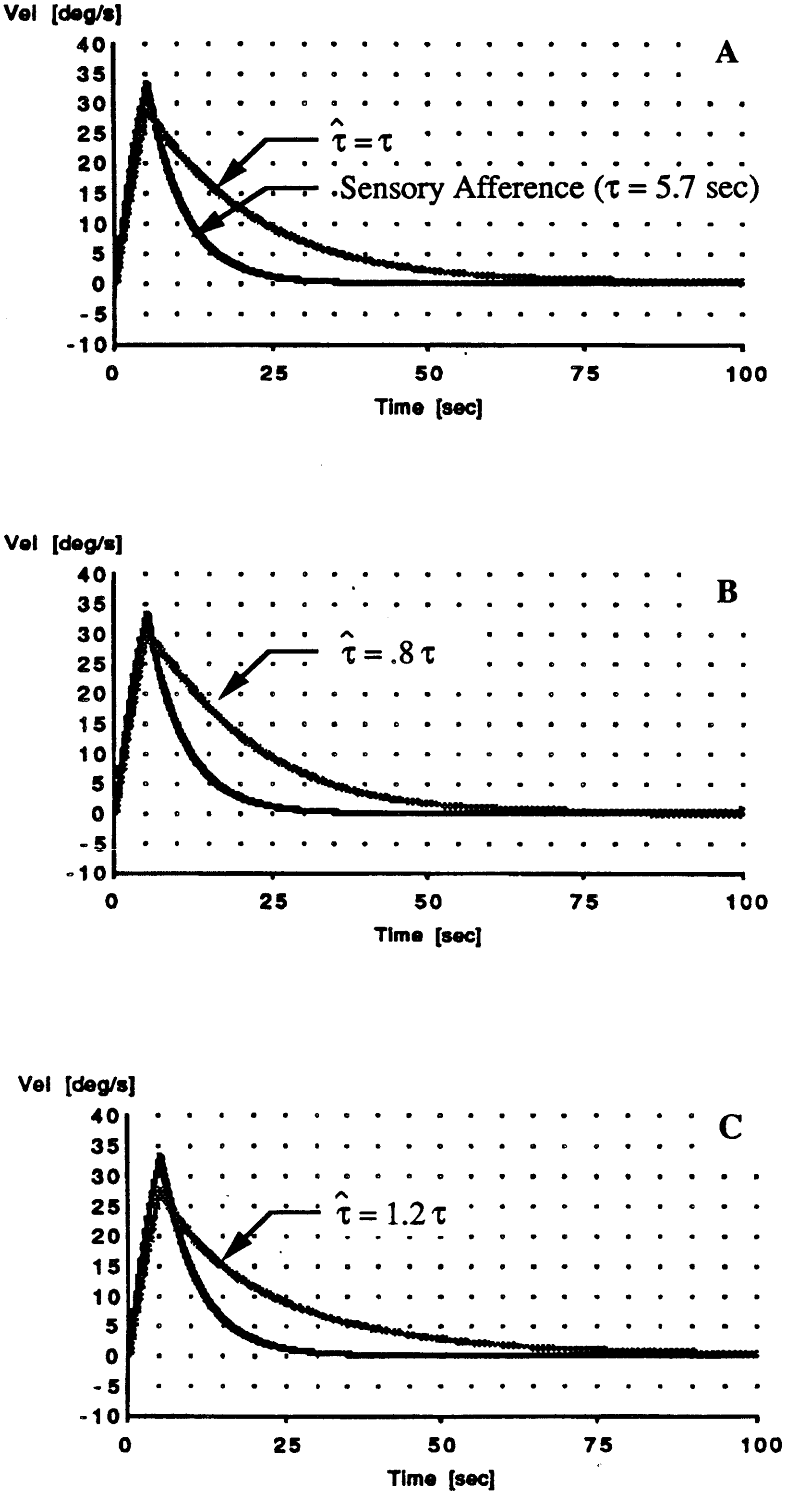


Figure 4.16

of these predictions is likely. Therefore; a modified transfer function will be implemented to represent the semicircular canals.

[Note: The probable cause of the error is that the frequency range of the data being fit was limited, but the parameters were not limited in any way. (e.g. The maximum frequency tested was 8 Hz, but a pole was fit with a frequency of 50 Hz.)]

For purposes of simplicity and accuracy, all high frequency effects will be ignored, and the model inputs will be limited to low frequency disturbances (< 1 Hz). With this limitation, the semicircular canal may be represented with a transfer function of the form:

$$\frac{y(s)}{\omega(s)} = scc(s) = \frac{\tau_a \tau_l s^2}{(1 + \tau_a s)(1 + \tau_l s)} ,$$

where,

$$\tau_a = 80 \text{ s} \qquad \text{and} \qquad \tau_l = 5.7 \text{ s} .$$

This form of the transfer function still suffers from one of the problems with the Fernandez and Goldberg model. It predicts a constant response beyond approximately 0.03 Hz, but the response will never increase as the frequency exceeds 0.03 Hz.

For the above stated reasons, I have chosen to represent the dynamics of the semicircular canals with the simplified transfer function shown above. However, this simple model may not be used to make predictions beyond 1 Hz.

I will try to choose the simplest form of the internal model of sensory dynamics as possible. There are a number of reasons for this approach. Most importantly, the internal knowledge of sensory dynamics will predominantly be a knowledge of the dominant characteristics of the sensors. It is unreasonable to suspect that the internal system will have a detailed knowledge of all aspects of the sensory dynamics. Second, the additional parameters associated with more complicated forms of the internal model will, of course, allow better fits of the data. But the goal of this model is not to fit the data, but rather to provide a framework with which predictions might be made and experiments suggested.

More parameters would confuse the situation by allowing better (in a least mean squares sense), but not necessarily more realistic fits.

For these reasons I choose to greatly simplify the internal model of sensory dynamics by assuming the form:

$$\widehat{scc}(s) = \frac{\tau_l\, s}{(1 + \tau_l\, s)}\,,$$

with

$$\tau_l = 5.7 \text{ s}\,.$$

With this model of the internal sensory dynamics and the model for the actual sensory dynamics presented on the previous page, we can easily determine the transfer function relating the internal state estimate of angular velocity ($\hat{\vec{\omega}}$) to the actual angular velocity ($\vec{\omega}$):

$$\frac{\hat{\omega}(s)}{\omega(s)} = \frac{K\, \tau_a\, \tau_l\, s^2\, (\hat{\tau}_l\, s + 1)}{(\tau_l\, s + 1)\, (\tau_a\, s + 1)\, ((k+1)\hat{\tau}_l\, s + 1)}\,.$$

With,

$$\hat{\tau}_l = \tau_l\,,$$

the transfer function reduces to:

$$\frac{\hat{\omega}(s)}{\omega(s)} = \frac{K\, \tau_a\, \tau_l\, s^2}{(\tau_a\, s + 1)\, ((k+1)\hat{\tau}_l\, s + 1)}\,.$$

The system response of this velocity storage model is shown in Figure 4.17.

4.2.3 Multi-dimensional Sensory Conflict Model

I will extend this model to a three dimensional representation by replacing all scalar values ($\boldsymbol{\omega_z}$, $\boldsymbol{\hat{\omega}_z}$, **y**, $\boldsymbol{\hat{y}}$, and **e**) with vectors having three components ($\vec{\omega}$, $\hat{\vec{\omega}}$, $\vec{y}$, $\hat{\vec{y}}$, and $\vec{e}$), by replacing the unity operators of the scalar model with 3x3 identity matrices (I), and by replacing the transfer function of the semicircular canal with a 3x3 matrix transfer function. (Figure 4.18 shows a block diagram representation.) For simplicity, let:

$$S_{scc}(s) = \begin{bmatrix} scc(s) & 0 & 0 \\ 0 & scc(s) & 0 \\ 0 & 0 & scc(s) \end{bmatrix},$$

"VELOCITY STORAGE WITH ADAPTATION" - MODEL PREDICTIONS

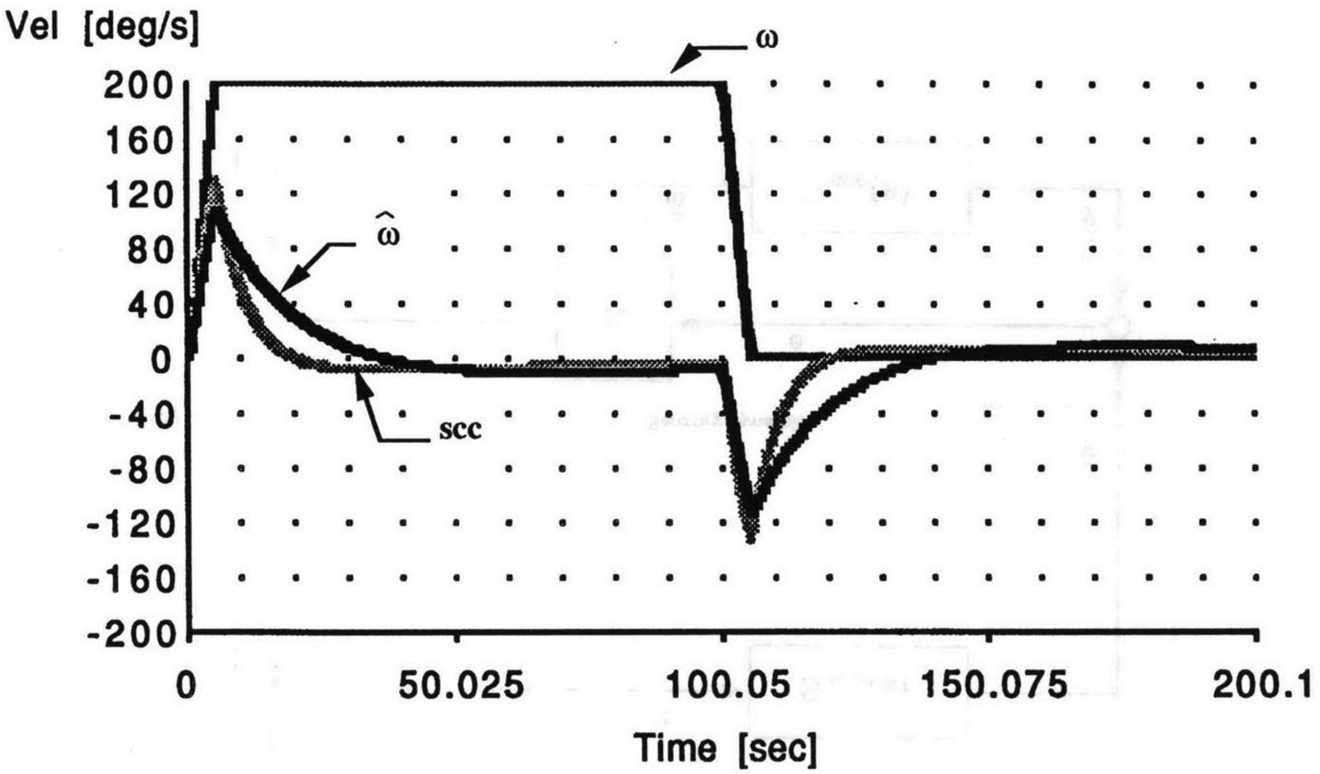


Figure 4.17

THREE DIMENSIONAL VELOCITY STORAGE MODEL

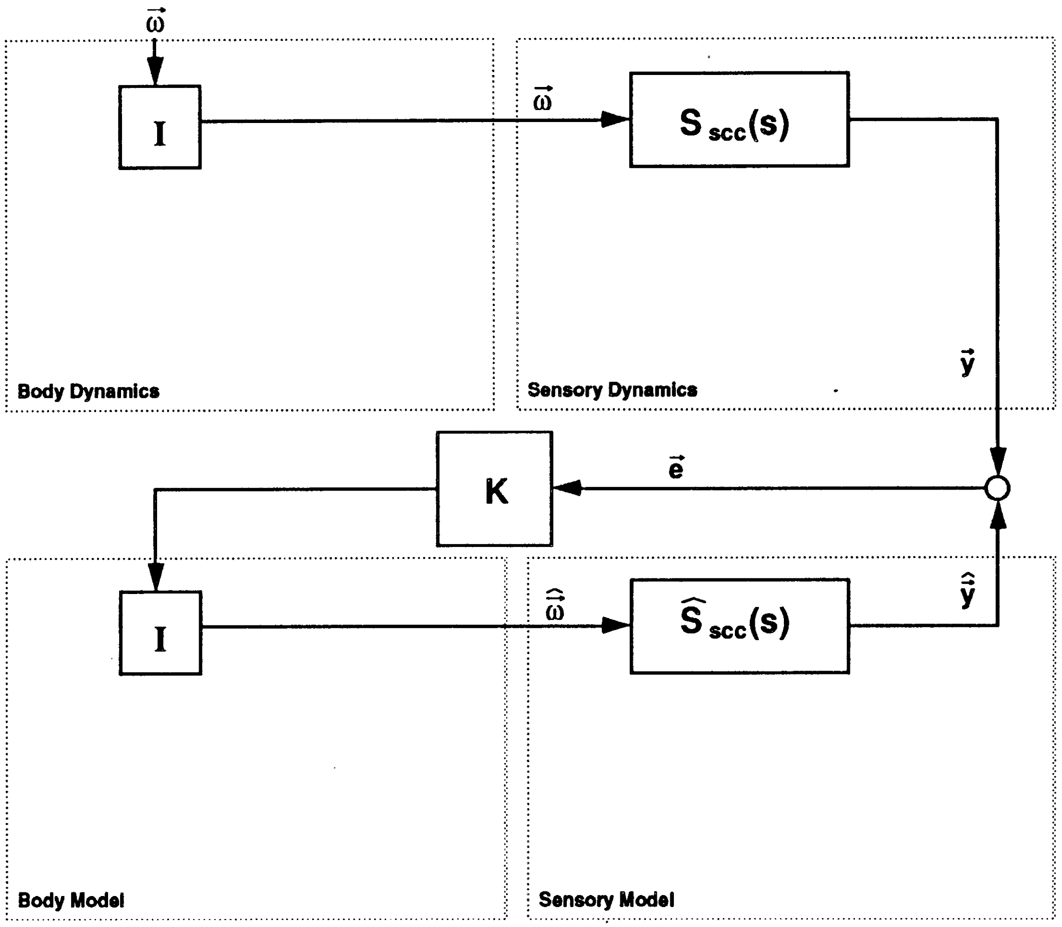


Figure 4.18

with,

$$scc(s) = \frac{\tau_a \tau_l s^2}{(1 + \tau_a s)(1 + \tau_l s)} .$$

In a similar manner:

$$\hat{S}_{scc}(s) = \begin{bmatrix} \widehat{scc}(s) & 0 & 0 \\ 0 & \widehat{scc}(s) & 0 \\ 0 & 0 & \widehat{scc}(s) \end{bmatrix} ,$$

with,

$$\widehat{scc}(s) = \frac{\tau_l s}{(1 + \tau_l s)} .$$

Through use of this diagonal matrix transfer function, it is implicitly assumed that the semicircular canals are mutually orthogonal and that they are aligned with the axes of the coordinate system. As a first approximation, the canals may be treated as orthogonal, and the coordinate system may be chosen to align with that defined by the semicircular canals. Anatomical accuracy may easily be provided by changing the transfer function matrix from the diagonal form shown above to one which truly represents the geometry of the semicircular canals. [See Robinson (1982) for further details.] The feedback gain matrix (K) will require analogous changes to represent the change in the sensory matrix. From an input/output perspective, however, these additions are transparent, and therefore were avoided. Since simplicity is gained without any cost in terms of performance, I have chosen the simplest possible representation. This will ease further model development and will allow more insight to be derived from the model.

Three dimensional rotation will, in general, constantly change the orientation of the gravitational force. If we know the current position of gravity ($\vec{g}_o$), and we have an imposed angular disturbance ($\vec{\omega}$), we can easily keep track of the orientation of gravity. This physical effect is represented as part of the body dynamics shown in Figure 4.19.

THREE DIMENSIONAL SENSORY CONFLICT MODEL

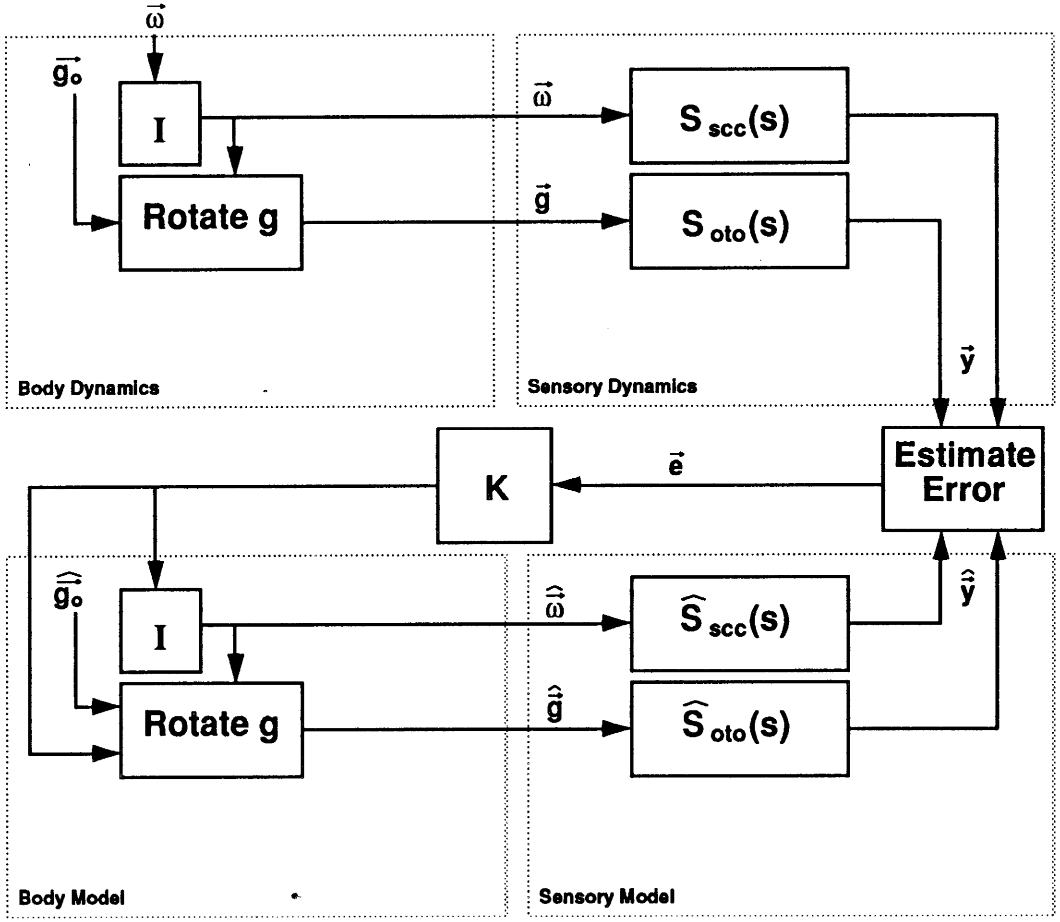


Figure 4.19

The block labeled "rotate g" performs these calculations. The actual calculation are implemented via a quaternion integrator.

Analogously, if there is a current internal estimate of gravity ($\hat{\vec{g}}_o$), and there is an internal estimate of angular velocity ($\hat{\vec{\omega}}$), we can keep track of an estimate of the orientation of gravity ($\hat{\vec{g}}$) through the same calculations as discussed above. The block "rotate g" is also used to perform these calculations. .

By arguments similar to those used when choosing the transfer function representing the semicircular canals, the otolith organs will be represented with a diagonal transfer function:

$$S_{oto}(s) = \begin{bmatrix} oto(s) & 0 & 0 \\ 0 & oto(s) & 0 \\ 0 & 0 & oto(s) \end{bmatrix}.$$

[This representation of the otolith organs assumes that the sensory afference from the two otolith organs, the utricle and the saccule, is integrated to yield a three dimensional representation of gravito-inertial force.]

Fernandez and Goldberg (1976 a,b, and c) investigated the dynamics response of first order otolith afferents. They found that the frequency response of the regular units is approximately constant from DC to approximately 2 Hz. Since inputs have already limited to less than 1 Hz, the otolith transfer function may be approximated as unity:

$$oto(s) = 1.$$

Therefore:

$$S_{oto}(s) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I \quad .$$

Data at higher frequencies would be required to determine a more accurate transfer function representation.

Analogously, I chose to represent the internal model of the otolith dynamics by the identity matrix:

$$\hat{S}_{oto}(s) = I\ .$$

The quaternion integration which keeps track of the orientation of gravity is, obviously, a nonlinear calculation. Therefore, error estimation should not be limited to subtraction as previously exhibited for linear systems.

Figure 4.20 shows the equations which determine the 6-component vector ($\vec{e}$). The linear portion of the process is estimated using subtraction:

$$\vec{e}_{\omega} = \vec{\omega} - \hat{\vec{\omega}}\ .$$

The estimation of the nonlinear error is a little more complicated. The magnitude of the error equals the angle between the specific force vector ($\vec{f}$) and the current internal estimate of gravity ($\hat{\vec{g}}$). Mathematically, this is written:

$$|\vec{e}_g| = \cos^{-1}\left(\frac{\vec{f} \cdot \hat{\vec{g}}}{|\vec{f}||\hat{\vec{g}}|}\right).$$

The direction of the gravitational error vector is given by the direction of the rotation needed to align the total gravito-inertial force ($\vec{f}$) with the internal estimate of gravity ($\hat{\vec{g}}$). Alternatively, we may consider this direction as the direction of the rotation which could yield the discrepancy between gravito-inertial force and the internal estimate of gravity. With either interpretation, this direction may be written as:

$$\frac{\vec{e}_g}{|\vec{e}_g|} = \frac{\vec{f} \times \hat{\vec{g}}}{|\vec{f} \times \hat{\vec{g}}|}\ .$$

The total error vector ($\vec{e}$) is formed by summing the the two error vectors such that the angular velocity error ($\vec{e}_{\omega}$) fills the first three components of the error vector, and the

Error Estimate

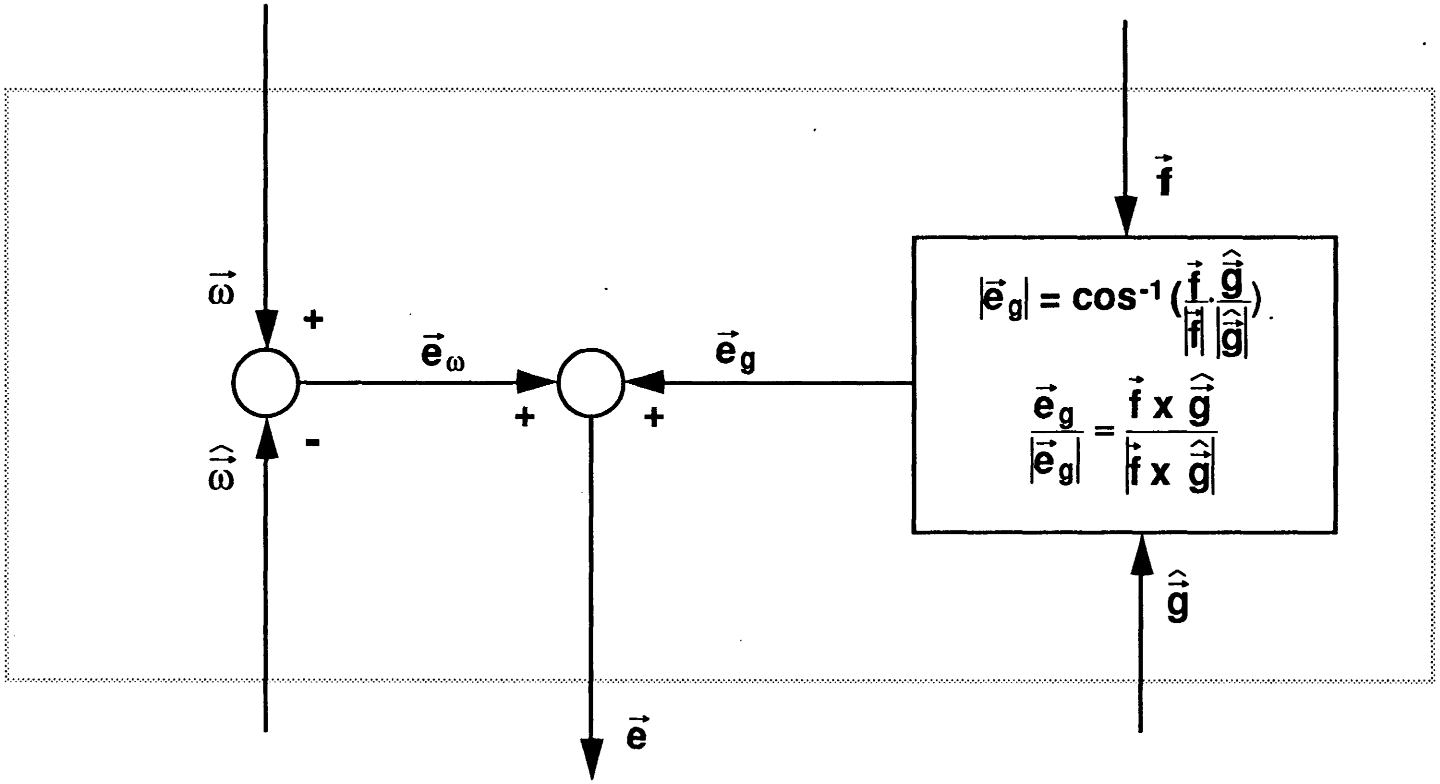

$\vec{\omega}$
+
$\hat{\vec{\omega}}$
-
$\vec{e}_\omega$
+
$\vec{e}_g$
+
$\vec{f}$
$|\vec{e}_g| = \cos^{-1}(\frac{\vec{f}}{|\vec{f}|} \cdot \frac{\hat{\vec{g}}}{|\hat{\vec{g}}|})$
$\frac{\vec{e}_g}{|\vec{e}_g|} = \frac{\vec{f} \times \hat{\vec{g}}}{|\vec{f} \times \hat{\vec{g}}|}$
$\hat{\vec{g}}$
$\vec{e}$

Figure 4.20

gravitational error ($\vec{e}_g$) completes the bottom three elements of the error vector. This may be written:

$$\vec{e} = \begin{bmatrix} \vec{e}_\omega \\ \varnothing \end{bmatrix} + \begin{bmatrix} \varnothing \\ \vec{e}_g \end{bmatrix} .$$

The 6 x 6 Kalman gain matrix (K) is chosen to have just nine non-zero elements as shown below:

$$K = \begin{bmatrix} k_{\omega_x} & 0 & 0 & & & \\ 0 & k_{\omega_y} & 0 & & \varnothing & \\ 0 & 0 & k_{\omega_z} & & & \\ k_{g\omega_x} & 0 & 0 & k_{g_x} & 0 & 0 \\ 0 & k_{g\omega_y} & 0 & 0 & k_{g_y} & 0 \\ 0 & 0 & k_{g\omega_z} & 0 & 0 & k_{g_z} \end{bmatrix} .$$

The angular velocity error feedback gains (k_{ω_x}, k_{ω_y}, and k_{ω_z}) are set such that the appropriate velocity storage is obtained for each independent axis of rotation. For simplicity, I have assumed that all of velocity feedback gains are equal and have the value of two [(deg/sec)/(deg/sec)]. By previous calculations this yields a dominant time constant of 17.1 seconds for each of the rotational responses. For the simulations which are shown on the following pages all of the gravity error gains (k_{g_x}, k_{g_y}, and k_{g_z}) are set equal to zero. Small values (.01 to 1 [(deg/sec)/deg] were also implemented in a number of trials and had only minor effects on the system dynamics. The remaining set of feedback gains ($k_{g\omega_x}$, $k_{g\omega_y}$, and $k_{g\omega_z}$) were all set to the relatively high value of 100 [(deg/sec)/deg]. This high gain was required to yield steady estimates of the rotation rate during constant velocity barbecue spit rotation and to yield rapid axis transformation effects.

From physics we know that gravito-inertial force is the vector difference formed by subtracting acceleration from gravity:

$$\vec{f} = \vec{g} - \vec{a} .$$

If an animal (or human) has an internal representation of gravity and a sensory measurement of gravito-inertial force, it is consistent that acceleration might be calculated as the vector difference between these quantities:

$$\hat{\vec{a}} = \hat{\vec{g}} - \vec{f} \, .$$

I will assume that this is the case. Evidence from the previous chapter supports this assumption.

4.3 Predictions

Five types of experiments have been simulated using this model. The "experimental" conditions include:

a) upright yaw in the dark
b) visual "dumping" of the post rotary VOR
c) barbecue spit rotation
d) OVAR at a 30 degree tilt from vertical
e) post rotary "dumping" of the VOR

The upright yaw response has already been investigated as part of the model development (Figure 4.17). As previously discussed, the model exhibits "velocity storage" as observed experimentally and as modelled by other researchers (Robinson, 1977; Raphan et al., 1977).

The remaining experimental conditions will be presented by showing the experimental data followed by the model predictions. A brief description of the experimental conditions and a summary of the results will be provided. Please see the cited works for the complete description of the experimental procedures and for the results. All of the model parameters are fixed at the values discussed previously.

4.3.1 "Visual Dumping" Experiment

The "visual dumping" experiment which will be modelled was first performed in the early 1960's (Guedry et al. 1961). Horizontal eye measurements were recorded using EOG techniques. In total darkness, subjects were accelerated to a constant angular velocity of 73.2 deg/s. This velocity was maintained for ninety seconds. During the next five seconds, the subject was decelerated to a stop. In some trials the subjects' post rotary responses were recorded in total darkness. In the other trials the lights were turned on for a period of five seconds and then extinguished.

The results of this study are shown in figure 4.21. Note that the experimental data show that the response decreases when the lights are turned on and the the response increases when the lights are turned off again. However, note that the response never fully recovers.

Since the model does not include visual effects the study was modelled by maintaining the angular velocity state at zero until after the subject reached a stop. After a predetermined time delay the state was released, simulating "lights off", and the responses recorded for various values of the time delay.

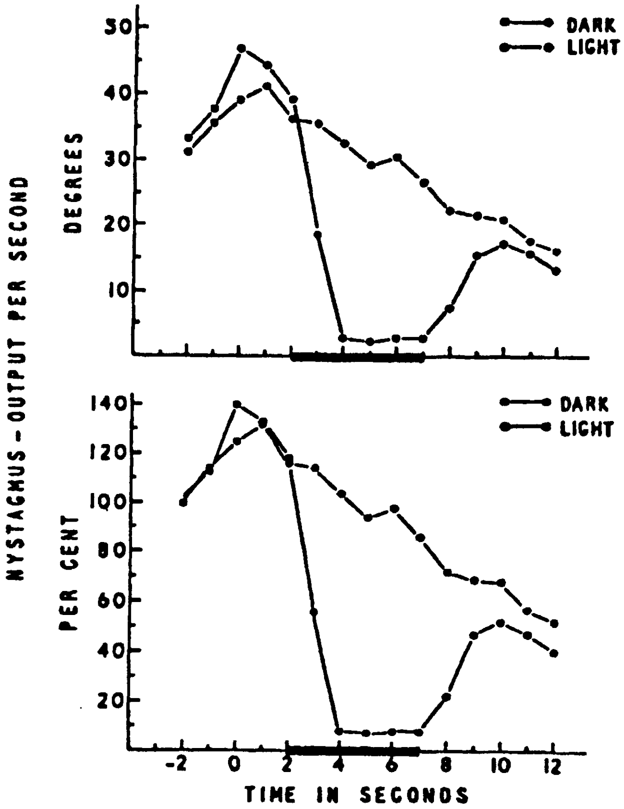
Figure 4.22 shows the modelled responses for a series of different time delays. Note that the responses for delay of 2.5 seconds and 7.5 seconds show a qualitative similarity to the experimental data. With time delays of 12.5 seconds and 17.5 seconds, the model shows a very small response, and interestingly, with a time delay of 17.5 seconds, the model predicts only a negative response.

4.3.2 "Barbecue Spit" Experiment

Experiments performed using a horizontal rotation axis which yaw the experimental subject (barbecue spit rotation) will also be modeled. Goldberg and Fernandez (1982) measured the horizontal eye movements of squirrel monkeys during this type of stimulation while rotating the monkey at a constant velocity of 30 degrees per second. Figure 4.23 shows the measured response for rightward (A) and leftward (B) rotations.

Note that the eye movement response extends well beyond that which would be elicited with a vertical axis of rotation. Also note that the response seems to indicate the presence of the steady state component with a superimposed sine wave.

"VISUAL DUMPING" - DATA



From: Guedry, Collins, and Sheffey (1961)

Average slow-phase nystagmic response in degrees per 1-sec interval for all S's except DD

Figure 4.21

"VISUAL DUMPING" - MODEL PREDICTIONS

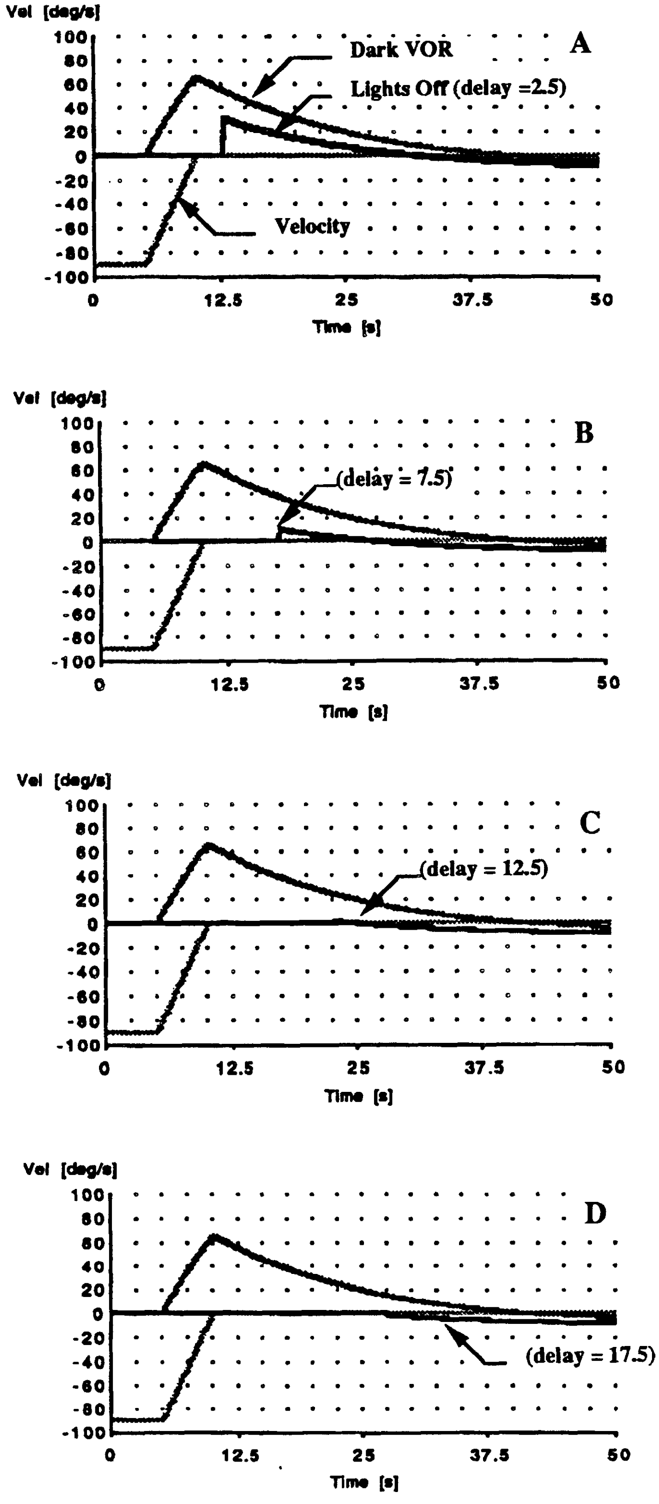
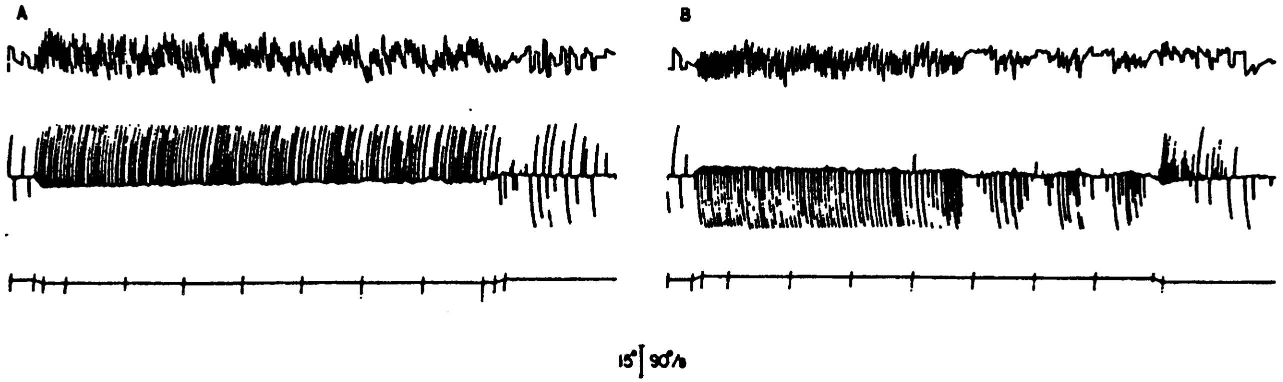


Figure 4.22

"BARBECUE SPIT" - DATA



From: Goldberg and Fernandez (1982)

Horizontal eye movements during barbecue -spit rotations in the dark. All recordings are derived from a single session in one animal. Each panel consists of an eye-position trace (middle), and the tachometer signal and superimposed calibration pulses (bottom). The pulses occur at the start of each period of the velocity trapezoid (T1-T5) and whenever the animal passes through the nose-down position during the constant-velocity (T3) period. A is a rightward rotation of 30 degrees/second. B is a leftward rotation of 30 degrees/second. The 15 degree calibration in the middle refers to the eye-position traces, the 90 degrees/second calibration to the eye-velocity traces. Rightward eye position and eye velocity are signified by upward deflections, rightward table velocity by a downward tachometer deflection.

Figure 4.23

Figure 4.24 show the internal states predicted by the model. Note that the model predicts a response which lasts as long the rotational stimulus, and that the model predicts a sinusoidal modulation via a state representing linear acceleration.

The most important observation to be derived from this data set is that the model predicts that gravitational error ($\vec{e}_g$) can stimulate a prolonged sense of rotation even when in conflict with the semicircular canal afference.

4.3.3 "OVAR" Experiment

In a similar study, Raphan and Cohen (1981) recorded horizontal and vertical eye movements while rotating monkeys about an axis tilted 30 degrees from vertical. See figure 4.25 for the published data. Notice a prolonged response similar to that recorded by Goldberg and Fernandez during barbecue spit rotation. Also note the sinusoidal modulation which dominates the vertical response. Upon deceleration the monkey develops horizontal and vertical nystagmus even though the the stimulus to the semicircular canals would only indicate a yaw rotation.

The model qualitatively demonstrates <u>all</u> of these characteristics and even makes some additional predictions. Figure 4.26 shows the response of the model to the same OVAR stimulation. Note that the yaw rotation state ($\hat{\omega}_z$) shows prolonged activity. Also notice that a sinusoidal pitch ($\hat{\omega}_y$) response builds up as the yaw response decays to a steady state level. Furthermore, the model predicts a sinusoidal roll response ($\hat{\omega}_x$) which lags behind the pitch response by 90 degrees. The model also predicts an oscillating sense of linear acceleration. Doubly integrating the sinusoidal acceleration yields a circular trajectory similar to that traversed at the edge of a Ferris wheel. The model also correctly predicts a vertical response following deceleration.

4.3.4 "Dumping" Experiment

The dumping paradigm was described in the previous chapter. Briefly, the stimulus is a constant velocity rotation about a vertical axis. After approximately a minute of constant velocity rotation (after the VOR has decayed to near zero), the monkey is

"BARBECUE SPIT" - MODEL PREDICTIONS

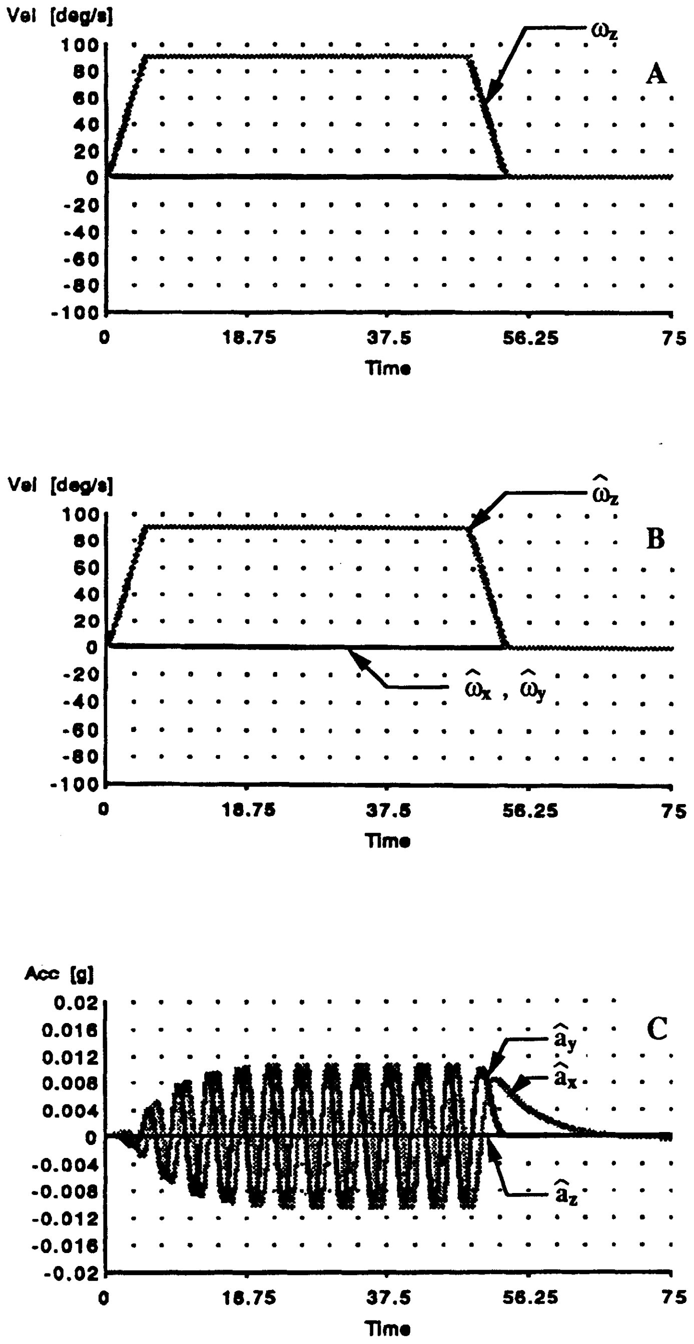
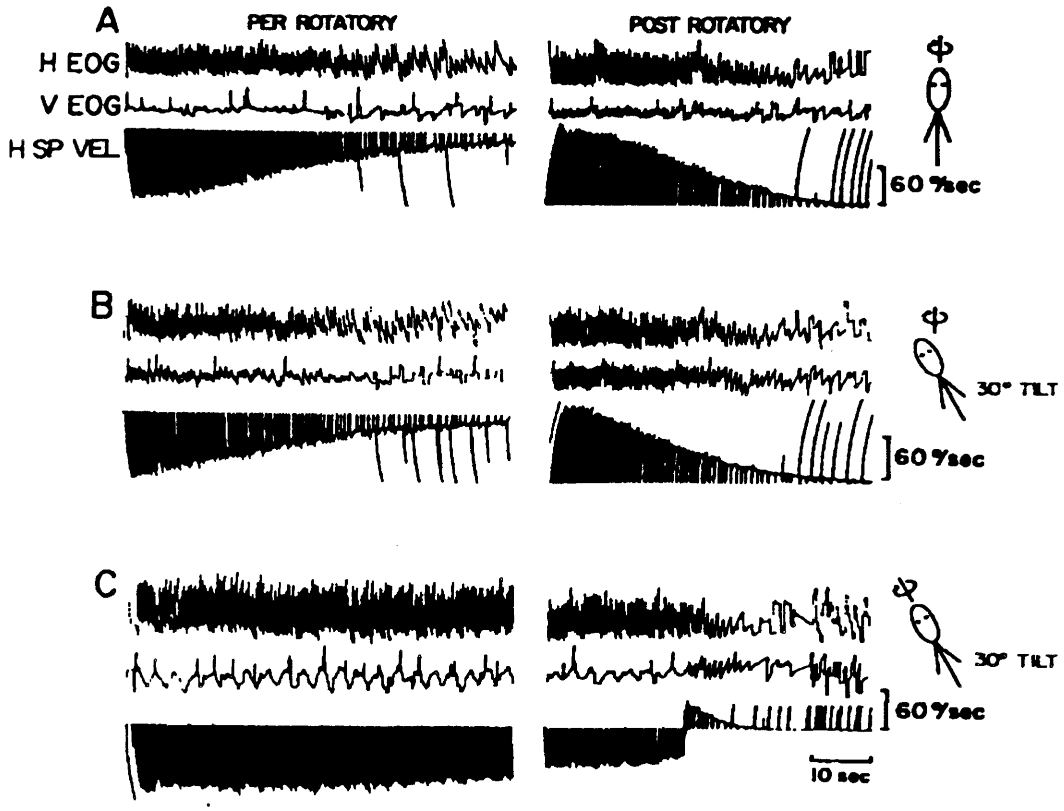


Figure 4.24

"OVAR" - DATA



From: Raphan and Cohen (1981)

Per- and postrotatory nystagmus induced by a step in velocity of 90deg/sec. The upper trace is horizontal EOG, the second trace is vertical EOG, and the third trace is slow phase eye velocity. In **A**, the animal had vertical axis rotation. In **B**, it had vertical axis rotation but was tilted 30 degrees. In **C**, the animal was rotated about an off-vertical axis tilted 30 degrees from the vertical. It was stopped in the same right-side-down position as in B. Note the slow decline in nystagmus after vertical axis rotation and the rapid decline after off-vertical rotation.

Figure 4.25

"OVAR" - MODEL PREDICTIONS

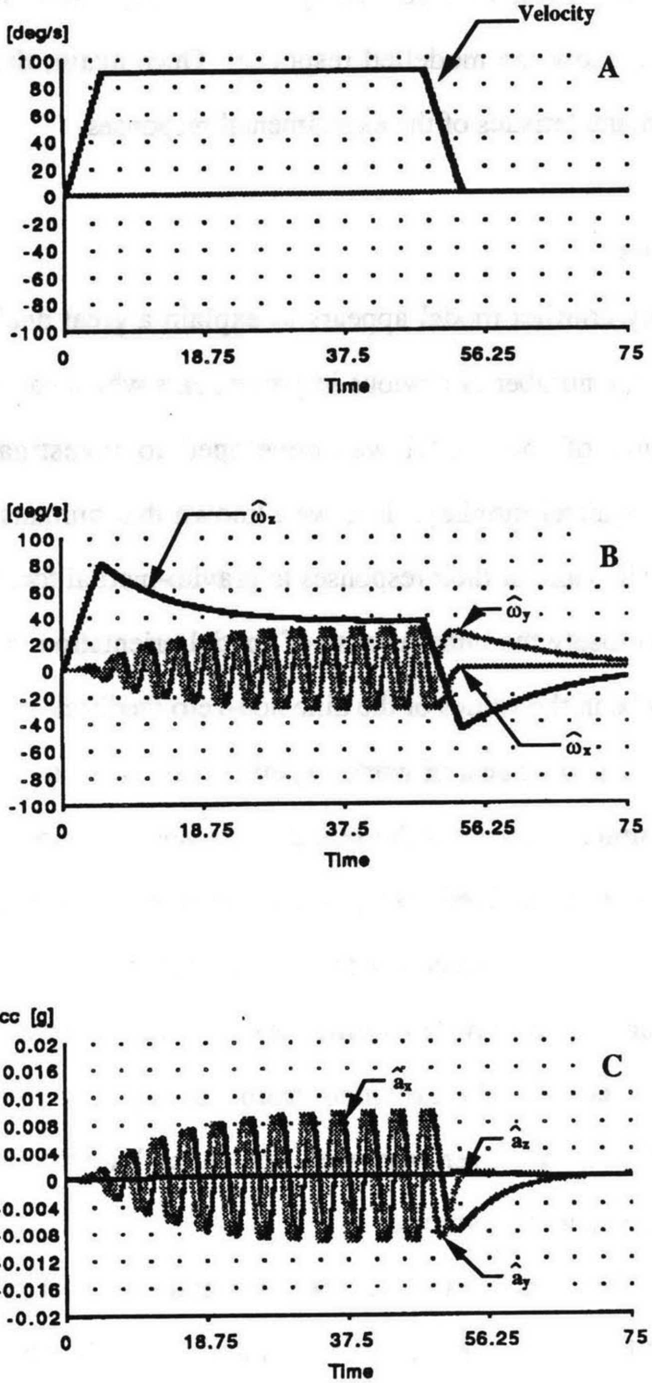


Figure 4.26

decelerated to a stop. In the example shown in figure 4.27, the monkey is tilted 45 degrees to the right immediately after stopping. This tilt is indicated by the roll response shown in D. Observe the sudden build up in vertical eye velocity (C) immediately following the tilt.

Figure 4.28 show the modelled response. Once again, the model, qualitatively, demonstrates the characteristics of the experimental responses.

4.4 Conclusions

The sensory conflict model appears to explain a great deal of experimental data. There are, however, a number of obvious improvements which can be made.

This version of the model was developed to investigate and model spatial orientation in the squirrel monkey. It is well known that humans and squirrel monkeys differ dramatically in some of their responses to gravito-inertial force. Changing this model to represent more closely the human sense of spatial orientation is a trivial exercise. The main changes will be in the values of the nine non-zero feedback gains.

Furthermore, a great deal of work, experimental and theoretical, needs to be done in order to more accurately represent the sensation of linear motion. The process by which gravito-inertial force is resolved into gravity and linear acceleration is not completely understood and is a problem which needs to be further investigated. Figure 4.29 shows a block diagram which could provide a framework for future work on this topic.

Vision was completely excluded from playing any role in this model. The importance of vision is, however, very well known, and, therefore, represents an error in the model which needs to be corrected.

One relatively simply change can be made at the level of the sensory dynamics. In the present model both the semicircular canals and the otoliths have frequency responses which are constant no matter how much the frequency exceeds 1 Hz. For the otoliths this may be remedied by adding a low pass filter with a cut-off frequency of approximately 5 Hz. The semicircular canals may be altered in a similar manner. These changes will not

"DUMPING" - DATA

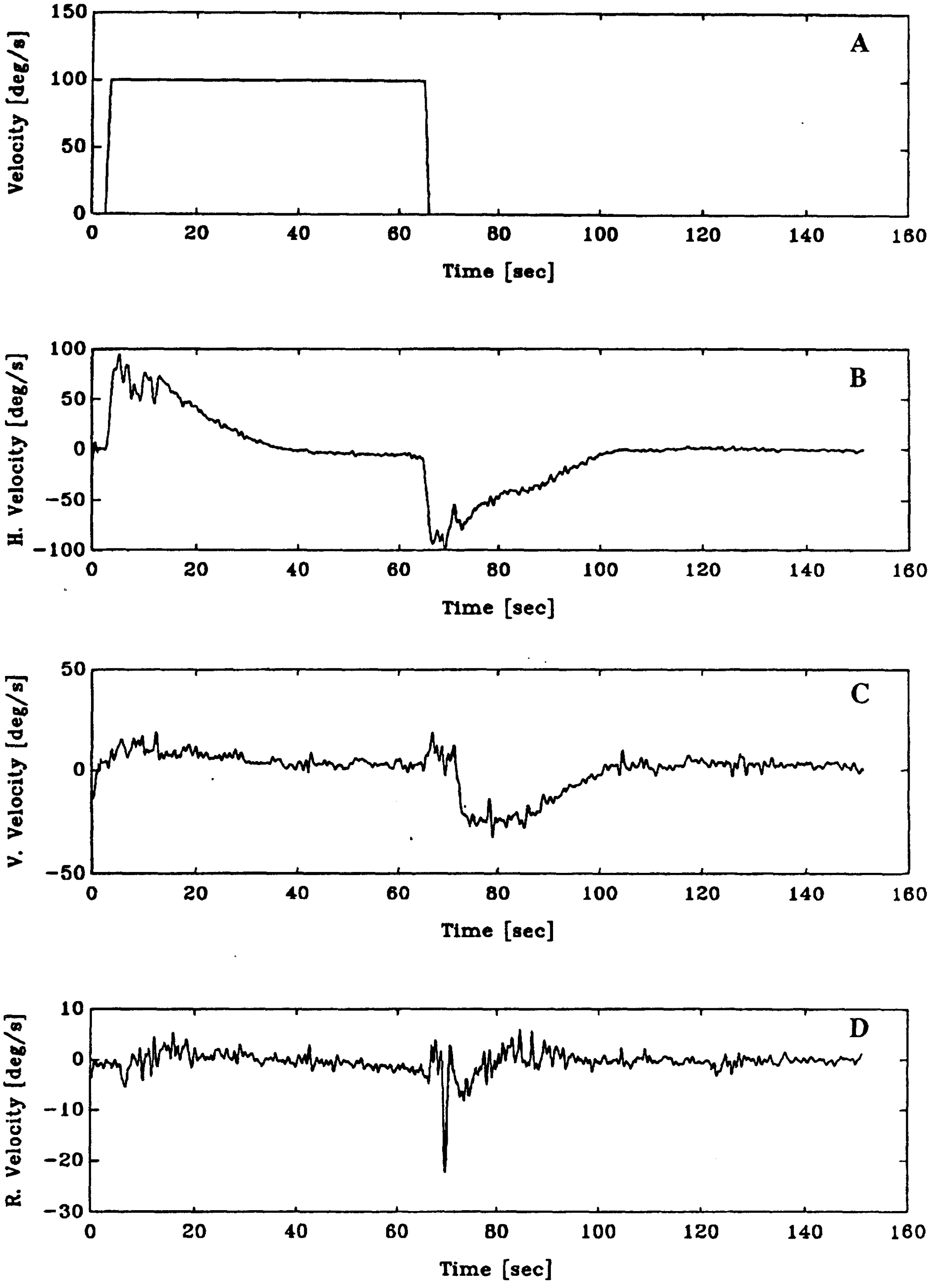


Figure 4.27

"DUMPING" - MODEL PREDICTIONS

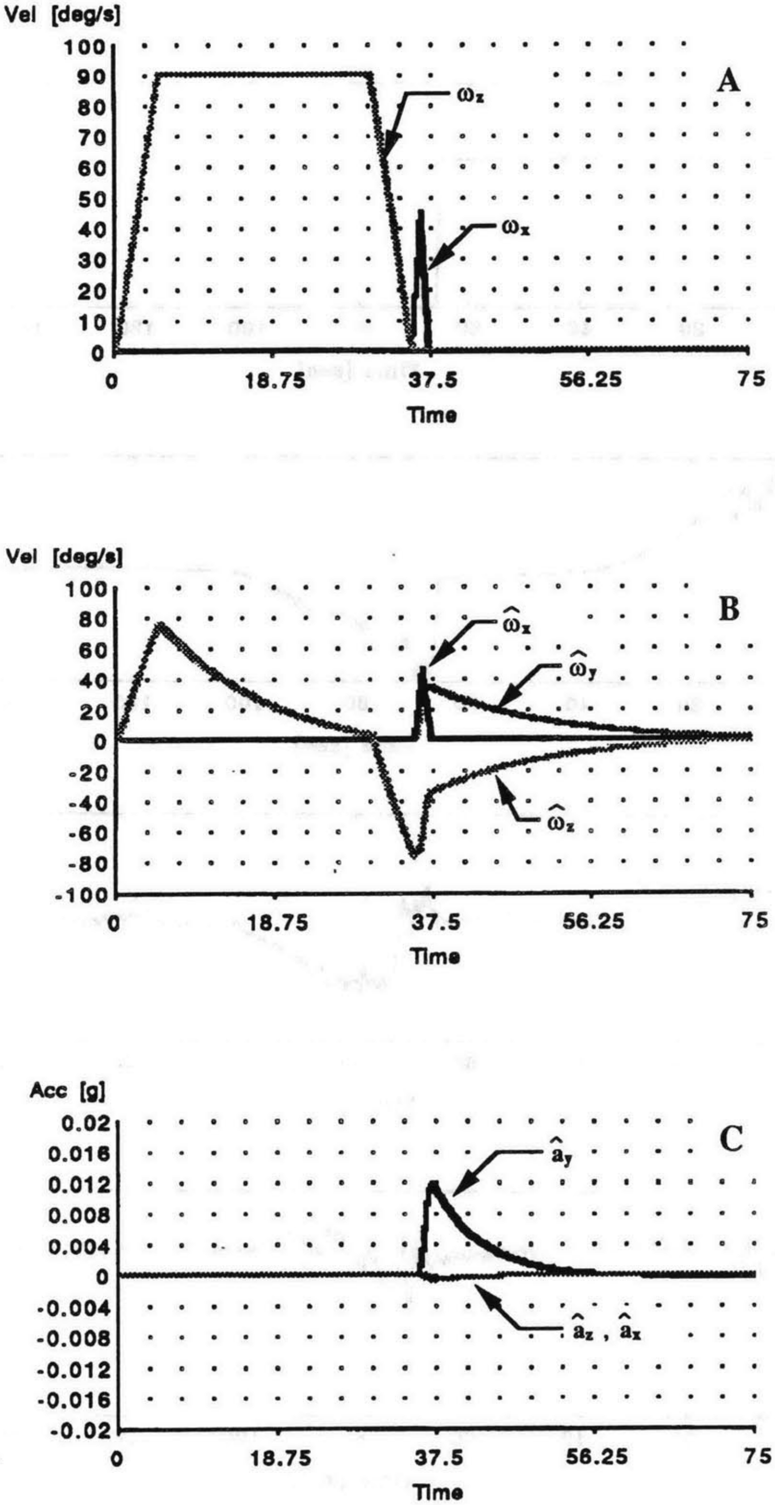


FIGURE 4.28

PROPOSED SENSORY CONFLICT MODEL

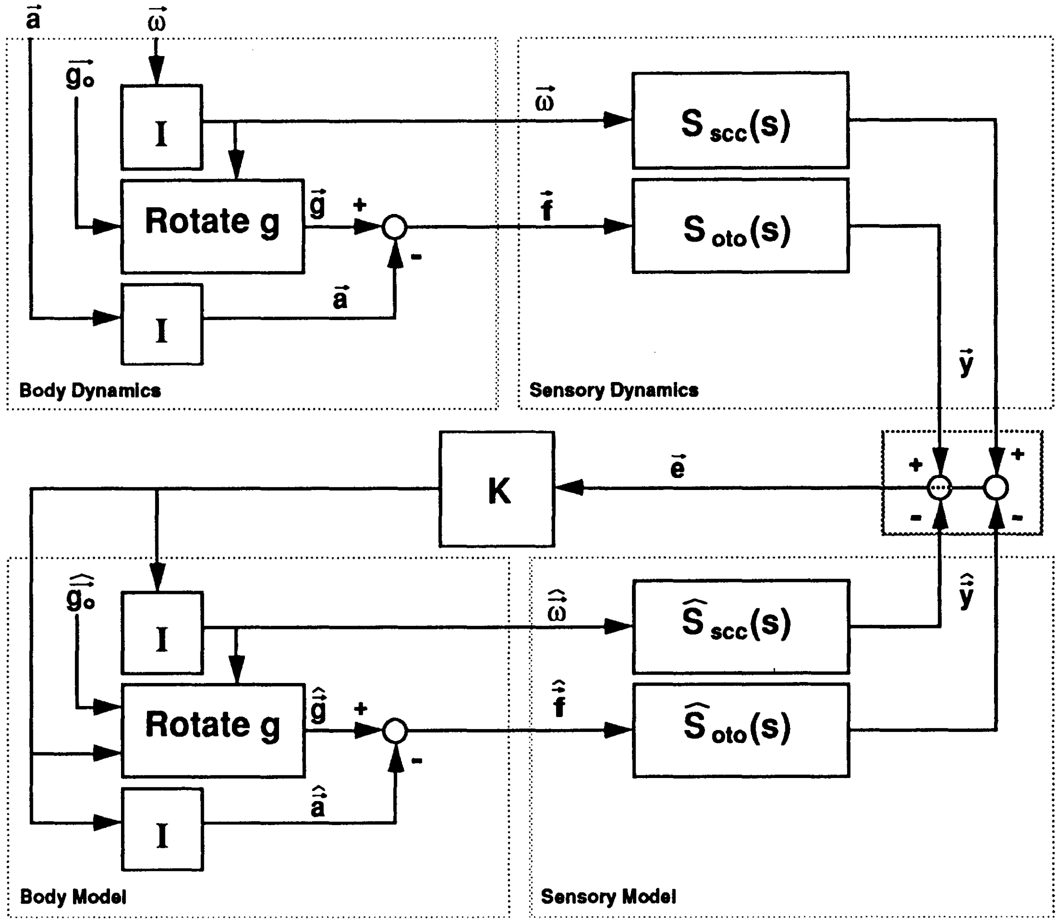


Figure 4.29

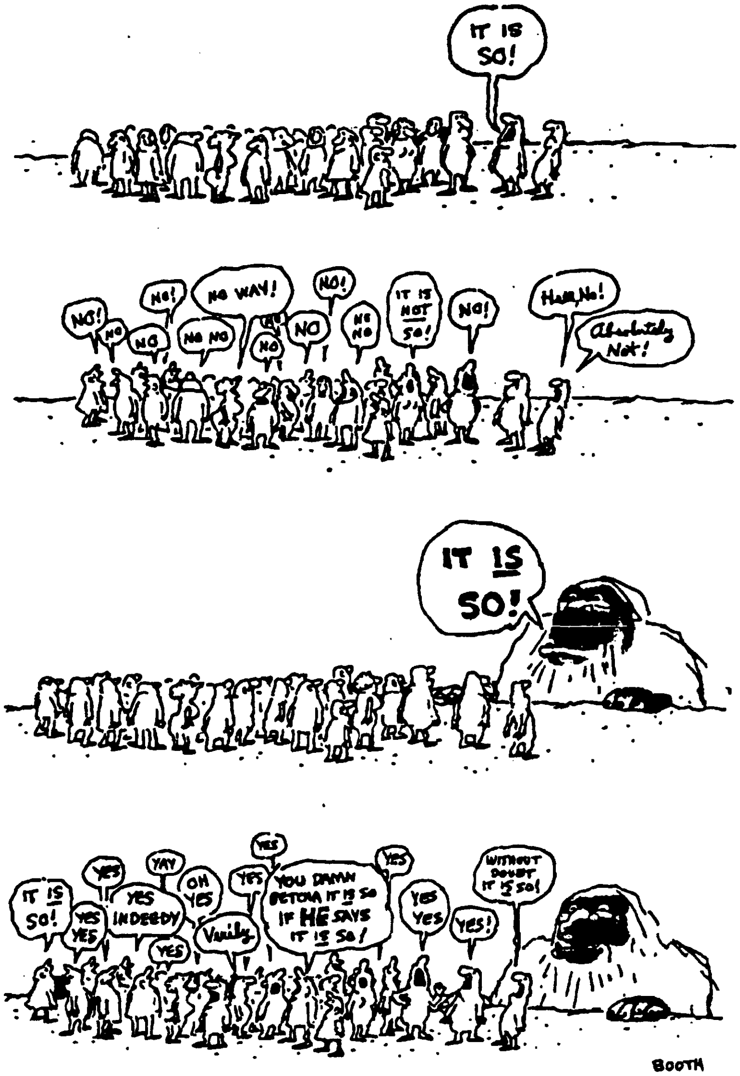
affect the system responses already modelled, but will provide a more stable model and a more realistic model particularly at high frequencies (> 1 Hz).

The roll played by tactile and proprioceptive cues in determining spatial orientation has been investigated (Bles et al. 1983). Adding these sensory modalities to the model may prove fruitful.

A great deal of work could be done to implement the motor system as part of the sensory conflict model. For simplicity, this portion of the model was discarded as the first step of development. This does not indicate that I feel that the motor system is superfluous. Rather, I feel that this portion of the model may be the most important contribution of this type of model. Extensions of this portion of the model are applicable to models of posture control and to models of voluntary movement.

V

SUMMARY AND CONCLUSIONS



"There was things which he stretched, but mainly he told the truth."
Mark Twain

"What's done cannot be undone, to bed, to bed, to bed."
Shakespeare

CHAPTER 5

SUMMARY AND CONCLUSIONS

The GIF resolution hypothesis, as stated in Chapter 1, claims that gravito-inertial force is resolved into two components: one representing gravity and one representing linear acceleration. Furthermore, the hypothesis states that the CNS uses all available information to create an internal estimate of "down". This estimate is compared to the measured GIF to determine linear acceleration.

In the experiments presented in Chapter 3, I showed that the axis of eye rotation, the peak value of SPV, and the SPV decay time constant were all dependent upon subject orientation during centrifugation. Further processing of the data indicated the presence of a linear VOR. As discussed in Chapter 3, all of these findings are consistent with the GIF resolution hypothesis, and therefore provide evidence supporting it. In addition, a number of results from the literature (OVAR, centrifugation) for both monkeys and humans are discussed. These results provide additional support for the GIF resolution hypothesis.

In Chapter 4 a model which incorporates the GIF resolution hypothesis was implemented. This model is based upon observer theory, optimal observer theory (Kalman Filtering), and general nonlinear concepts. The model is very successful in its ability to mimic data from this thesis as well as data from the literature.

After developing this sensory conflict model, I felt that improvements could be made. At the end of Chapter 4, I suggested an improved structure for the sensory conflict model. This new model presumed that the internal states representing gravity and linear acceleration sum before passing through an internal model of the otolith dynamics. In this implementation, the internal states representing gravity and linear acceleration mutually affect one another other. I think this model more accurately represents the processing performed by the CNS.

I feel that the GIF resolution hypothesis should be altered to be congruent with the suggested structure of the model. The new hypothesis states that the CNS resolves gravito-inertial force into two components: one representing gravity and one representing linear acceleration. These states will mutually and interactively sum to yield the total measured gravito-inertial force under most conditions.

APPENDIX A

DATA ACQUISITION

Data was sampled at 200 Hz using a Metrabyte data acquisition board (DAS 16) installed on a Compaq 386/20 computer. A program called Notebook (copyright by LabTech) was used to interface with the Metrabyte board.

Notebook is an easy to use menu-driven program which allows flexible top level management of data acquistion hardware without forcing the user to write any source code. Users must create a file containing the parameters required by Notebook. Once created, this file only needs to be changed when the users requirements change.

In order to automate the process of data acquisition, I wrote a C program which is automatically called every time Notebook is run. The program asks the user for the name of the previously stored set-up file and asks for the data code which the user wants to use. After receiving and verifying this information, the program automatically catalogs the data using the information provided by the user. This program is called autorun.c and is listed on the following pages.

```
/* PROGRAM autorun.c
Programmer: Dan Merfeld
Date: 6/15/88

This program helps to automate data acquisition as performed with LABTECH
Notebook. The program is automatically called when notebook is run. The
program queries the user for the desired set up file and the desired output
file data code. The program then automatically takes care of bookkeeping
by creating a sequence of data files. */

#include <c:\c\stdio.h>
#include <C:\c\math.h>
#include <C:\c\string.h>
#include <C:\c\io.h>
#include <C:\c\fcntl.h>
main()
{
    int count, chan, counter;
   char ans,dummy[2],name[5],set_up[10],dataout[25],out[50],kount[5];
   char kounter, variable1[10],variable2[15],variable3[15];
   FILE *fopen(), *fd1;
    int j,i,length[8];
   char string[15];
   char offset[15][8],scale[15][8],chan_name[10][8],number[8];
    ans='!'; /* ! is a dummy character */
    while(ans != 'N' && ans != 'n' && ans != 'Y' && ans != 'y')
     {
      printf("Recall a previously stored set-up (Y/N)?\n");
      scanf("%s",dummy);
      ans=dummy[0];
     }
    if(ans == 'Y' || ans == 'y')
     {
     while(ans == 'Y' || ans == 'y')
      {
        printf("Enter the set-up name.\n");
        scanf("%s",set_up);

       printf("Enter three letter data code (e.g. D2A).\n");
        scanf("%3s",name);

       printf("Enter the initial three digit sequence code.\n");
        scanf("%3d",&count);
         count=count+1000;
             printf("The set-up file name is ");
             printf("%s\n",set_up);
             sprintf(kount,"%d",count);
             sprintf(dataout,"%3s%c%c%c",name,kount[1],kount[2],kount[3]);
             printf("The data file code is %sCx\n",dataout);
             printf("Do you want to make any changes (Y/N)?\n");
             scanf("%s",dummy);
             ans=dummy[0];
      }
     sprintf(out,"copy setup\\%s",set_up);
     printf("%s\n",out);
     system(out);
    printf("Do you want to convert data to matlab format (Y/N)?\n");
    scanf("%s",dummy);
```

```
ans=dummy[0];
if(ans=='Y' || ans=='y')
 {
    fd1=fopen("junk.dat","r");
     chan=0;
     fscanf(fd1,"%s",string);
    while((strcmp(string,"\"eof\"") != 0) && (chan!=8))
     {
       if(strcmp(string,"block\"")==0)
        {
         fscanf(fd1,"%s",string);
         length[chan]=strlen(string)-2;
         for(i=0;i<10;i++) chan_name[i][chan]='\0';
         for(i=0;i<(length[chan]);i++)
             chan_name[i][chan]=string[i+1];
         for(i=0;i<3;i++) fscanf(fd1,"%s",string);
         if(strcmp(string,"0")==0)
          {
             for(i=0;i<5;i++) fscanf(fd1,"%s",string);
             number[chan]=string[0];
             for(i=0;i<8;i++) fscanf(fd1,"%s",string);
              i=0;
            while((offset[i][chan]=string[i])!='\0')
              i++;
             for(i=0;i<3;i++) fscanf(fd1,"%s",string);
              i=0;
            while((scale[i][chan]=string[i])!='\0')
              i++;
             chan=chan+1;
          }
    }
   fscanf(fd1,"%s",string);
    }
 fclose(fd1);
 ans='n';
 while(ans != 'N' && ans != 'n')
 {
  counter=0;
  sprintf(kount,"%d",count);
 sprintf(dataout,"%3s%c%c%cCx",name,kount[1],kount[2],kount[3]);
 printf("\n\n\nThe data code is %s.\n",dataout);
  printf("GO\n");
  system("GO");
  while (counter < chan)
   {
 sprintf(out,"Rename temp%d.prn temp.prn",counter+1);
 /*  +1 added to make the first channel number one since #0 is bad */
 printf("%s\n",out);
 system(out);
 for(i=0;i<8;i++) variable1[i]='\0';
 variable2[12]='\0';
 variable3[12]='\0';
 for(i=0;i<length[counter];i++) variable1[i]=chan_name[i][counter];
  for(i=0;i<12;i++)
 {    variable2[i]=offset[i][counter];
     variable3[i]=scale[i][counter];
  }
 sprintf(out,"convert %s %s %s",variable1,variable2,variable3);
```

```
    printf("%s\n",out);
    system(out);
   sprintf(dataout,"%3s%c%c%cC%c",name,kount[1],kount[2],kount[3],number[counter]+1);
   /* +1 added to compensate for fact that channel #0 is broken */
   sprintf(out,"Rename temp.prn %s.dat\n",dataout);
    printf("%s",out);
    system(out);
   sprintf(out,"Rename temp.mat %s.mat\n",dataout);
       printf("%s",out);
    system(out);
    counter++;
      }
   printf("Do you want to start another run (Y/N)?\n");
   scanf("%s",ans);
   ans=dummy[0];
    count++;
    }
    }
    else
    {
   Printf("How many data channels do you have?\n");
   scanf("%d",&chan);
   printf("Do you want to start a run (Y/N)?\n");
   scanf("%s",dummy);
   ans=dummy[0];
    while(ans != 'N' && ans != 'n')
    {
          counter=0;
               sprintf(kount,"%d",count);
               sprintf(dataout," %.3s%c%c%cC",name,kount[1],kount[2],kount[3]);
               printf("The data code is%sX.\n",dataout);
               printf("GO\n");
               system("GO");
               while (counter < chan)
               {
              sprintf(out,"Rename temp%d.prn%s%d.dat",counter+1,dataout,counter+1);
              /* +1 added (two places) to compensate for a/d # 0 not working */
               printf("%s\n",out);
               system(out);
               counter++;
               }
    printf("Do you want to start another run (Y/N)?\n");
    scanf("%s",dummy);
    ans=dummy[0];
     count++;
     }
     }
     }
}
```

APPENDIX B

DATA ANALYSIS (CALCULATION OF SPV)

B.1 Automation Programs

WARNING: Be careful when running "spv.m" (discussed below). This set of programs was set up to analyze a very specific set of files. This program deletes all of the *.mat and *.dat files in the directory in which you are running the program. Make sure that you are running the program from a subdirectory that doesn't contain any of these files.

A set of programs was developed to automate the process by which this specific set of raw data was converted to an estimate of slow phase eye velocity. Figure B.1 shows a block diagram which represents the basic structure of the program.

The auto4 programs (auto4.m and auto4.c) automate the process by asking the user for information such as the threshold values that she wants to use and by asking the user which files she wants to analyze. The programs take this information and store it in various files. The most important of these files is called "spv.m" and contains the list of commands that the user would need to type in order to analyze the data. By typing "spv", after running auto4, this process is activated.

Three programs (analyz41.bat, desaccade.m, and analyz42.bat) are called in the spv.m file for each of the data files analyzed. Analyz41.bat in turn calls a series of programs that convert the measurements of eye position to eye velocity and eye acceleration. Desaccade.m takes the eye velocity and eye acceleration files and calculates an estimate of slow phase eye velocity. Analyz42.bat renames the output file with the same name as the original data code and deletes all of the files created while processing.

To run the automated process, the user needs to enter Matlab by typing "matlab" and then type "auto4". This command starts running the program "auto4.m". This program will ask a set of self explanatory questions that require either yes/no answers or numbers. Before stopping, this program calls the program "auto4.c" which further

Global Process Flow

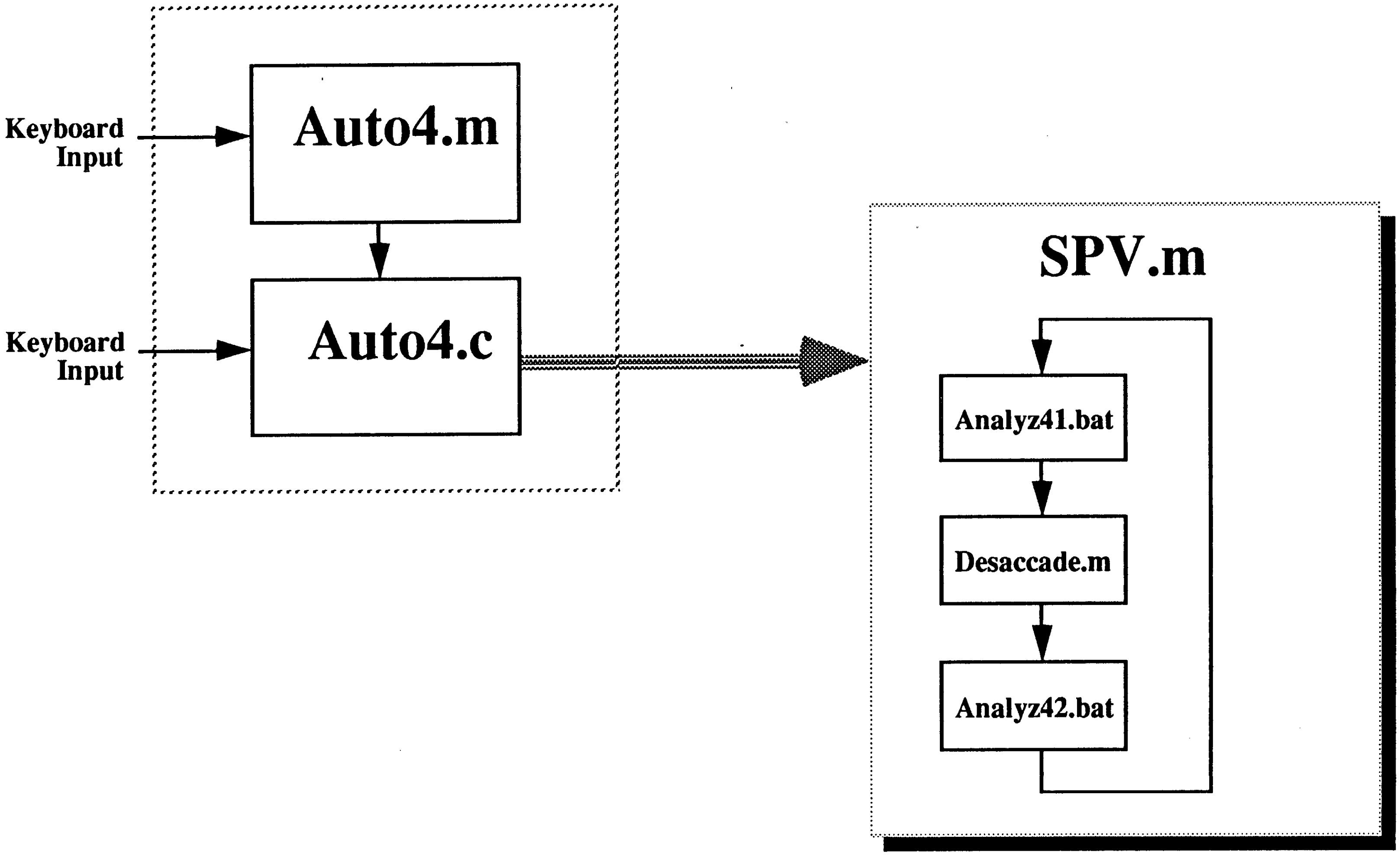


Figure B.1

queries the user about scale factors and the data files that the user wants to analyze. After all of the questions are answered the program creates an m-file called "spv.m"

The auto4 programs are only a preprocessing step. The actual data analysis begins when the "spv" program is activated.

The source code for auto4.m and auto4.c is shown on the following pages.

```
% m-file auto4.m
% Programmer: Dan Merfeld
% Date: 9/15/89

% This m-file queries the user to enter the acceleration thresholds to
% be used and the velocity thresholds to be used. It then calls the program
% auto4.c to load the scale factors, the bias values, and the angle between
% coils. The output files are called "thresh.mat" and "thresh2.mat"

load c:\data\thresh1

thresh_acc
thresh_end


ans=input('Are these threshold values the ones you want to use in detect1?  ','s');
if(ans == 'Y')
        ans='y';
end

while( ans ~= 'y')

        thresh_acc=input('Enter the value for thresh_acc  ');
        thresh_end=input('Enter the value for thresh_end  ');
        thresh_acc
        thresh_end
        ans=input('Are these threshold values the ones you want to use in detect1?  ','s');
        if(ans == 'Y')
                ans='y';
        end
end

save c:\data\thresh thresh_acc thresh_end


load c:\data\thresh2

vel_test

ans=input('Is this threshold value the one you want to use in detect2? ','s');
if(ans == 'Y')
        ans='y';
end

while( ans ~= 'y')

        vel_test=input('Enter the value for vel_test  ');
        vel_test
        ans=input('Is this threshold value the one you want to use in detect2?  ','s');
        if(ans == 'Y')
                ans='y';
        end
end

save c:\data\thresh2 vel_test
clearall

!auto4            %call auto4.c
```

```
/* PROGRAM auto4.c
Programmer: Dan Merfeld
Date: 9/15/89

This program completely automates the automated analysis process described
in my thesis. This program is called by auto4.m as the first step
of the process. The program queries the user for scale factors,
bias values, the angle between the coils, and for the data files
to be analyzed. The program then creates a set of files
which automate the analysis process. The parameters entered by the user
are stored in a file called "data.con". The product in an m-file
called "spv.m" which contains the list of commands needed to analyze all
of the data. */

#include <c:\c\stdio.h>
#include <C:\c\math.h>
#include <C:\c\string.h>
#include <C:\c\io.h>
#include <C:\c\fcntl.h>
main()
{
    int i,j;
   unsigned long int dlength;
    float blength;
   int fi1,bias[3],nwrite,nread,rtotal;
   float scale[3],dot,angle;
   FILE *fd1,*fd2,*fd3;
   char string[45],fname[45],file[12];
   char length[45],ans;

loop:
   if((fi1=open("c:\\data\\data.con",O_RDONLY|O_BINARY))==-1)
            {ans='y';}
    else
            {nread=read(fi1,bias,12);
            rtotal=nread;
            nread=read(fi1,scale,12);
            rtotal=rtotal+nread;
            nread=read(fi1,&dot,4);
            rtotal=rtotal+nread;
            i=close(fi1);
            if(rtotal != 28)
                    {ans='y';}
            else
                    {printf("The bias values are %d (H), %d (V), %d (T)\n",bias[0],bias[1],bias[2]);
                    printf("The scale factors are %e (H), %e (V), %e (T)\n",scale[0],scale[1],scale[2]);
                    angle=(180.-180/3.1415*acos(dot));
                    printf("(The angle between the coils is %5.1f degrees)\n",angle);
                    printf("Do you want to change these values?\n");
                    scanf("%s",string);
                    ans=string[0];
                    if(ans == 'N') ans='n';}}
if(ans != 'n')
            {printf("Enter the value of the horizontal A/D bias:\n");
            scanf("%d",&bias[0]);
            printf("Enter the value of the vertical A/D bias:\n");
            scanf("%d",&bias[1]);
            printf("Enter the value of the torsional A/D bias:\n");
```

```
        scanf("%d",&bias[2]);
        printf("Enter the value of the horizontal scale factor:\n");
        scanf("%e",&scale[0]);
        printf("Enter the value of the vertical scale factor:\n");
        scanf("%e",&scale[1]);
        printf("Enter the value of the torsional scale factor:\n");
        scanf("%e",&scale[2]);
        printf("Enter the value of the angle between the eye coils [deg]:\n");
    scanf("%f",&angle);
        dot=180-angle;
        dot=cos(dot*3.141592654/180.);
        sprintf(string,"attrib -r data.con\0");
         i=system(string);
        sprintf(string,"del data.con\0");
         i=system(string);
        fi1=open("data.con",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
        nwrite=write(fi1,bias,12);
        nwrite=write(fi1,scale,12);
        nwrite=write(fi1,&dot,4);
        i=close(fi1);
        sprintf(string,"attrib -r data.con\0");
         i=system(string);
        sprintf(string,"copy data.con c:\\data\\data.con\0");
         i=system(string);
         goto loop;}
else
        {sprintf(string,"copy c:\\data\\data.con c:\\matlab\\dan\\analyze\0");
        i=system(string);}
printf("Enter the file code which you wish to analyze\n");
scanf("%s",file);
sprintf(string,"dir c:\\data\\%s.dat >dir.lst",file);
printf("%s\n",string);
i=system(string);
sprintf(string,"del spv.m");
printf("%s\n",string);
i=system(string);
sprintf(string,"attrib -r log\n");
printf("%s\n",string);
i=system(string);
sprintf(string,"del log");
printf("%s\n",string);
i=system(string);
sprintf(string,"attrib -r file*.nam");
printf("%s\n",string);
i=system(string);
sprintf(string,"del file*.nam");
printf("%s\n",string);
i=system(string);

fd1=fopen("dir.lst","r");
fd2=fopen("spv.m","w");
sprintf(string,"!attrib -r temp.*\n");
j=fputs(string,fd2);
sprintf(string,"!del temp.*\n");
j=fputs(string,fd2);
sprintf(string,"!attrib -r *.mat\n");
j=fputs(string,fd2);
sprintf(string,"!del *.mat\n");
```

```
    j=fputs(string,fd2);
    sprintf(string,"!attrib -r *.dat\n");
    j=fputs(string,fd2);
    sprintf(string,"!del *.dat\n");
    j=fputs(string,fd2);
    sprintf(string,"diary log\n\n");
    j=fputs(string,fd2);
    for(i=0;i<10;i++){fscanf(fd1,"%s",string);}

    for(i=0;i<100;i++)
    {       fscanf(fd1,"%s",fname);
            fscanf(fd1,"%s",string);
            fscanf(fd1,"%s",length);
            if(strcmp(string,"DAT")==0)
            {   blength=atof(length);
                    dlength=blength/2;
/*                  sprintf(string,"fprintf('analyzing %.6s\\n')\n",fname);
                    j=fputs(string,fd2);  */
                    sprintf(file,"file%.3d.nam\0",i+100);
                    fd3=fopen(file,"w");
                    sprintf(string,"%.6s",fname);
                    j=fputs(string,fd3);
                    j=fclose(fd3);

                    sprintf(string,"!analyz41 %.6s %s %u\n",fname,file,dlength);
                    j=fputs(string,fd2);
                    sprintf(string,"desaccade\n");
                    j=fputs(string,fd2);
                    sprintf(string,"!analyz42 %.6s\n\n",fname);
                    j=fputs(string,fd2);
/*                  sprintf(string,"fprintf('%.6s analyzed\\n\\n\\n\\n')\n\n",fname);
                    j=fputs(string,fd2);  */
                    for(j=0;j<17;j++) fscanf(fd1,"%s",string);
            }
            }
    sprintf(string,"diary off\n");
    j=fputs(string,fd2);
    sprintf(string,"clearall\n");
    j=fputs(string,fd2);
/*    sprintf(string,"!print log\n");
    j=fputs(string,fd2);  */
    sprintf(string,"!c:\\util\\park\n");
    j=fputs(string,fd2);
    i=fclose(fd1);
    i=fclose(fd2);
}
```

B.2 C Analysis Programs

This appendix contain a number of C programs which were written as part of the data analysis project discussed in chapter 2 of this thesis. There are seven c programs in this appendix and each is called from the analyz41.bat file previously discussed. A flow diagram of analyz41.bat is shown on the following page (Figure B.2) followed by the actual analyz41.bat file. (The analyz42.bat file is also included at this point.) The programs are described in the order that they are called by analyz41.bat.

Euler3f.c calculates an euler angle representation of the coil system with respect to the axis system defined by the magnetic field coils. The program implements the equations of section 2.1.1.2.

Filter5.c differentiates the euler angles to determine the euler rates. A five element digital filter is implemented to perform the differentiation.

Inertial.c calculates the inertial rates from angular rotation from the euler angles and euler rates. The program implements the equations developed in section 2.1.1.3.

Filter9.c differentiates the inertial rates to determine the rates of angular acceleration. A nine element digital filter is implemented to perform the differentiation.

Acc_mag.c calculates the magnitude of the angular acceleration vector from the components of angular acceleration calculated by Filter9.c.

F_convert converts floating point data files to the Matlab format. The program is used to creat Matlab files of angular velocity and the magnitude of angular acceleration.

I_convert converts short integer data files to Matlab format. The program is used to create a Matlab file of the stimulus velocity.

Analyz41.bat Process Flow

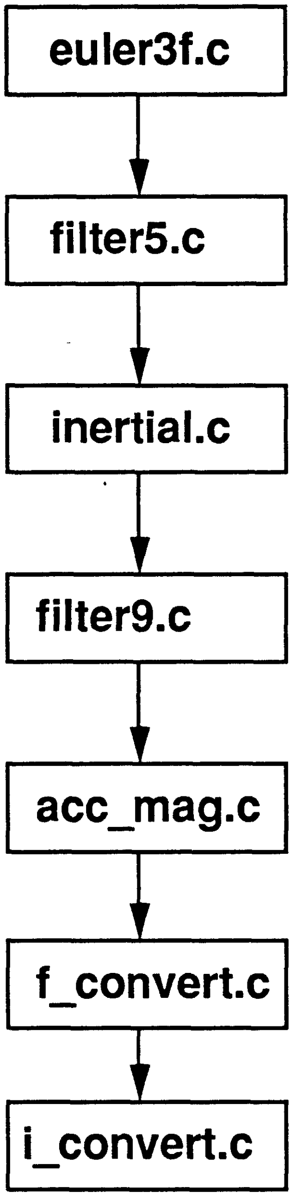


Figure B.2

```
rem Analyz41.bat
rem Programmer:  Dan Merfeld
rem Date:  9/15/89

rem This file helps to automate the analysis of eye movements by
rem taking care of some simply bookkeeping and calling the programs
rem in the proper order.

copy %2 file.nam

copy c:\data\%1C1.dat c12.prn
copy c:\data\%1C2.dat c13.prn
copy c:\data\%1C3.dat c23.prn

copy c:\data\%1C4.dat omega.prn

euler3f %3

filter5 %3

inertial %3

filter9 %3

acc_mag %3

rename h_vel.dat temp.dat
f_convert h_vel %3
rename temp.dat h_vel.dat
rename temp.mat h_vel.mat

rename v_vel.dat temp.dat
f_convert v_vel %3
rename temp.dat v_vel.dat
rename temp.mat v_vel.mat

rename r_vel.dat temp.dat
f_convert r_vel %3
rename temp.dat r_vel.dat
rename temp.mat r_vel.mat

rename m_acc.dat temp.dat
f_convert m_acc %3
rename temp.dat m_acc.dat
rename temp.mat m_acc.mat

rename omega.prn temp.prn
i_convert omega %3
rename temp.prn omega.prn
rename temp.mat omega.mat

del *.prn

del *.dat
```

```
rem Analyz42.bat
rem Programmer: Dan Merfeld
rem Date: 9/15/89

rem This batch file simply stores the matlab file which contains the output
rem data with a filecode similar to the original data files. It also deletes
rem some of the intermediate files.

copy temp.mat c:\matlab\dan\%1.mat

del *.mat
```

```
/* PROGRAM euler3f.c
Programmer: Dan Merfeld
Date: 9/15/89

This program reads from four input files and writes to three output files.

File "c12.prn" should be the direction cosines between the 1st coil and
the y magnetic field. (Horizontal eye position)
File "c13.prn" should be the direction cosines between the 1st coil and
the z magnetic field. (Vertical eye position)
File "c23.prn" should be the direction cosines between the 2nd coil and
the z magnetic field. (Torsional eye position)
File "data.con" should contain bias values for each of the three data
channels. It also should contain scale factors for the data channels.
Finally, the file should contain the cosine of the angle between the coils.

The program will correct for the fact that the eye coils are not orthogonal
and determine an euler angle representation of the coil system with respect
to the axis system defined by the magnetic fields.

The output files are "theta1.dat", "theta2.dat", and "theta3.dat". The time
series represented in each of these files are represented in radians.

The program should be called by the sequence "euler3f XXX", where XXX is the
number of points in each of the input data files. */

#include <c:\c\stdio.h>
#include <c:\c\io.h>
#include <c:\c\fcntl.h>
#include <c:\c\math.h>
#include <c:\c\errno.h>
#include <c:\c\stdlib.h>
#define  BLOCK    1024      /* Bytes to read and write each time */

short int c13[BLOCK],c23[BLOCK],c12[BLOCK]; /*2 bytes per integer*/
float theta1[BLOCK],theta2[BLOCK],theta3[BLOCK]; /* 4 bytes per output */
float C[9]; /* equivalent to a 3 by 3 matrix */

main(argc,argv)
int argc;
char *argv[];
{

extern int errno;
unsigned long int dlength,wtotal;
float blength;
int i,j,k,l,fi1,fi2,fi3,fi4,fo1,fo2,fo3,nread,nwrite;
int bias[3],junk;
float scale[3],pi;
float truedot,angle;
char out[40];

pi=3.1415926536;
if(argc!=2)
{         printf("\n\7Error - file length not specified");
          exit(-3);}
blength=atof(argv[1]);
dlength=blength;
```

```
printf("Finding euler angles for %u values\n",dlength);
if((fi1=open("C12.PRN",O_RDONLY|O_BINARY))==-1) /* Open the input files */
   {printf("\n\7Error opening C12.PRN");
   exit(-1);}
if((fi2=open("C13.PRN",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening C13.PRN");
   exit(-1);}
if((fi3=open("C23.PRN",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening C23.PRN");
   exit(-1);}

fo1=open("THETA1.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY); /* Open the output */
if(fo1==-1)
         {printf("\n\7Error opening THETA1.DAT for output");
         if(errno==EEXIST)printf("\nTHETA1.DAT already exists,must delete it");
         exit(-2);}
fo2=open("THETA2.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fo2==-1)
         {printf("\n\7Error opening THETA2.DAT for output");
         if(errno==EEXIST)printf("\nTHETA2.DAT already exists,must delete it");
         exit(-2);}
fo3=open("THETA3.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fo3==-1)
         {printf("\n\7Error opening THETA3.DAT for output");
         if(errno==EEXIST)printf("\nTHETA3.DAT already exists,must delete it");
         exit(-2);}

if((fi4=open("DATA.CON",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening DATA.CON");
   exit(-1);}
nread=read(fi4,bias,12); /* read the bias values */
nread=read(fi4,scale,12); /* read the scale values */
nread=read(fi4,&truedot,4); /* read the cos of the angle between the coils */
junk=close(fi4);
if(fabs(truedot) == 1.0)
  {printf("Enter the value of the angle between the eye coil (in degrees)\n");
  junk=scanf("%f",&angle);
  angle=180.-angle;
  truedot=cos(angle*3.1415926/180.);}

/*bias=2050;
scale=2.119728731e-04; 20./4096./(4./cos(90.-10.)); */

printf("Scale factors equal %e (H), %e (V), %e (T)\n",scale[0],scale[1],scale[2]);
printf("The A/D bias are %d (H), %d (V) %d (T)\n",bias[0],bias[1],bias[2]);
angle=180.-180./3.1415*acos(truedot);
printf("The angle between the coils is %5.1f)\n",angle);
wtotal=0;
for(j=0;;j++)
{
   nread=read(fi1,c12,BLOCK*2);
   nread=read(fi2,c13,BLOCK*2);
   nread=read(fi3,c23,BLOCK*2);
   for(k=0;k<nread/2;k++)
    {
             *(C+3)=(*(c12+k)-(4096-bias[0]))*scale[0]; /*1st scale the data*/
             *(C+6)=-(*(c13+k)-(4096-bias[1]))*scale[1];
             *(C+7)=(*(c23+k)-(4096-bias[2]))*scale[2];
```

```
            i=correction(truedot);  /* performs actual calculations */
            if(i!=0) printf("\nError from correction routine\n");

            *(theta2+k)=-asin(*(C+6));       /* calculate the output angles */
            *(theta1+k)=asin((*(C+3))/(cos(*(theta2+k))));
            *(theta3+k)=asin((*(C+7))/(cos(*(theta2+k))));
   }
  nwrite=write(fo1,theta1,4*(nread/2));  /* 4 bytes per value */
  nwrite=write(fo2,theta2,4*(nread/2));  /* 4 bytes per value */
  nwrite=write(fo3,theta3,4*(nread/2));  /* 4 bytes per value */
   wtotal=wtotal+nwrite/4;
  if(nwrite!=4*(nread/2)) printf("\nError 7");
  if(nread!=BLOCK*2) break;
}
if(dlength!=wtotal) printf("\n\7Error-total data bytes written mismatch");
junk=close(fi1);
junk=close(fi2);
junk=close(fi3);
junk=close(fo1);
junk=close(fo2);
junk=close(fo3);
sprintf(out,"attrib -r theta1.dat\0");
j=system(out);
sprintf(out,"attrib -r theta2.dat\0");
j=system(out);
sprintf(out,"attrib -r theta3.dat\0");
j=system(out);
}



/*  correction:  performs the corrections calculations
This subroutine just implements the cross products and other calculations
defined in Section 2.1.1.2 of my thesis.  */
int correction(truedot)
float truedot;
{

int j;
float temp,plus,minus,m;
temp=(*(C+3))*(*(C+3))+(*(C+6))*(*(C+6));
if(temp>1)
        {*(C)=0.00000001;}
else
        {*(C)=sqrt(1-temp);}
temp=2*(*(C+6))*(*(C+7))*truedot+(*(C+3))*(*(C+3))+(*C)*(*C)-(*(C+7))*(*(C+7));
temp=temp-truedot*truedot;
if(temp<0)
        {printf("ERROR calculating c22 [sqrt(temp<0)] temp=%f\n",temp);
        exit;}
else if(temp>1)
        {printf("ERROR calculating c22 [sqrt(temp>1)] temp=%f\n",temp);
        exit;}
else
        {temp=*(C)*sqrt(temp) + *(C+3)*truedot - *(C+3)*(*(C+6))*(*(C+7));
        *(C+4)=temp/(*(C+3)*(*(C+3)) + *(C)*(*(C)));}
```

```
*(C+1)=(truedot - *(C+3)*(*(C+4)) - *(C+6)*(*(C+7)))/(*(C));

*(C+2)=-(*(C+4))*(*(C+6)) + (*(C+3))*(*(C+7));
*(C+5)=-(*(C+7))*(*(C+0)) + (*(C+1))*(*(C+6));
*(C+8)=-(*(C+1))*(*(C+3)) + (*(C+4))*(*(C+0));
m=((*(C+2))*(*(C+2)) + (*(C+5))*(*(C+5)) + (*(C+8))*(*(C+8)));
if(m <=1)
{           m=sqrt(m);
            *(C+2)=(*(C+2))/m;
            *(C+5)=(*(C+5))/m;
            *(C+8)=(*(C+8))/m;}
*(C+1)=(*(C+5))*(*(C+6)) - (*(C+3))*(*(C+8));
*(C+4)=(*(C))*(*(C+8)) - (*(C+2))*(*(C+6));
*(C+7)=(*(C+3))*(*(C+2)) - (*(C))*(*(C+5));
return 0;
}
```

```
/* PROGRAM filter5.c
Programmer: Dan Merfeld
Date: 9/15/89

This program takes the output files from euler3f.c and differentiates
each time series to determine the euler rates.

The input files are "Theta1.dat", "Theta2.dat", and "Theta3.dat".
The output files are "Th1_vel.dat", "Th2_vel.dat", and "Th3_vel.dat".

The program should be called with the sequence "filter5 XXX", where XXX
is the number of points in each of the input data files. */

#include <c:\c\stdio.h>
#include <c:\c\io.h>
#include <c:\c\fcntl.h>
#include <c:\c\math.h>
#include <c:\c\errno.h>
#include <c:\c\stdlib.h>
#define  BLOCK    1024      /* Bytes to read and write each time */

float in[BLOCK];    /* 4 bytes per input */
float out[BLOCK],temp[BLOCK+4];
main(argc,argv)
int argc;
char *argv[];
{
extern int errno;
int fd1,fd2,j,k;
char string[40];
unsigned long int dlength;
float blength;

if(argc!=2)
{         printf("\n\7Error - file length not specified");
          exit(-2);}
blength=atof(argv[1]);
dlength=blength;
printf("Filtering %u values",dlength);

if((fd1=open("THETA1.DAT",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening THETA1.DAT");
   exit(-1);}

fd2=open("TH1_VEL.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fd2==-1)
          {printf("\n\7Error opening TH1_VEL.DAT for output");
          if(errno==EEXIST) {printf("\nTH1_VEL.DAT already exists, delete it");}
          exit(-2);}
j=filt5(fd1,fd2,dlength);  /* Subroutine performs the actual calculations */
if(j!=0)
          {printf("\n\7Error from filter function");
          exit(-1);}
close(fd1);
close(fd2);
sprintf(string,"attrib -r th1_vel.dat\0");
k=system(string);
```

```
if((fd1=open("THETA2.DAT",O_RDONLY|O_BINARY))==-1)
    {printf("\n\7Error opening THETA2.DAT");
    exit(-1);}

fd2=open("TH2_VEL.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fd2==-1)
          {printf("\n\7Error opening TH2_VEL.DAT for output");
          if(errno==EEXIST) {printf("\nTH2_VEL.DAT already exists, delete it");}
          exit(-2);}
j=filt5(fd1,fd2,dlength); /* Subroutine performs the actual calculations */
if(j!=0)
          {printf("\n\7Error from filter function");
          exit(-1);}
close(fd1);
close(fd2);
sprintf(string,"attrib -r th2_vel.dat\0");
k=system(string);

if((fd1=open("THETA3.DAT",O_RDONLY|O_BINARY))==-1)
    {printf("\n\7Error opening TH3.DAT");
    exit(-1);}

fd2=open("TH3_VEL.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fd2==-1)
          {printf("\n\7Error opening TH3_VEL.DAT for output");
          if(errno==EEXIST) {printf("\nTH3_VEL.DAT already exists, delete it");}
          exit(-2);}
j=filt5(fd1,fd2,dlength); /* Subroutine performs the actual calculations. */
if(j!=0)
          {printf("\n\7Error from filter function");
          exit(-1);}
close(fd1);
close(fd2);
sprintf(string,"attrib -r th3_vel.dat\0");
k=system(string);
}


/* filt5: performs all of the calculations */
int filt5(fd1,fd2,dlength)
int fd1,fd2;
unsigned long int dlength;
{
unsigned long int wtotal;
int j,k,nread,nwrite;
int bias;
float sample;

sample=200.; /* sampling rate in Hz */
nread=read(fd1,in,8);
*(temp)=*(in);
*(temp+1)=*(in);
*(temp+2)=*(in);
*(temp+3)=*(in+1);
wtotal=0;
for(j=0;;j++)
          {nread=read(fd1,in,BLOCK*4);
```

```
         if(nread!=BLOCK*4)
                  {*(in+nread/4+1)=*(in+nread/4-1);
                  *(in+nread/4)=*(in+nread/4-1);
                  nread=nread+8;}
   for(k=0;k<nread/4;k++)
            {*(temp+4+k)=*(in+k);
            *(out+k)=sample*(.125*(*(temp+4+k)-*(temp+k))+.25*(*(temp+3+k)-*(temp+1+k)));} /*
actual filtering equation */
   nwrite=write(fd2,out,4*(nread/4)); /* 4 bytes per value */
    wtotal=wtotal+nwrite/4;
   if(nwrite!=4*(nread/4)) printf("\nError 7");
   if(nread!=BLOCK*4)
   {break;}
    else
            {*temp=*(in+1020);
            *(temp+1)=*(in+1021);
            *(temp+2)=*(in+1022);
            *(temp+3)=*(in+1023);}
}
if(dlength!=wtotal) printf("\n\7Error-total data bytes written mismatch");
return 0;
}
```

```
/* PROGRAM inertial.c
Programmer: Dan Merfeld
Date: 9/15/89

This program opens six input files; 3 euler angle files and 3 euler rate
files. The euler angle files were created by the euler3f program, and the
euler rate files were created by the filter5 program.

The three output files are "h_vel.dat", "v_vel.dat", and "r_vel.dat". These
files represent the inertial rates about three orthogonal axes. These
calculations are discussed in section 2.1.1.3 of my thesis.

The program is called by the sequence "inertial XXX", where XXX is the
number of data points in each of the input data files. */

#include <c:\c\stdio.h>
#include <c:\c\io.h>
#include <c:\c\fcntl.h>
#include <c:\c\math.h>
#include <c:\c\errno.h>
#include <c:\c\stdlib.h>
#define  BLOCK    1024     /* Bytes to read and write each time */

float theta1 [BLOCK],theta2[BLOCK];
float th1_vel[BLOCK],th2_vel[BLOCK],th3_vel[BLOCK];
float h_vel[BLOCK],v_vel[BLOCK],r_vel[BLOCK]; /* 4 bytes per output */
main(argc,argv)
int argc;
char *argv[];
{
extern int errno;
unsigned long int dlength,wtotal;
float blength;
int j,k,fi1,fi2,fi3,fi4,fi5,fo1,fo2,fo3,nread,nwrite;
char out[40];
float sin_th1,cos_th1,sin_th2,cos_th2;
float pi,scale;

pi=3.1415926536;
if(argc!=2)
{        printf("\n\7Error - file length not specified");
         exit(-3);}
blength=atof(argv[1]);
dlength=blength;
printf("Finding velocity in inertial coordinates for %u values\n",dlength);

if((fi1=open("THETA1.DAT",O_RDONLY|O_BINARY))==-1) /* open all files */
   {printf("\n\7Error opening THETA1.DAT");
   exit(-1);}
if((fi2=open("THETA2.DAT",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening THETA2.DAT");
   exit(-1);}
if((fi3=open("TH1_VEL.DAT",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening TH1_VEL.DAT");
   exit(-1);}
if((fi4=open("TH2_VEL.DAT",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening TH2_VEL.DAT");
   exit(-1);}
```

```
if((fi5=open("TH3_VEL.DAT",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening TH3_VEL.DAT");
   exit(-1);}

fo1=open("H_VEL.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fo1==-1)
         {printf("\n\7Error opening H_VEL.DAT for output");
         if(errno==EEXIST)printf("\nH_VEL.DAT already exists, must delete it");
         exit(-2);}
fo2=open("V_VEL.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fo2==-1)
         {printf("\n\7Error opening V_VEL.DAT for output");
         if(errno==EEXIST)printf("\nV_VEL.DAT already exists, must delete it");
         exit(-2);}
fo3=open("R_VEL.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fo3==-1)
         {printf("\n\7Error opening R_VEL.DAT for output");
         if(errno==EEXIST)printf("\nR_VEL.DAT already exists, must delete it");
         exit(-2);}
scale=180./3.1415926536;
wtotal=0;
for(j=0;;j++)
{
   nread=read(fi1,theta1,BLOCK*4);
   nread=read(fi2,theta2,BLOCK*4);
   nread=read(fi3,th1_vel,BLOCK*4);
   nread=read(fi4,th2_vel,BLOCK*4);
   nread=read(fi5,th3_vel,BLOCK*4);
   for(k=0;k<nread/4;k++) /* the actual calculation are implemented below */
            {sin_th1=sin(*(theta1+k));
            cos_th1=cos(*(theta1+k));
            sin_th2=sin(*(theta2+k));
            cos_th2=cos(*(theta2+k));
            *(h_vel+k)=(*(th1_vel+k) - *(th3_vel+k)*sin_th2)*scale;
/*          *(h_vel+k)=(*(th1_vel+k))*scale; */
            *(v_vel+k)=((*(th2_vel+k))*cos_th1+(*(th3_vel+k))*cos_th2*sin_th1)*scale;
/*          *(v_vel+k)=((*(th2_vel+k)))*scale; */
            *(r_vel+k)=((*(th3_vel+k))*cos_th2*cos_th1-(*(th2_vel+k))*sin_th1)*scale;
}
   nwrite=write(fo1,h_vel,4*(nread/4)); /* 4 bytes per value */
   nwrite=write(fo2,v_vel,4*(nread/4)); /* 4 bytes per value */
   nwrite=write(fo3,r_vel,4*(nread/4)); /* 4 bytes per value */
    wtotal=wtotal+nwrite/4;
   if(nwrite!=4*(nread/4)) printf("\nError 7");
   if(nread!=BLOCK*4) break;
}
if(dlength!=wtotal) printf("\n\7Error-total data bytes written mismatch");
close(fi1);
close(fi2);
close(fi3);
close(fi4);
close(fi5);
close(fo1);
close(fo2);
close(fo3);
sprintf(out,"attrib -r h_vel.dat\0");
j=system(out);
sprintf(out,"attrib -r v_vel.dat\0");
```

```
j=system(out);
sprintf(out,"attrib -r r_vel.dat\0");
j=system(out);
}
```

```
/* PROGRAM filter9.c
Programmer: Dan Merfeld
Date: 9/15/89

This programs takes the output files from inertial.c and differentiates
each time series to determine the angular acceleration of the eye. This
information is used in the detection software which was implemented in
Matlab.

The output files are "h_acc.dat", "v_acc.dat", and "r_acc.dat".

The program is called with the sequence "filter9 XXX", where XXX is the
number of points in each of the input data files. */

#include <c:\c\stdio.h>
#include <c:\c\io.h>
#include <c:\c\fcntl.h>
#include <c:\c\math.h>
#include <c:\c\errno.h>
#include <c:\c\stdlib.h>
#define  BLOCK     1024      /* Bytes to read and write each time */

float in[BLOCK];   /* 4 bytes per input */
float out[BLOCK],temp[BLOCK+8];
main(argc,argv)
int argc;
char *argv[];
{
extern int errno;
int fd1,fd2,j,k;
char string[40];
unsigned long int dlength;
float blength;

if(argc!=2)
{          printf("\n\7Error - file length not specified");
           exit(-2);}
blength=atof(argv[1]);
dlength=blength;
printf("Filtering %u values",dlength);

if((fd1=open("H_VEL.DAT",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening H_VEL.DAT");
   exit(-1);}

fd2=open("H_ACC.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fd2==-1)
          {printf("\n\7Error opening H_ACC.DAT for output");
          if(errno==EEXIST) {printf("\nH_ACC.DAT already exists, delete it");}
          exit(-2);}
j=filt9(fd1,fd2,dlength);
if(j!=0)
          {printf("\n\7Error from filter function");
          exit(-1);}
close(fd1);
close(fd2);
sprintf(string,"attrib -r h_acc.dat\0");
k=system(string);
```

```
if((fd1=open("V_VEL.DAT",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening V_VEL.DAT");
   exit(-1);}

fd2=open("V_ACC.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fd2==-1)
         {printf("\n\7Error opening V_ACC.DAT for output");
         if(errno==EEXIST) {printf("\nV_ACC.DAT already exists, delete it");}
         exit(-2);}
j=filt9(fd1,fd2,dlength);
if(j!=0)
         {printf("\n\7Error from filter function");
         exit(-1);}
close(fd1);
close(fd2);
sprintf(string,"attrib -r v_acc.dat\0");
k=system(string);

if((fd1=open("R_VEL.DAT",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening R_VEL.DAT");
   exit(-1);}

fd2=open("R_ACC.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fd2==-1)
         {printf("\n\7Error opening R_ACC.DAT for output");
         if(errno==EEXIST) {printf("\nR_ACC.DAT already exists, delete it");}
         exit(-2);}
j=filt9(fd1,fd2,dlength);
if(j!=0)
         {printf("\n\7Error from filter function");
         exit(-1);}
close(fd1);
close(fd2);
sprintf(string,"attrib -r r_acc.dat\0");
k=system(string);
}


/* filt9: performs all of the calculations */
int filt9(fd1,fd2,dlength)
int fd1,fd2;
unsigned long int dlength;
{
unsigned long int wtotal;
int j,k,nread,nwrite;
int bias;
float sample;

sample=200.;
nread=read(fd1,in,16);
*(temp)=*(in);
*(temp+1)=*(in);
*(temp+2)=*(in);
*(temp+3)=*(in);
*(temp+4)=*(in);
*(temp+5)=*(in+1);
*(temp+6)=*(in+2);
```

```
*(temp+7)=*(in+3);
wtotal=0;
for(j=0;;j++)
        {nread=read(fd1,in,BLOCK*4);
        if(nread!=BLOCK*4)
                {*(in+nread/4+3)=*(in+nread/4-1);
                *(in+nread/4+2)=*(in+nread/4-1);
                *(in+nread/4+1)=*(in+nread/4-1);
                *(in+nread/4)=*(in+nread/4-1);
                nread=nread+16;}
   for(k=0;k<nread/4;k++)
           {*(temp+8+k)=*(in+k);
           *(out+k)=sample*((*(temp+8+k)-*(temp+k))/8);} /* actual filter */
   nwrite=write(fd2,out,4*(nread/4)); /* 4 bytes per value */
    wtotal=wtotal+nwrite/4;
   if(nwrite!=4*(nread/4)) printf("\nError 7");
   if(nread!=BLOCK*4)
   {break;}
    else
           {*temp=*(in+1016);
           *(temp+1)=*(in+1017);
           *(temp+2)=*(in+1018);
           *(temp+3)=*(in+1019);
           *(temp+4)=*(in+1020);
           *(temp+5)=*(in+1021);
           *(temp+6)=*(in+1022);
           *(temp+7)=*(in+1023);}
}
if(dlength!=wtotal) printf("\n\7Error-total data bytes written mismatch");
return 0;
}
```

```
/* PROGRAM acc_mag.c
Programmer: Dan Merfeld
Date: 9/15/89

This program calculates the magnitude of the acceleration by finding
the square root of the sum of squares of the components.

The input files are the output of filter9.c.

The output file is "m_acc.dat"

The program should be called with the sequence "acc_mag XXX", where XXX is
the number of points in each of the input files. */

#include <c:\c\stdio.h>
#include <c:\c\io.h>
#include <c:\c\fcntl.h>
#include <c:\c\math.h>
#include <c:\c\errno.h>
#include <c:\c\stdlib.h>
#define   BLOCK     1024      /* Bytes to read and write each time */

float m_acc[BLOCK];
float h_acc[BLOCK],v_acc[BLOCK],r_acc[BLOCK]; /* 4 bytes per output */
main(argc,argv)
int argc;
char *argv[];
{
extern int errno;
unsigned long int dlength,wtotal;
float blength;
int j,k,fi1,fi2,fi3,fo1,nread,nwrite;
char out[40];
float temp1,temp2,temp3;

if(argc!=2)
{         printf("\n\7Error - file length not specified");
          exit(-2);}
blength=atof(argv[1]);
dlength=blength;
printf("Finding acceleration magnitude for %u values",dlength);

if((fi1=open("H_ACC.DAT",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening H_ACC.DAT");
   exit(-1);}
if((fi2=open("V_ACC.DAT",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening V_ACC.DAT");
   exit(-1);}
if((fi3=open("R_ACC.DAT",O_RDONLY|O_BINARY))==-1)
   {printf("\n\7Error opening R_ACC.DAT");
   exit(-1);}

fo1=open("M_ACC.DAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fo1==-1)
          {printf("\n\7Error opening M_ACC.DAT for output");
          if(errno==EEXIST)printf("\nM_ACC.DAT already exists, must delete it");
          exit(-2);}
wtotal=0;
for(j=0;;j++)
{
   nread=read(fi1,h_acc,BLOCK*4);
   nread=read(fi2,v_acc,BLOCK*4);
   nread=read(fi3,r_acc,BLOCK*4);
   for(k=0;k<nread/4;k++)
       {temp1=(*(h_acc+k))*(*(h_acc+k));/*These four lines calculate */
           temp2=(*(v_acc+k))*(*(v_acc+k));
           temp3=(*(r_acc+k))*(*(r_acc+k));
           *(m_acc+k)=sqrt(temp1+temp2+temp3);}
   nwrite=write(fo1,m_acc,4*(nread/4)); /* 4 bytes per value */
    wtotal=wtotal+nwrite/4;
   if(nwrite!=4*(nread/4)) printf("\nError 7");
   if(nread!=BLOCK*4) break;
}
if(dlength!=wtotal) printf("\n\7Error-total data bytes written mismatch");
close(fi1);
close(fi2);
close(fi3);
close(fo1);
sprintf(out,"attrib -r m_acc.dat\0");
j=system(out);
}
```

```
/* PROGRAM f_convert.c
Programmer: Dan Merfeld
Date: 11/17/89

This program opens a data file called "temp.dat" and converts it to Matlab
format and calls the output file "temp.mat". The input file should be an
array of floating values like that created by "euler3f.c", "inertial.c",
"acc_mag.c", etc.

The program should be called with the sequence "f_convert YYY XXX", where
YYY is the matlab variable name and XXX is the number of data values in
the input file. */

#include <c:\c\stdio.h>
#include <c:\c\io.h>
#include <c:\c\fcntl.h>
#include <c:\c\math.h>
#include <c:\c\errno.h>
#include <c:\c\stdlib.h>
#define  BLOCK    512      /* Bytes to read and write each time */

/* These must be variables to provide pointers for write functions */
long int namlen;         /* length of variable name + 1 */
long int type=000;       /* matrix, double prec, column-wise */
long int ncols=1;        /* number of columns */
long int imagf=0;        /* no imaginary part */

float datain[BLOCK];   /* 2 bytes per input integer */
double dataout[BLOCK];     /* 8 bytes per output double */

main(argc,argv)
int argc;
char *argv[];
{
extern int errno;
unsigned long int dlength,wtotal;
float blength;
int j,k,fd1,fd2,nread,nwrite;
char out[40];

if(argc!=3)
          {printf("\n\7Error - variable name and/or file length not specified");
          exit(-3);}
namlen=strlen(argv[1])+1;
blength=atof(argv[2]);
dlength=blength;
printf("\nTranslating TEMP.DAT; %u integers",dlength);
if((fd1=open("TEMP.DAT",O_RDONLY|O_BINARY))==-1)
    {printf("\n\7Error opening TEMP.DAT");
    exit(-1);}
fd2=open("TEMP.MAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fd2==-1)
          {printf("\n\7Error opening TEMP.MAT for output");
          if(errno==EEXIST) printf("\nTEMP.MAT already exists, must delete it");
          exit(-2);}
nwrite=write(fd2,&type,4);            /* write all 5 header values */
if (nwrite!=4) printf("\nError 1");
nwrite=write(fd2,&dlength,4);
if (nwrite!=4) printf("\nError 2");
nwrite=write(fd2,&ncols,4);
if (nwrite!=4) printf("\nError 3");
nwrite=write(fd2,&imagf,4);
if (nwrite!=4) printf("\nError 4");
nwrite=write(fd2,&namlen,4);
if (nwrite!=4) printf("\nError 5");
nwrite=write(fd2,argv[1],namlen);
if (nwrite!=namlen) printf("\nError 6");
wtotal=0;
for(j=0;;j++)
     {
    nread=read(fd1,datain,BLOCK*4);
    for(k=0;k<nread/4;k++) dataout[k]=datain[k];
    nwrite=write(fd2,dataout,8*(nread/4));  /* 8 bytes per value */
     wtotal=wtotal+nwrite/8;
    if(nwrite!=8*(nread/4)) printf("\nError 7");
    if(nread!=BLOCK*4) break;
     }
if(dlength!=wtotal) printf("\n\7Error total data bytes written mismatch");
close(fd1);
close(fd2);
sprintf(out,"attrib -r temp.mat\0");
j=system(out);
}
```

```
/* PROGRAM i_convert.c
Programmer: Dan Merfeld
Date: 11/17/89

This program open a data file called "temp.dat" and converts it to Matlab
format and calls the output file "temp.mat". The input file should be an
array of short integer values.

The program should be called with the sequence "i_convert YYY XXX", where
YYY is the matlab variable name and XXX is the number of data values in the
input file.  /*

#include <c:\c\stdio.h>
#include <c:\c\io.h>
#include <c:\c\fcntl.h>
#include <c:\c\math.h>
#include <c:\c\errno.h>
#include <c:\c\stdlib.h>
#define  BLOCK    512       /* Bytes to read and write each time */

/* These must be variables to provide pointers for write functions */
long int namlen;          /* length of variable name + 1 */
long int type=000;        /* matrix, double prec, column-wise */
long int ncols=1;         /* number of columns */
long int imagf=0;         /* no imaginary part */

short int datain[BLOCK];   /* 2 bytes per input integer */
double dataout[BLOCK];     /* 8 bytes per output double */

main(argc,argv)
int argc;
char *argv[];
{
extern int errno;
unsigned long int dlength,wtotal;
float blength;
int j,k,fd1,fd2,nread,nwrite;
char out[40];

if(argc!=3)
          {printf("\n\7Error - variable name and/or file length not specified");
          exit(-3);}
namlen=strlen(argv[1])+1;
blength=atof(argv[2]);
dlength=blength;
printf("\nTranslating TEMP.PRN; %u integers",dlength);
if((fd1=open("TEMP.PRN",O_RDONLY|O_BINARY))==-1)
    {printf("\n\7Error opening TEMP.PRN");
    exit(-1);}
fd2=open("TEMP.MAT",O_CREAT|O_EXCL|O_WRONLY|O_BINARY);
if(fd2==-1)
          {printf("\n\7Error opening TEMP.MAT for output");
          if(errno==EEXIST) printf("\nTEMP.MAT already exists, must delete it");
          exit(-2);}
nwrite=write(fd2,&type,4);           /* write all 5 header values */
if (nwrite!=4) printf("\nError 1");
nwrite=write(fd2,&dlength,4);
if (nwrite!=4) printf("\nError 2");
nwrite=write(fd2,&ncols,4);
if (nwrite!=4) printf("\nError 3");
nwrite=write(fd2,&imagf,4);
if (nwrite!=4) printf("\nError 4");
nwrite=write(fd2,&namlen,4);
if (nwrite!=4) printf("\nError 5");
nwrite=write(fd2,argv[1],namlen);
if (nwrite!=namlen) printf("\nError 6");
wtotal=0;
for(j=0;;j++)
    {
   nread=read(fd1,datain,BLOCK*2);
   for(k=0;k<nread/2;k++) dataout[k]=datain[k];
   nwrite=write(fd2,dataout,8*(nread/2));  /* 8 bytes per value */
    wtotal=wtotal+nwrite/8;
   if(nwrite!=8*(nread/2)) printf("\nError 7");
   if(nread!=BLOCK*2) break;
    }
if(dlength!=wtotal) printf("\n\7Error total data bytes written mismatch");
close(fd1);
close(fd2);
sprintf(out,"attrib -r temp.mat\0");
j=system(out);
}
```

B.3 Matlab Analysis Programs

The actual desaccade algorithm is implemented in a set of m-files that can only be called from within Matlab (copyright MathWorks). Figure B.3 shows the tree structure of all of the routines which I wrote as part of this analysis process. "Desaccade.m" is called directly from the "spv.m" program. It's primary function is to call each of the subroutines which perform the actual calculations.

The first subroutine called is detect1.m. This program and it's subroutines (heart.m, remove.m and fill.m) performs a 3 dimensional acceleration detection similar, though not identical, to the 1 dimensional Massoumnia algorithm.

The second subroutine called is decimat.m. This program decimates the data from the original 200 Hz sampling rate to 20 Hz.

The third subroutine called is beg_end.m It is not important.

The fourth subroutine is detect2.m. This is the part of the program which performs the parameter estimation and evaluates the residuals to find any missed saccades.

The fifth subroutine is low_pass.m, and, true to its name, it low pass filters the output data to eliminate some of the high frequency noise.

The final subroutine called is hrdcopy.m. The program plots each of the data channels and generates a paper copy.

Source code for each of these m-files follows.

Desaccade.m Process Flow

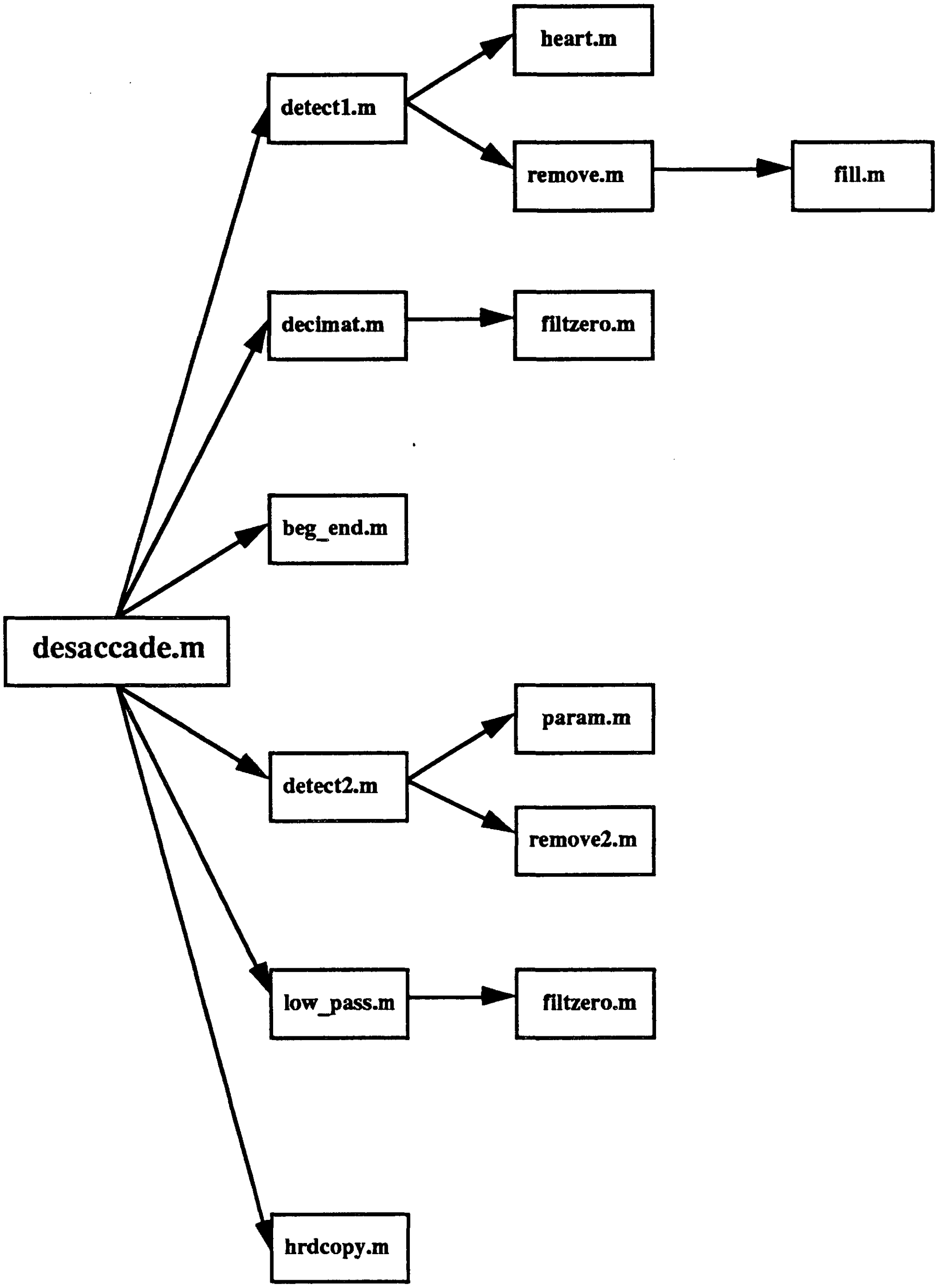


Figure B.3

```
%DESACCADE          This program detects fast phases and removes them.  It does
%       this in two passes at the data.  First, the data is analyzed by a
%       program called DETECT1.  Then a program called DECIMAT is called to
%       decimate the data to 20 Hz.  The second pass at the data is performed
%       by a program called DETECT2.  Another program called BEG_END is also
%       called to find the begining and ending of the stimulus acceleration.
%       A program called LOW_PASS is invoked to low pass filter the slow
%       phase eye velocity.  And, finally, a program called HRDCOPY is used
%       to plot the data to the screen and create a hardcopy output of the
%       data.
%
%       D.M. Merfeld  7/9/89

fprintf('running detect1\n');
detect1


decimat

load temp1
load temp1_1
load temp2
load temp1_2
load temp3
load temp1_3
load temp4

beg_end

detect2

low_pass

hrdcopy

h_vel1=h_vel;
v_vel1=v_vel;
r_vel1=r_vel;
h_vel=h_vel_f;
v_vel=v_vel_f;
r_vel=r_vel_f;

clear h_vel1 v_vel1 r_vel1
clear h_v v_v r_v
clear h_vel_f v_vel_f r_vel_f
clear h_vel2 v_vel2 r_vel2
clear ans
load flag
save temp

clearall
```

```
%DETECT1      This program uses an acceleration threshold to detect fast
%         phases.  The actual detection algorithm is in the subroutine HEART.
%         The program also calls REMOVE which takes the periods with detected
%         fast phases and fills the region with an estimate of slow phase
%         velocity.

%         D.M. Merfeld 3/30/89
%         Edited 7/9/89

load c:\data\thresh1.mat
load m_acc
thresh_acc
thresh_end

stop=max(size(m_acc))-151;
m=151;
finish=0;
event=0;
pack

heart
save flag flag thresh_acc thresh_end

clearall

remove
```

```
%HEART          Detects the majority of fast phases using an acceleration threshold
%        set by the user.  The subroutine finds the start and finish of fast
%        phases.  The variable [start] marks the last good data point before
%        the fast phase, while the variable [finish] marks the last data point
%        of the fast phase.

%        D.M. Merfeld 3/30/89

while (m <= stop)

   if (m_acc(m) >= thresh_acc),
                k=1;
                while(min(m_acc(m-k:m-k-2))>thresh_end)
                        k=k+1;
                end
                if((m-k-2) > finish),
                        event=event+1;
                        start=m-k-1;
                end

                m=m+5;
                while(min(m_acc(m:m+2))>thresh_end)
                        m=m+1;
                end

                m=m+2;
                finish=m;
                flag(event,:)=[start finish];
   end
   m=m+1;
end
```

```
%REMOVE     Takes the events which have been classified as fast phases and removes
%           them to yield slow phase eye velocity.  The fast phases which are
%           removed are those found by the heart program. The program fill is
%           called to perform the actual interpolation.


%           D.M. Merfeld 3/30/89

load flag
load h_vel
data='h_vel';
len=length(h_vel);
pack
fill
save h_vel_slo h_vel
clearall

load flag
load v_vel
data='v_vel';
len=length(v_vel);
pack
fill
save v_vel_slo v_vel
clearall

load flag
load r_vel
data='r_vel';
len=length(r_vel);
pack
fill
save r_vel_slo r_vel

clearall
```

```
%FILL   This subroutine estimates the slow phase velocity during a fast phase
%       to be equal to the slow phase velocity just before the saccade.  This
%       estimate of the slow phase velocity is used to fill in the duration of
%       the fast phase.

%       D.M. Merfeld 3/30/89

finish=0;
number=max(size(flag));
n=1;
while (n <= number),
        start=flag(n,1);
        finish=flag(n,2);
        eval(['v_beg=median(',data,'(start-4:start));']);
        eval([data,'(start+1:finish)=v_beg*ones(finish-start,1);']);
        n=n+1;
end
```

```
%DECIMAT This program is used to decimate the data stream from
%        200 Hz to 20 Hz.  This program was specifically designed
%        for the coil measurements that I obtained from squirrel
%        monkeys.  With this data I didn't need to filter the data
%        before decimating.  BEWARE of aliasing.

load h_vel
len=length(h_vel);
h_v=h_vel(1:10:len);
save temp1_1 h_v
clear h_v

load h_vel_slo
len=length(h_vel);
h_vel=h_vel(1:10:len);
save temp1 h_vel
clear h_vel

load v_vel
len=length(v_vel);
v_v=v_vel(1:10:len);
save temp1_2 v_v
clear v_v

load v_vel_slo
len=length(v_vel);
v_vel=v_vel(1:10:len);
save temp2 v_vel
clear v_vel

load r_vel
len=length(r_vel);
r_v=r_vel(1:10:len);
save temp1_3 r_v
clear r_v

load r_vel_slo
len=length(r_vel);
r_vel=r_vel(1:10:len);
save temp3 r_vel
clear r_vel

load omega
len=length(omega);
x=omega(1:2:len);
clear omega
B=[1/5 1/5 1/5 1/5 1/5];
A=1;
filtzero
len=length(x);
x=x(1:5:len);
B=[1 0 -1];
filtzero
alpha=x;
save temp4 alpha
clearall
```

```
%FILTZERO Performs zero phase shift filtering for FIR filters (ONLY!)
%	by padding signal to be filtered with non-zero values at
%	beginning and end of data sequence.

%	D.M. Merfeld 3/30/89

nx=max(size(x));   %this section of code calculates the eye velocity
nB=max(size(B));
x(nx+1:nx+((nB-1)/2))=x(nx).*ones(((nB-1)/2),1);
[temp,z]=filter(B,A,x(1).*ones(((nB-1)/2),1));
clear temp
x=filter(B,A,x,z);
x=x(((nB-1)/2+1):nx+((nB-1)/2));
clear z nx nB
```

```
%BEG_ENG This program detects the beginning and ending of acceleration
%        and deceleration for trapezoidal profiles.  This is not an
%        important program and probably can be removed.

len=length(alpha);
count=0;
j=1;
while(j<len-20)
        if(abs(mean(alpha(j:j+10))) > 2)
                k=j+5;
                sgn2=sign(alpha(k));
                sgn1=sign(mean(alpha(j:j+10)));
                while(sgn1 == sgn2)
                        k=k-1;
                        if(k < 10)
                                break;
                        end
                        sgn2=sign(alpha(k));
                end
                i=j+6;
                sgn2=sign(mean(alpha(i:i+10)));
                while(sgn1 == sgn2)
                        i=i+1;
                        if(i > len-10)
                                break;
                        end
                        sgn2=sign(alpha(i));
                end
                if((i - k) > 15)
                        count=count+1;
                        beg(count)=k + 1;
                        stop(count)=i - 1;
                        j=i+1;
                end
        end
        j=j+1;
end
if(count > 0)
        alpha(1:beg(1)-1) = zeros(1,beg(1)-1);
        for j=1:(count-1)
                alpha(stop(j)+1:beg(j+1)-1)=zeros(1,beg(j+1)-stop(j) - 1);
        end
        alpha(stop(count)+1:len)=zeros(1,len-stop(count));
end

clear len count j  k ans i sgn1 sgn2
```

```
%DETECT2 This program uses a least squares fit parameter estimation
%        algorithm. The program calls two other programs: PARAM and REMOVE2
%        PARAM estimates a model for the slow phase eye velocity and
%        determines the errors. REMOVE2 eliminates the detected saccades.

len=length(h_vel);
h_vel2=h_vel;
v_vel2=v_vel;
r_vel2=r_vel;

load c:\data\thresh2
vel_test

param   %Estimate a model for the SPV and determine the prediction errors

remove2 %Eliminate the detected saccades

clear ans len
```

```
%PARAM		This program divides the original data sequence into ten
%	sequences each sampled at 2 Hz. The parameter estimation
%	is performed on this low frequency rate data. The output of
%	the program is a set of residuals which will be compared to
%	a threshold set by the user to indicate an "event".

e_x_f(len)=0;
e_y_f(len)=0;
e_z_f(len)=0;
%e_x_b(len)=0;
%e_y_b(len)=0;
%e_z_b(len)=0;


for index=1:10
	u=alpha(20+index:10:len-20)';
	x=h_vel(20+index:10:len-20)';
	y=v_vel(20+index:10:len-20)';
	z=r_vel(20+index:10:len-20)';

	[th_x,r_x]=ls(x,u,1,1,1);
	[th_y,r_y]=ls(y,u,1,1,1);
	[th_z,r_z]=ls(z,u,1,1,1);

	e_x_f(30+index:10:len-20)=r_x;
	e_y_f(30+index:10:len-20)=r_y;
	e_z_f(30+index:10:len-20)=r_z;
end

test_f= sqrt(e_x_f.^2 + e_y_f.^2 + e_z_f.^2);

clear u u_b x x_b y y_b z z_b th_x th_x_b th_y th_y_b th_z th_z_b
clear r_x r_x_b r_y r_y_b r_z r_z_b index e_x_b e_y_b e_z_b len2
```

```
%REMOVE2 This program evaluates the residual errors determined in the
%        PARAM subroutine and interpolates the "bad" data.

number=0;
j=10;
while(j  < len-15)
        if(min([test_f(j) test_f(j+10)]) >= vel_test);  %Point > threshold
                number=number+1;
                start=j;
                count=j+1;
                while(min([test_f(count) test_f(count+10)]) >= vel_test)
                        count=count+1;
                end
                finish=count;
                for index=j:count-1
                        h_vel2(index)=h_vel2(index)-e_x_f(index);
                        v_vel2(index)=v_vel2(index)-e_y_f(index);
                        r_vel2(index)=r_vel2(index)-e_z_f(index);
                        test_f(j+10)=0;
                end
                j=count;
        end
j=j+1;
end

clear j start count finish index e_x_f e_y_f e_z_f
clear test_f
```

```
%LOW_PASS This program low-pass filters the three components of eye velocity
%         The filter coefficients come from a file called lpf.mat  FILTZERO is
%         called to perform the actual zero-phase filtering.

load c:\data\lpf

x=h_vel2;
filtzero
h_vel_f=x;

x=v_vel2;
filtzero
v_vel_f=x;

x=r_vel2;
filtzero
r_vel_f=x;

clear A B x
```

```
%HRDCOPY This program provides plots of each of the three components of slow
%        phase eye velocity during various stages of processing. The first
%        plot is eye velocity, the second plot is after running the detect1
%        algorithm, the third plot is after running detect2, and the fourth
%        plot is the final output data.

len=length(h_vel);
time=0:.05:(len-41)*.05;
clg

%if(sum(count) < 0)
%        !print warning.fil
%        pause(10);
%end

subplot(222),plot(time,h_vel(21:len-20))
title('H_VEL AFTER PASS 1')
axis;
subplot(223),plot(time,h_vel2(21:len-20))
title('H_VEL AFTER PASS 2')
subplot(224),plot(time,h_vel_f(21:len-20))
title('H_VEL AFTER LPF')
subplot(221),plot(time,h_v(21:len-20))
title('H_VEL BEFORE')
axis;
prtsc('ff');
!print file.nam
clg

subplot(222),plot(time,v_vel(21:len-20))
title('V_VEL AFTER PASS 1')
axis;
subplot(223),plot(time,v_vel2(21:len-20))
title('V_VEL AFTER PASS 2')
subplot(224),plot(time,v_vel_f(21:len-20))
title('V_VEL AFTER LPF')
subplot(221),plot(time,v_v(21:len-20))
title('V_VEL BEFORE')
axis;
prtsc('ff');
!print file.nam
clg

subplot(222),plot(time,r_vel(21:len-20))
title('R_VEL AFTER PASS 1')
axis;
subplot(223),plot(time,r_vel2(21:len-20))
title('R_VEL AFTER PASS 2')
subplot(224),plot(time,r_vel_f(21:len-20))
title('R_VEL AFTER LPF')
subplot(221),plot(time,r_v(21:len-20))
title('R_VEL BEFORE')
axis;
prtsc('ff');
!print file.nam
clg
clear len ans time
```

APPENDIX C

EYE MOVEMENT CALIBRATION

The following calibration procedure is slightly adapted from Appendix A-3 of the VRF centrifuge standard operating procedures.

Section 1. EYE MOVEMENT DETECTOR

The procedure presented in this section describe the hardware, steps and actions required to conduct an eye movement recording experiment. Present experiment specific equipent and dedicated science support equipment consists of the Eye Movement Detectors (EMD1 and EMD2) and associated preamplifiers, a Haffler two channel 100 watt amplifier, the buffer amplifier chassis, two EMD Auxilliary chassis (EMD-AC1 and EMD-AC2), test coil calibration jig, Helmholtz field coils, and single cell unit recording instrumentation. General purpose science support equipment required includes oscilloscopes, digital volt meters (DVMs), and a strip chart recorder.

CAUTION

EMD1 and the Haffler amplifier must NOT be activated unless the field coils are plugged in. DAMAGE TO EMD1 WILL RESULT IF THE SYSTEM IS RUN WITHOUT A LOAD. Do not activate the EMD preamplifier until the sensor coils are plugged in.

Part A: Calibration Procedures

MECHANICAL SET-UP

____ 1. Mount calibration jig and field coil assembly on the plastic platform.
____ 2. Screw the eye coil bridge fixture to the top of the field coil assembly.
____ 3. Clamp eye coil lead ends to bridge clamp fixture at the top of the field coil assembly.
____ 4. Bolt platform onto dovetail.
____ 5. Place dovetail and platform assembly into STC.
____ 6. Tighten dovetail locking screw.
____ 7. Connect all signal wires.
____ 8. Level STC to desired position.

ELECTRICAL SET-UP

____ 1. Connect field coil to amplifier outputs.
____ 2. Connect calibrator to EMD pre-amplifier inputs. (Verify that the polarity of the leads is correct.)
____ 3. Switch on 1) EMD1
2) Haffler amplifier
3) EMD1 pre-amplifier
4) Buffer amplifers
5) EMD Auxilliary Chassis #1
6) EMD Auxilliary Chassis #2

____________________________NOTE____________________________

The EMD system requires 30 to 60 minutes of warm-up for proper operation.

____ 4. Monitor field coil currents for 1 volt amplitude peak to peak at EMD1 Channel 1 monitor and EMD1 Channel 2 monitor outputs. (Channel 1 should lead channel 2 by about 90 degrees.) To adjust the field coil current amplitudes, use the potentiometers labelled "AGC adjust H and V" on the EMD1 cards.

____ 5. Check eye coil voltages at EMD1 Channel 1 "front" and EMD1 Channel 2 "side coil" inputs to signal demodulator (EMD1). Verify that there are two sinusoids there. This indicates that the pre-amplifier is functioning. (Phase is not important at this point.) If periodic maintenance is required or if there are problems with the system, then perform step 6, 'phase calibration of EMD1'; otherwise proced to step 7, 'Cross-bleed adjustment'.

____ 6. Phase calibration of EMD1

a. Put EMD1 channel 1 "front coil" output on scope and put pulse monitor on a second channel. Set pulse monitor switch selector on back of EMD1 to channel 1H.

b. Adjust test coil with pitch and yaw coarse adjustment so that channel 1 output is zero.

c. Introduce a downward pitch offset on the test coil. Using the EMD1 "Phase H adjust" potentiometer on the right eye card, adjust the pulse to be in phase with channel 1's output.

d. Rezero until channel 1's output is zero again. Select pulse monitor switch selector on back of EMD1 to channel 1V. Introduce a rightward yaw on the test coil. With the EMD1 "Phase V adjust" potentiometer on the right eye card, adjust the pulse to be in phase with the channel 1 output (see Figure 1). When finished rezero until channel 1 output is flat.

e. Display channel 2 "side coil" output. Select pulse monitor switch to channel 2H. Adjust the EMD1 "Phase H adjust" potentiometer on the left eye (bottom) card, so that sine wave positive peak is at the trailing edge of the square wave.

f. IMPORTANT. Reset the pulse monitor switch on the back of the EMD1 to channel 2V. Calibration will not be correct unless this step is done.

____ 7. Cross-bleed adjustment

a. Make sure that the outputs of the appropriate buffer amplifiers are connected to the strip chart recorder. Place the calibration coil into the 'geometric zero' position.

b. Zero the horizontal, vertical and torsional channels using the position potentiometer on the front of EMD1.

c. Pitch the test coil with the horizontal position zeroed, using the coarse calibration drive, and adjust the phase potentiometers on the front of the horizontal channel of the EMD1 until the horizontal trace is flat. At the same time, adjust the gain of the torsional channel cross-bleed circuit (EMD-AC2) so that the torsional channel output is flat.

d. Yaw the test coil with the vertical position zeroed and adjust the phase potentiometers on the front of the vertical channel on the EMD1 so that the vertical trace is flat. If reduced by adjusting the torsional EMD1 channel phase potentiometer, and then re-zeroing using the torsional channel position potentiometer.

e. Rezero the test coil in the jig before continuing.

____ 8. Calibrate the system with the STC covered.

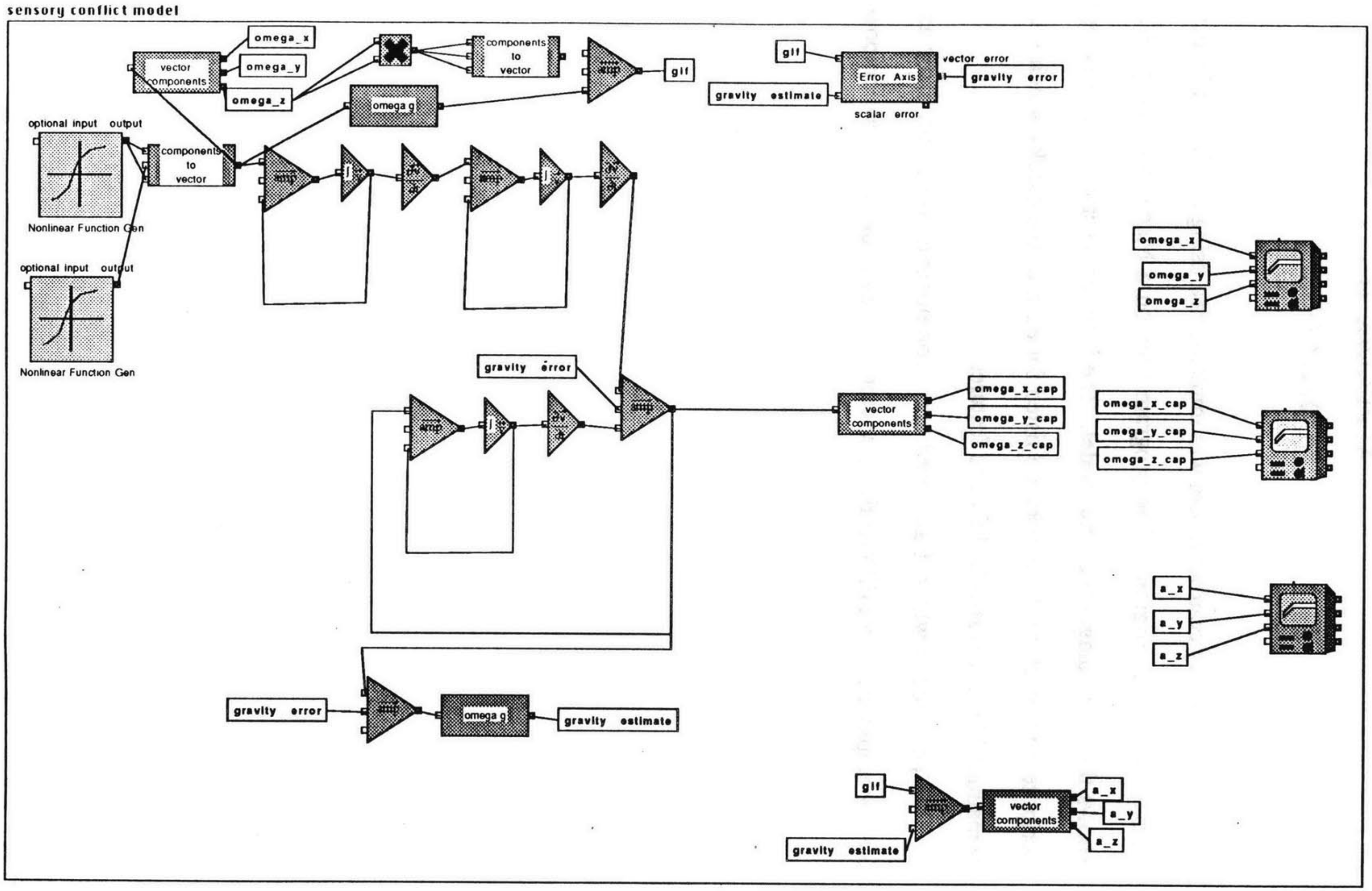
a. With the test coil zeroed, zero everything using the DVMs. Zeroing should be done using the position potentiometers on the front of EMD1.

b. Calibrate the pitch, yaw, and roll coil outputs. Place the test coil sequentially to +10 degrees, 0 degrees, and - 10 degrees in pitch, yaw, and roll. The outputs on the DVMs should +4V, 0V, -4V respectively for pitch and yaw. (For the roll channel 10 degrees is defined by 7 turns of the vernier.) Any voltages which require adjustment should be adjusted at this point using the gain potentiometers on the buffer amplifiers.

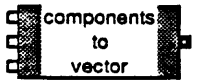
APPENDIX D
MODEL CODE

The sensory conflict model was implemented using the Extend (copyright Imagine That) simulation program. The program runs on a Macintosh computer. Figure D.1 shows the block diagram of the model. The icon, connector names, dialog box, and the script text for each of the blocks are listed for each of the blocks which I authored. The remaining blocks are standard Extend features.

The blocks which I authored are: components to vector, error axis, omega g, vector amplifier, vector differentiator, vector integrator, and vector to components.


sensory conflict model
vector components
omega_x
omega_y
omega_z
components to vector
gif
omega g
optional input output
Nonlinear Function Gen
optional input output
Nonlinear Function Gen
components to vector
gif
gravity estimate
Error Axis
vector error
gravity error
scalar error
omega_x
omega_y
omega_z
gravity error
vector components
omega_x_cap
omega_y_cap
omega_z_cap
omega_x_cap
omega_y_cap
omega_z_cap
a_x
a_y
a_z
gravity error
omega g
gravity estimate
gif
gravity estimate
vector components
a_x
a_y
a_z

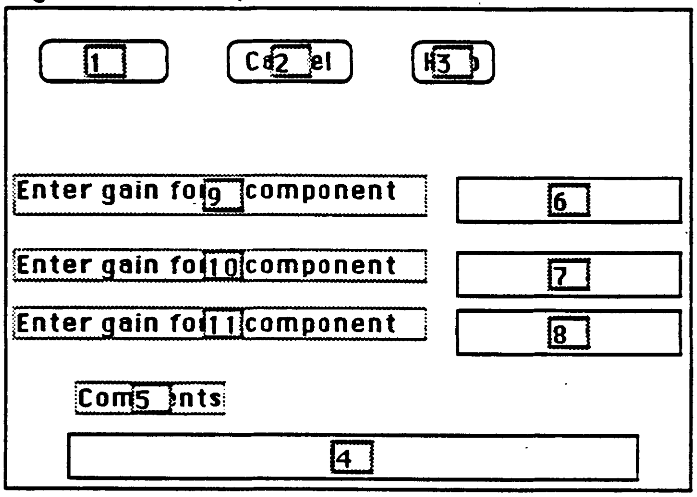
Icon of block components to vector



Connectors of block components to vector

xIn
vectorOut
yIn
zIn

Dialog of block components to vector


1
Ca2 el
H3)
Enter gain fo 9 component
6
Enter gain fo 10 component
7
Enter gain fo 11 component
8
Com5 nts
4

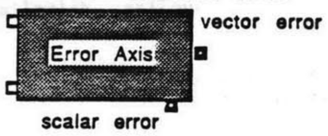
User messages of block components to vector

<1>	OK
<2>	Cancel
<3>	Help
<4>	Comments
<5>	
<6>	gain1
<7>	gain2
<8>	gain3
<9>	
<10>	
<11>	

Script of block components to vector

```
real dummy [];


** This message occurs for each step in the simulation.
on simulate
{

** put the three components into a vector
dummy[0] = xIn*gain1;
dummy[1] = yIn*gain2;
dummy[2] = zIn*gain3;

vectorOut = passArray(dummy);
}


** If the dialog data is inconsistent for simulation, abort.
on checkdata
{

}


** Initialize any simulation variables.
on initsim

{

makearray(dummy, 3);

}
on createBlock
{
gain1=1.0;
gain2=1.0;
gain3=1.0;
}

** User clicked the dialog HELP button.
on help
{
showHelp();
}
```

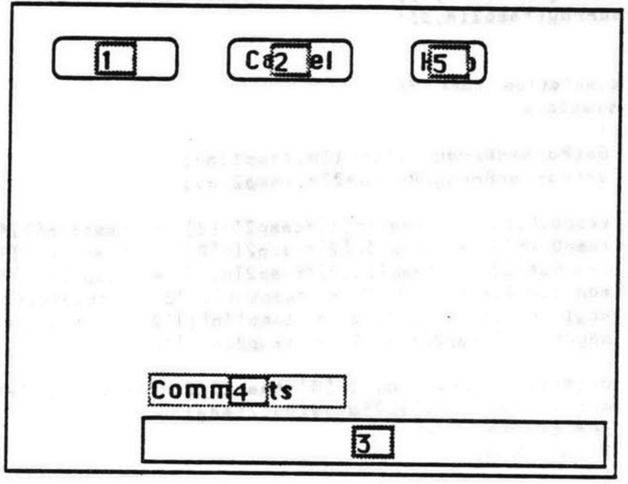
Icon of block error axis



Connectors of block error axis

Vector1In
Vector2In
vectorOut
scalarOut

Dialog of block error axis



User messages of block error axis

<1>	OK
<2>	Cancel
<3>	Comments
<4>	
<5>	help

Script of block error axis

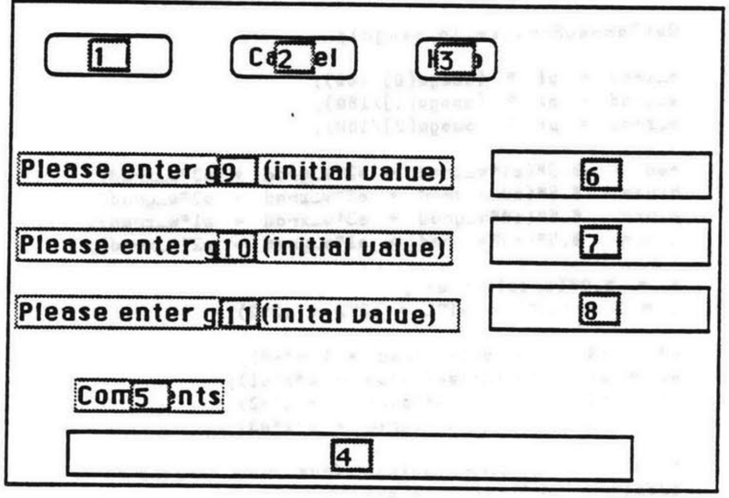
```
** cross product */
** totalSteps, curStep, totalTime, curTime, deltaTime */
** are defined by the system */

real        tempOut[],mag,mag1,mag2,dot,angle;
real            temp1In[],temp2In[];


on checkdata
{

}

on initsim
{
makearray(tempOut,3);
makearray(temp1In,3);
makearray(temp2In,3);
}

** simulation part */
on simulate
    {
    GetPassedArray(Vector1In,temp1In);
    GetPassedArray(Vector2In,temp2In);

    tempOut[0]  =  temp1In[1]*temp2In[2]  -  temp1In[2]*temp2In[1];
    tempOut[1]  =  temp1In[2]*temp2In[0]  -  temp1In[0]*temp2In[2];
    tempOut[2]  =  temp1In[0]*temp2In[1]  -  temp1In[1]*temp2In[0];
    mag=sqrt(tempOut[0]^2  +  tempOut[1]^2  +  tempOut[2]^2);
    mag1=sqrt(temp1In[0]^2  +  temp1In[1]^2  +  temp1In[2]^2);
    mag2=sqrt(temp2In[0]^2  +  temp2In[1]^2  +  temp2In[2]^2);

    dot=temp1In[0]*temp2In[0]+temp1In[1]*temp2In[1]+temp1In[2]*temp2In[2];
    angle=180/3.1415927*acos(dot/(mag1*mag2));
    scalarOut=angle;
    if(mag > 0)
    {
        tempOut[0]=tempOut[0]*angle/mag;
        tempOut[1]=tempOut[1]*angle/mag;
        tempOut[2]=tempOut[2]*angle/mag;
    }
    else
    {
        tempOut[0]=0;
        tempOut[1]=0;
        tempOut[2]=0;
    }
    vectorOut  =  PassArray(tempOut);
    }


on help
{
showHelp();
}
```

Icon of block omega g

Connectors of block omega g

g_out
w_in

Dialog of block omega g


1
Ca2 el
H3 p
Please enter g9 (initial value)
6
Please enter g10 (initial value)
7
Please enter g11 (inital value)
8
Com5 ents
4

User messages of block omega g

<1>	OK
<2>	Cancel
<3>	Help
<4>	Comments
<5>	
<6>	g_x
<7>	g_y
<8>	g_z
<9>	
<10>	
<11>	

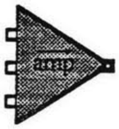
Script of block omega g

```
real  dcos[],g_body[];
real  e0,e1,e2,e3;
real  w_xrad,w_yrad,w_zrad;
real  omega[];
** This message occurs for each step in the simulation.
on simulate
{

**declare variables

real  red,blue,pink,grey;
real  k,z;

GetPassedArray(w_in,omega);

w_xrad  =  pi  *  (omega[0]/180);
w_yrad  =  pi  *  (omega[1]/180);
w_zrad  =  pi  *  (omega[2]/180);

red  =  -0.5*(e1*w_xrad  +  e2*w_yrad  +  e3*w_zrad);
blue=    0.5*(e0*w_xrad  +  e2*w_zrad  -  e3*w_yrad);
pink=    0.5*(e0*w_yrad  +  e3*w_xrad  -  e1*w_zrad);
grey=    0.5*(e0*w_zrad  +  e1*w_yrad  -  e2*w_xrad);

k  =  0.9*(1/deltatime);
z  =  1  -  (e0^2  +  e1^2  +  e2^2  +  e3^2);

e0  =  e0  +  deltatime*(red  +  k*z*e0);
e1  =  e1  +  deltatime*(blue  +  k*z*e1);
e2  =  e2  +  deltatime*(pink  +  k*z*e2);
e3  =  e3  +  deltatime*(grey  +  k*z*e3);

** Calculate dirn cosines from quarternions
dcos[0]  =e0^2+e1^2-e2^2-e3^2;
dcos[1]  =2*(e1*e2+e0*e3);
dcos[2]  =2*(e1*e3-e0*e2);

dcos[3]  =2*(e1*e2-e0*e3);
dcos[4]=e0^2-e1^2+e2^2-e3^2;
dcos[5]  =2*(e2*e3+e0*e1);

dcos[6]  =2*(e0*e2+e1*e3);
dcos[7]=2*(e2*e3-e0*e1);
dcos[8]=e0^2-e1^2-e2^2+e3^2;

** perform the matrix multiplication
g_body[0]=dcos[0]*g_x  +  dcos[1]*g_y  +  dcos[2]*g_z;
g_body[1]=dcos[3]*g_x  +  dcos[4]*g_y  +  dcos[5]*g_z;
g_body[2]=dcos[6]*g_x  +  dcos[7]*g_y  +  dcos[8]*g_z;

** pass the array

g_out  =  passArray(g_body);
}


** If the dialog data is inconsistent for simulation, abort.
on checkdata
```

Script of block omega g

```
{

}


** Initialize any simulation variables.
on initsim
{

** Assume body coordinates line up with inertial coords at
**  initialization.

e0 = 1.0;
e1 = 0.0;
e2 = 0.0;
e3 = 0.0;

** allocate the arrays to be passed.
makearray(omega,  3);
makearray(dcos,  9);
makearray(g_body,  3);
}


** User clicked the dialog HELP button.
on help
{
showHelp();
}
```

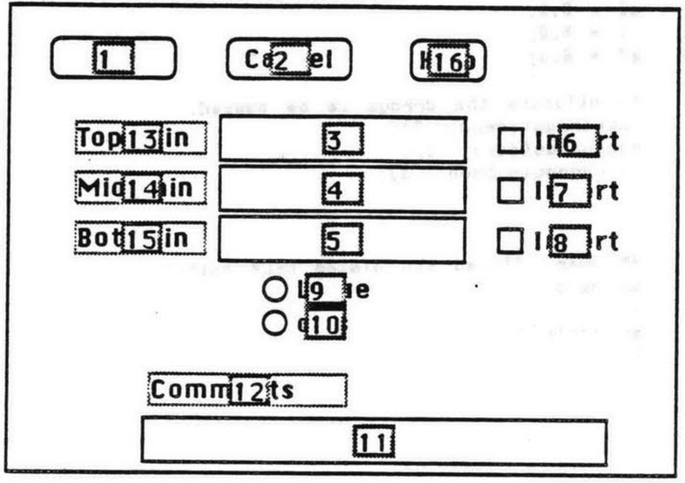
Icon of block vector amplifier



Connectors of block vector amplifier

Vector1In
Vector2In
Vector3In
VectorOut

Dialog of block vector amplifier



User messages of block vector amplifier

<1>	OK
<2>	Cancel
<3>	TopGain
<4>	MidGain
<5>	BotGain
<6>	TopInvert
<7>	MidInvert
<8>	BotInvert
<9>	Value
<10>	dbs
<11>	Comments
<12>	
<13>	
<14>	
<15>	
<16>	help

Script of block vector amplifier

```
** amplifier */
** totalSteps, curStep, totalTime, curTime, deltaTime */
** are defined by the system */

real         g1, g2, g3;
real         tempOut[];
real             temp1In[],temp2In[],temp3In[];

on dBs
{
** check for neg values */
if (not novalue(topGain) && topGain < 0.0)
    {
    topinvert = TRUE;            ** changes checkbox */
    topGain = -topGain; ** corrects gain */
    }
if (not novalue(midGain) && midGain < 0.0)
    {
    midinvert = TRUE;
    midGain = -midGain;
    }
if (not novalue(botGain) && botGain < 0.0)
    {
    botinvert = TRUE;
    botGain = -botGain;
    }
** convert vals to dBs */
topGain = 20.0*log10(topGain);
midGain = 20.0*log10(midGain);
botGain = 20.0*log10(botGain);
}

on value
{
** convert to value */
topGain = 10.0^(topGain/20.0);
midGain = 10.0^(midGain/20.0);
botGain = 10.0^(botGain/20.0);
}

on checkdata
{
if ((Vector1In && novalue(topgain)) ** if connected and no gain entered,
         || (Vector2In && novalue(midgain))   ** abort!!
         || (Vector3In && novalue(botgain)))
    abort;
}

on initsim
{
makearray(tempOut,3);
makearray(temp1In,3);
makearray(temp2In,3);
makearray(temp3In,3);

if (dBs)** if in dBs only */
    {
    ** convert sim vals to values for simulation */
```

Script of block vector amplifier

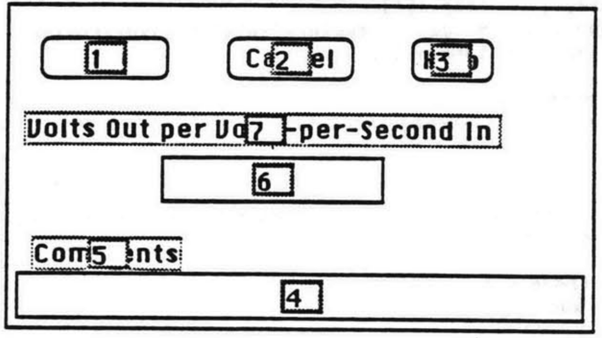
```
        g1  =  10.0^(topGain/20.0);
        g2  =  10.0^(midGain/20.0);
        g3  =  10.0^(botGain/20.0);
        }
   else
        {
        g1  =  topGain;
        g2  =  midGain;
        g3  =  botGain;
        }
   if  (topInvert)       **  is  it  inverted?  */
        g1  =  -g1;
   if  (midInvert)
        g2  =  -g2;
   if  (botInvert)
        g3  =  -g3;
   if  (novalue(g1))
        g1  =  0.0;
   if  (novalue(g2))
        g2  =  0.0;
   if  (novalue(g3))
        g3  =  0.0;
   }

   **  simulation  part  */
   on  simulate
        {
        GetPassedArray(Vector1In,temp1In);
        GetPassedArray(Vector2In,temp2In);
        GetPassedArray(Vector3In,temp3In);

        tempOut[0]  =  temp1In[0]*g1+temp2In[0]*g2+temp3In[0]*g3;
        tempOut[1]  =  temp1In[1]*g1+temp2In[1]*g2+temp3In[1]*g3;
        tempOut[2]  =  temp1In[2]*g1+temp2In[2]*g2+temp3In[2]*g3;

        VectorOut  =  passArray(tempOut);
        }


   on  help
   {
   showHelp();
   }
```

Icon of block vector differentiator

Connectors of block vector differentiator

NewIn
ConOut

Dialog of block vector differentiator



User messages of block vector differentiator

<1>	OK
<2>	Cancel
<3>	Help
<4>	Comments
<5>	
<6>	gain
<7>	

Script of block vector differentiator

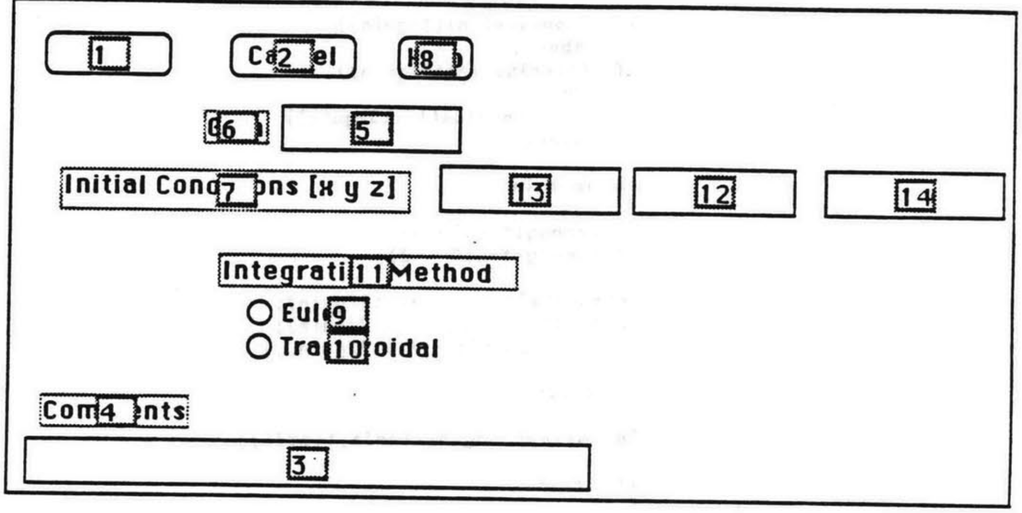
```
real  tempIn[];
real  tempOut[];
real  oldIn[3];

** This message occurs for each step in the simulation.
on simulate
{
** call getPassedArray
GetPassedArray(NewIn,tempIn);

** perform the calculations three times
tempOut[0]  =  gain*(tempIn[0]-oldIn[0])/deltaTime;
oldIn[0]  =  tempIn[0];
tempOut[1]  =  gain*(tempIn[1]-oldIn[1])/deltaTime;
oldIn[1]  =  tempIn[1];
tempOut[2]  =  gain*(tempIn[2]-oldIn[2])/deltaTime;
oldIn[2]  =  tempIn[2];

** pass the array
ConOut  =  passArray(tempOut);
}


** If the dialog data is inconsistent for simulation, abort.
on checkdata
{
if  (noValue(gain))
    abort;
}


** Initialize any simulation variables.
on initsim
{
makearray(tempIn,3);
makearray(tempOut,3);
oldIn[0]  =  0.0;          ** initial values are zero
oldIn[1]  =  0.0;
oldIn[2]  =  0.0;
}

on createBlock
{
gain = 1.0; ** initial value
}

** User clicked the dialog HELP button.
on help
{
showHelp();
}
```

Icon of block vector integrator

Connectors of block vector integrator

VectorIn
VectorOut

Dialog of block vector integrator



User messages of block vector integrator

<1>	OK
<2>	Cancel
<3>	Comments
<4>	
<5>	Gain
<6>	
<7>	
<8>	help
<9>	euler
<10>	trap
<11>	
<12>	init2
<13>	init1
<14>	init3

Script of block vector integrator

```
reala[4],b[4],c[4];
real     tempIn[];
realtempOut[];

on  createmodule
{
init1  =  0.0;
init2  =  0.0;
init3  =  0.0;
Gain  =  1.0;
}

on  checkdata
{
if  (novalue(init1+gain))
    abort;
if  (novalue(init2+gain))
    abort;
    if  (novalue(init3+gain))
    abort;}

on  initsim
{
makearray(tempIn,3);
makearray(tempOut,3);

integrateInit(a,  init1/gain);
integrateInit(b,  init2/gain);
integrateInit(c,  init3/gain);}

on  simulate
{
GetPassedArray(VectorIn,tempIn);

if  (euler)
{
    tempOut[0]  =  gain*integrateEuler(a,  tempIn[0],  deltaTime);
    tempOut[1]  =  gain*integrateEuler(b,  tempIn[1],  deltaTime);
    tempOut[2]  =  gain*integrateEuler(c,  tempIn[2],  deltaTime);
}
else
{   tempOut[0]  =  gain*integrateTrap(a,  tempIn[0],  deltaTime);
    tempOut[1]  =  gain*integrateTrap(b,  tempIn[1],  deltaTime);
    tempOut[2]  =  gain*integrateTrap(c,  tempIn[2],  deltaTime);
}

VectorOut  =  passArray(tempOut);
}

on  help
{
showHelp();
}
```

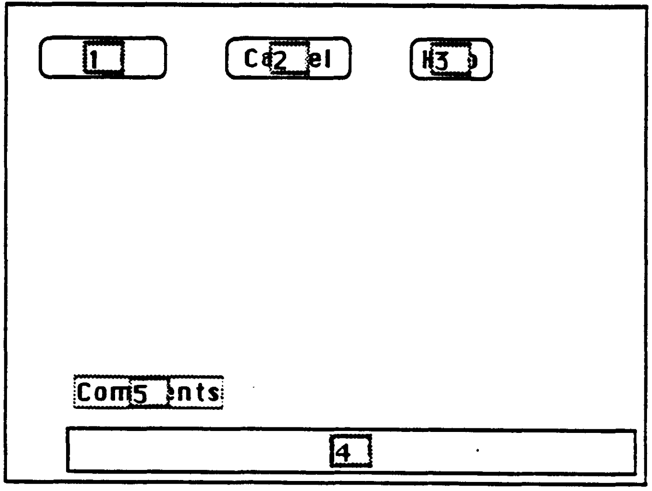
Icon of block vector to components

vector
components

Connectors of block vector to components

gx_out
CosineIn
gy_out
gz_out

Dialog of block vector to components



User messages of block vector to components

<1>	OK
<2>	Cancel
<3>	Help
<4>	Comments
<5>	

Script of block vector to components

```
real dummy [];
real  g_xout,g_yout,g_zout;

** This message occurs for each step in the simulation.
on simulate
{

** call get PassedArray with the connector and the array that it
** will be assigned to.

GetPassedArray(CosineIn,dummy);

** break out the three components
gx_out = dummy[0];
gy_out = dummy[1];
gz_out = dummy[2];

}


** If the dialog data is inconsistent for simulation, abort.
on checkdata
{

}


** Initialize any simulation variables.
on initsim

{

makearray(dummy,  3);

}


** User clicked the dialog HELP button.
on help
{
showHelp();
}
```

REFERENCES

Allum, J H, Tole J R, Weiss, A D. (1975): MITNYS-II: A digital program for on-line analysis of nystagmus. *IEEE Transactions on Biomedical Engineering* BME-22 No. 3.

Alpern, D. (1969): Kinematics of the eye. In: *The Eye* 3:13-25, Ed. Davson. Academic Press.

Alpern, D. (1969): Anatomy of eye movements. In: *The Eye* 3:27-64, Ed. Davson. Academic Press.

Anderson, J E. (1983): *Grant's Atlas of Anatomy*. Williams & Wilkins, Baltimore/London.

Anzaldi E and Mira E. (1975): An interactive program for the analysis of ENG tracing. *Acta. Oto*. 80:120-127.

Arbib, M A and Amari S. (1985): Sensori-motor transformations in the brain (with a critique of the tensor theory of cerebellum). *J. Theor. Biol.* 112:123-155.

Baland, J F, Godaux, E R, and Cheron, G A. (1987): Algorithms for the analysis of the nystagmic eye movements induced by sinusoidal head rotations. *IEEE Transactions on Biomedical Engineering* Vol. BME-34 No. 10 pp.811-816 October.

Baloh R W, Kumley, W E, and Honrubia, V. (1976): Algorithm for analysis of saccadic eye movements using a digital computer. *Aviat., Space, and Envir. Med.* May.

Baloh, R W, Langhofer, L, Honrubia, V, and Yee R D. (1980): On-line analysis of eye movements using a digital computer. *Aviat., Space, and Envir. Med.* June.

Baloh, R W, Richman, L, Yee, R D, and Honrubia, V. (1983): The dynamics of vertical eye movements in normal human subjects. *Aviation, Space, and Environmental Med.* January, pp.32-38.

Barnes, G R. (1982): A procedure for the analysis of nystagmus and other eye movements. *Aviation, Space, and Environmental Medicine* July .

Benson, A J. (1962): Effect of linear acceleration on labyrinthine nystagmus in man. Proc. 22nd Int. Cong. Physiol. Sci.

Benson, A J. (1966): Modification of the per- and post-rotational responses by the concomitant linear acceleration. In: 2nd Symposium of the Role of the Vestibular Organs in Space Exploration. NASA SP-115:199-213.

Benson, A J. (1974): Modification of the response to angular accelerations by linear accelerations. In: *Handbook of Sensory Physiology. Vol. VI Vestibular System. Part 2 Psychophysics, Applied Aspects and General Interpretations*. Kornhuber, ed. Springer-Verlag, pp. 281-320.

Benson, A J and Bodin, M A. (1966a): Interaction of linear and angular accelerations on vestibular receptors in man. *Aerospace Med.* 37:144-154 .

Benson, A J and Bodin, M A. (1966b): Effect of orientation to the gravitational vertical on nystagmus following rotation about a horizontal axis. *Acta Oto.* 61:517-526.

Bles, W. (1979): *Sensory Interactions and Human Posture.* Academische Pers. Amsterdam.

Bles, W, Kloren, T, Buchele, W and Brandt, T. (1983): Somatosensory Nystagmus: Physiological and Clinical Aspects. In: *Advances in Oto-Rhino-Laryngology.* Volume 30.

Bergstedt, M. (1961): Studies of positional nystagmus in the human centrifuge. *Acta Oto. Suppl.* 165.

Borah, J, Young, L R, and Curry, R E. (1988): Optimal estimator model for human spatial orientation. *Annals of New York Academy of Sciences* 545:51-73 .

Brandt, T, Buchele, W, Arnold, F. (1977): Arthrokinetic Nystagmus and Ego-motion Sensation. *Exp. Brain Res.* 30:331.

Brown, R. (1983): *Introduction to Random Signal Analysis and Kalman Filtering.* John Wiley and Sons, NY, NY .

Breuer, J. (1874): Ueber die Funktionder Bogengange des Ohrlabyrinths. *Med. Jahrbucher. Wien* 72-124.

Buizza, A, Leger, A, Berthoz, A, Schmid, R. (1979): Otolithic-acoustic interaction in the control of eye movement. *Exp. Brain Res.* 46:509.

Cheng, M, Gannon R P, and Outerbridge, J S. (1973): Frequency Content of Nystagmus. *Aero. Med.* April.

Cheng, M and Outerbridge J S. (1974): Intersaccadic interval analysis of optokinetic nystagmus. *Vis. Res.* 14:1053-1058.

Cohen, B, Matsuo, V, and Raphan, T. (1977): Quantitative analysis of the velocity characteristics of optokinetic nystagmus and optokinetic after nystagmus. *J. Physiol.* 270:321-344.

Correia, M J and Guedry, F E. (1966): Modification of vestibular responses as a function of rate of rotation about an earth-horizontal axis. Report No. 957. Pensacola, FL Nav. Aerospace Med. Inst..

Crampton, G H. (1966): Does linear acceleration modify cupular deflection? 2nd Symposium of the Role of the Vestibular System in Space Exploration NASA SP-115:169-184.

Crum Brown, A. (1874): On the sense of rotation and the anatomy and physiology of the semicircular canals of the internal ear. *J. Anat. Physiol.* 8:327-331.

Degler, H E, Smith, J R, and Black, F O. (1975): Automatic detection and resolution of synchronous rapid eye movements. *Comp. Biomed. Res.* 8:393-404.

Dichgans, J and Brandt Th. (1974): The psychophysics of visually induced perception of self-motion and tilt. In: The Neurosciences: Third Study Program. Schmitt and Worden (eds.) MIT Press, Cambridge, MA, pp 123-129.

vanEgmond, A A J, Groen, J J, and Jongkees, L B W. (1949): The mechanics of the semicircular canal. *J. Physiol., London* 110:1-17.

Fernandez, C and Goldberg J. (1976a): Physiology of Peripheral Neurons Innervating the Otolith Organs of the Squirrel Monkey: I. Response to Static Tilts and to Long-Durations Centrifugal Force. *J. of Neurophys.* 39:970-984.

Fernandez, C and Goldberg J. (1976b): Physiology of Peripheral Neurons Innervating the Otolith Organs of the Squirrel Monkey: II. Directional Selectivity and Force-Response Relations. *J. of Neurophys.* 39:985-995.

Fernandez, C and Goldberg J. (1976c): Physiology of Peripheral Neurons Innervating the Otolith Organs of the Squirrel Monkey: III. Response Dynamics. *J. of Neurophys.* 39:996-1008.

Flourens, P. (1842): *Recherches Experimentales sur les Proprietes et les Fonctions du Systeme Nerveux dans les Animaux Vertebres.* Paries: J-B Bailliere.

Gelb, A ed. (1974): *Applied Optimal Estimation.* MIT Press, Cambridge, MA.

Gentles W. (1974): *Application of automated techniques to the study of vestibular function in man.* Ph.D thesis, Department of Electrical Engineering, University of Toronto.

Gentles W and Barber H O. (1973): Computer and human variation in the measurement of post caloric nystagmus. *Equilibrium Res. Bul..*

Goldstein, Herbert. (1959): *Classical Mechanics.* Addison-Wesley.

Goldberg, J A and Fernandez C. (1971a): Physiology of peripheral neuronons innervating semicircular canals of the squirrel monkey. I. Resting discharge and response to constant angular accelerations. *J. of Neurophysiology*, 34:635-660.

Goldberg, J A and Fernandez C. (1971b): Physiology of peripheral neuronons innervating semicircular canals of the squirrel monkey. II. Response to sinusoidal stimulation and dynamics of peripheral vestibular sytem. *J. of Neurophysiology*, 34:661-675.

Goldberg, J A and Fernandez C. (1971c): Physiology of peripheral neurons innervating semicircular canals of the squirrel monkey. III. Variations among units in their discharge properties. *J. of Neurophysiology*, 34:676-684.

Goltz,F. (1870): Uber die physiologische Bedeutung der Bogengange des Ohrlabyrinthes. *Pfluegers Arch. Gesamte Physiol. Menschen Tiere* 3 pp 172-192.

Graybiel, A and Brown, R. (1951): The delay in visual reorientation following exposure to an change in direction of resultant force on a human centrifuge. *J. General Psychology*, 45:143-150.

Graybiel, A and Clark, B. (1965): The validity of the oculogravic illusion as a specific indicator of otolith function. *Aerospace Med.* 36:1173-1181.

Groen, J J, Lowenstein, O, and Vendrik, A J H. The mechanical analysis of the responses from the end-organs of the horizontal semicircular canal in the isolated elasmobranch labyrinth. *J. Physiol., London* 117:329-346.

Guedry, F E. (1965a): Orientation of the rotation-axis relative to gravity; its influence on nystagmus and the sensation of rotation. *Acta Oto.* 60:30-49.

Guedry, F E. (1965b): Psychophysiological studies of vestibular function. In: *Contributions to Sensory Physiology.* Academic Press.

Guedry, F E and Benson, A J. (1970): Tracking performance during sinusoidal stimulation of the vertical and horizontal semicircular canals. *Recent Advances in Aerospace Medicine*, pp. 276-288. D. Reidel Publishing Co., Dordrecht, Netherlands.

Guedry, F E , Collins, W E, and Sheffey, L. (1961): Perceptual and Oculomotor Reactions to Interacting Visual and Vestibular Stimulation. *Percept. Motor Skill* 12:307-324.

Guedry, F E and Harris, C S. Labyrinthine function related to experiments on the parallel swing. NSAM-874. Naval School of Aviation Medicine, Pensacola, FL.

Guedry, F E and Turnipseed, G T. (1968): Two devices for analysis of nystagmus. *Ann. Otol. Rhinol. Laryngol.* 77:1071.

Haddad, F B, Paige, G D, Doslak, M J, and Tomko, D L. (1988): Practical Method, and Errors, in 3-D Eye Movement Measurements. *Invest. Ophthalmol. & Vis. Sci. (ARVO suppl.)* 29:347.

Hain, T C. (1986): A model of the nystagmus induced by off vertical axis rotation. *Biol. Cybern.* 54:337-350.

Hallpike, C S and Hood, J D. (1953): Fatigue and adaptation of the cupular mechanism of the human horizontal semicircular canal: an experimental investigation. *Proc. Roy. Soc., London Ser. B* 141:542-561.

Hansson G A, Henriksson N G, Pyykko, I, Schalen, L, Wennmo, C, Magnussen M. Computer analysis of nystagmus. *Computers in Otoneurology.* Ed. by G-F Claussen.

Harris, L R. (1987): Vestibular and optokinetic eye movements evoked in the cat by rotation about a tilted axis. *Exp. Brain Res.* 66:522-532.

Hein, A and Held, R. (1961): A neural model for labile sensorimotor coordinations. *Biological Prototypes and Synthetic Systems* 1:71-74.

Held, R. (1961): Exposure history as a factor in maintaining stability of perception and coordination. *J. Nervous and Mental Disease* 132:26-32.

Held, R, Dichgans, J, Bauer, J. (1975): Characteristics of oving visual scenes influencing spatial orientation. *Vision Res.* 15:357-365.

Henn, V, Cohen, B, and Young, L R. (1980): Visual-vestibular interaction in motion perception and the generation of nystagmus. *Neurosciences Research Program Bulletin* 18.

Henriksson, N G. (1955): An electrical method for registration and analysis of the movements of the eyes in nystagmus. *Acta Oto.* 45:25.

Herberts, G, Abrahamsson, S, Einarsson, S, Hofmann, H, and Linder, P. (1968): Computer analysis of electronystagmographic data. *Acta oto.* 65:200-208.

Hixson, W.C., Niven, J.I., and Corrreia, M.J. (1966): Kinematics nomenclature for physiological accelerations: with special reference to vestibular applications. Monograph 14. Naval Aerospace Medical Institute, Pensacola, FL.

von Holst, E. (1954): Relations between the central nervous system and the peripheral organs. *Brit. J. Animal Behavior* 2:89-94.

von Holst, E. (1957): Aktive Leistungen der menschlichen Gesichtswahrnehmung. *Studium Generale* 10:231-243.

von Holst, E. and Mittelstaedt, H. (1950): Das Reafferenzprinzip. *Naturwissenschaften* 37: 464-476.

Honrubia, V, Katz, R D, Strelioff, D, Ward P H. (1971): Computer analysis of induced vestibular nystagmus. *Annals of Otology, Rhinology & Laryngology Sup.* 3 Vol LXXX, Dec..

Hughes, Peter C. (1989): *Spacecraft Attitude Dynamics*; Wiley.

Inchingolo, P and Spanio, M. (1985): On the identification and analysis of saccadic eye movements-A quantitative study of the processing procedures. *IEEE Transactions on Biomedical Engineering* Vol. BME-32 No. 9 Sept..

Jongkees, L B W. (1960): On positional nystagmus. *Acta Oto. Suppl.* 159:78-83.

Juhola, M. (1988): Detection of nystagmus eye movements using a recursive digital filter. *IEEE Transactions on Biomedical Engineering* Vol. 35 No. 5 pp.389-397 May.

Kalman, R E. (1960): Contributions to the Theory of Optimal Control. *Biol. Soc. Mat. Mexicana* 5:102-119.

Kalman, R E, and Bucy, R S. (1961): New Results in Linear Filtering and Prediction Theory. *J. Basic Eng. Trans. ASME* Ser D 83:95-108.

Khilov, K L. (1929): *Zh. ushn. nos. gorlov Bolezn.* 6:289-299.

Khilov, K L. (1936): in Sb. trudov tsentral'noy nauchnoy psikhofiziologicheskoy laboratorii po izucheniyu letnogo truda grazhdanskogo vozdushnogo flota SSSR. (Collection of papers by the Central Psychophysiologica Laboratory on the study of the activities of the civil air force of the USSR) 1:5-72.

Kitchen, B J. (1983): Horizontal and vertical eye deviations in response to linear accelerations. B.S. thesis, MIT.

Kleinman, D L, Baron, S and Levinson, W H. (1970): An optimal control model of human response, Par I. Theory and validation. *Automatica* 6:357-369.

Krieger, H P and Bender, M B. (1956): Optokinetic after-nystagmus in the monkey. *Electroencephalogr. Clin. Neurophysiol.* 8:96-106.

Lackner, J R. (1977): Induction of illusory self-rotation and nystagmus by a rotating sound field. *Aviation, Space, and Environmental Medicine* 48:129.

Lansberg, M P, Guedry, F E, and Graybiel, A. (1965): Effect of changing resultant linear acceleration relative to the subject on nystagmus generated by angular acceleration. *Aerospace Medicine* Vol. 36 No. 5 pp 456-460.

Levenberg, A. (1944): Method for the Solution of Certain Non-linear Problems in Least-squares. *Quart. Appl. Math.* 2:164-168.

Lowenstein, O, and Sand, A. (1940): The mechanism of the semicircular canal. A study of responses of single fibre preparations to angular acceleration and to rotation at constant speed. *Proc. Roy. Soc., London, Ser B* 129:256-275.

Ljung, L. (1987): *System Identification: Theory for the User.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Mach, E. (1873): Physikalische Versuche uber den Gleichgewichtssinn des Menschen 1. Mitteilung. *Akad. Wiss. Wien* 68:124-140.

Markaryan, S S. (1969): The problem of the interrelationtionship between the function of the semicircular canals and that of the otolith apparatus. In: *Problemy kosmicheskoy biologii,* 4th N. M. Sisakyan, E. Moscow: Nauka Press 176-181, 1967. English Translation, Problems of Space Biology, NASA TT-528, 185-190.

Marquardt, D. (1963): An Algorithm for Least-Squares Estimation of Nonlinear Parameters. J. *Soc. Ind. Appl. Math.* 11:431-441.

Massoumnia, M A. (1983): Detection of fast phases of nystagmus using digital filtering. M.S. Thesis, Department of Aeronautics and Astronautics, MIT May.

Matsuo, V and Cohen, B. (1984): Vertical optokinetic nystagmus and vestibular nystagmus in the monkey: up-down asymmetry and effects of gravity. *Exp. Brain Res* . 53: 197-216.

Matsuo, V, Cohen, B, Raphan, T, deJong, V, and Henn, B. (1979): Asymmetric velocity storage for upward and downward nystagmus. *Brain Res.* 176: 159-164.

McCabe, B F. (1964): Nystagmus response of the otolith organs. *Laryngoscope* 74:372-381.

McElligott, M H, Loughnane, M H, and Mays, L E. (1979): The Use of Synchronous Demodulation for the Measurment of Eye Movements by Means of an Ocular Magnetic Search Coil. *IEEE Trans. Biomed. Engr.* BME-26:370-374 June.

Melvill Jones, G, Barry, W, and Kowalsky, N. (1964): Dynamics of the semicircular canals compared in yaw, pitch, and roll. *Aerospace Med.* October.

Michaels, D L. (1977): Microprocessor analysis of eye movements. S.M. Thesis, Department of Electrical Engineering, MIT January.

Miller, E F, Fregley, A R, and Graybiel, A. (1966): Visual horizontal perception in relation to otolith function. NAMI-989 Naval Aerospace Medical Institute, Pensacola, FL.

NASA Ames Research Bulletin. Vestibular Research Facility (VRF). Brochure describing the purpose of the VRF and the equipment capabilities. (Undated, approximately 1988).

Ni, M D. (1980): A minicomputer program for automatic saccade detection and linear analysis of eye movement systems using sine wave stimulus. *Computer Programs in Biomedicine* 12:27-41.

Niven, J. I., Hixson, W. C., and Correia; Elicitation of Horizontal Nystagmus by Periodic Linear Acceleration. *Acta Oto.* 62:429-441.

Oman, C M. (1982): A heuristic mathematical model for the dynamics of sensory conflict and motion sickness. *Acta Oto. Suppl.* 392.

Oman, C M. (1988): Sensory conflict in motion sickness: an observer theory approach". Proceedings of the Symposium and Workshop on Spatial Displays and Spatial Instruments. Asilomar, California, September.

Oppenheim, A. V. and Schafer, R. W. (1975): *Digital Signal Processing*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Ormsby, C C, and Young, L R. (1977): Integration of semicircular canal and otolith information for multisensory orientation stimuli. *Mathematical Biosciences* 34:1-21.

Ostriker, G, Pellionisz, A, and Llinas, R. (1985): Tensorial computer model of gaze-Ocuomotor activity is expressed in non-orthogonal natural coordinates. *Neuroscience* 14:483-500.

Paige, G D. (1983): Vestibuloocular reflex and its interactions with visual following mechanisms in the squirrel monkey. I. Response Characteristics in Normal Animals. *J. of Neurophys.* 49:134-151 January.

Paige, G D. (1983): Vestibuloocular reflex and its interactions with visual following mechanisms in the squirrel monkey. II. Response Characteristics and Plasticity Following Unilateral Inactivation of Horizontal Canal. *J. of Neurophys.* 49:152-168 January.

Paige, G D and Tomko D L. (1989): Eye movements and visual-vestibular interactions during linear head motion. *Proceedings of the Centre National de la Recherche Scientific Symposium on Head Movement Control.* A. Berthoz, (ed.).

Pandovan, I and Pansini M. (1972): New possibilities of analysis in electronystagmography. *Acta Oto.* 73:121-125.

Pellionisz, A and Llinas, R. (1980): Tensorial approach to the geometry of brain function: Cerebellar cordination via a metric tensor. *Neuroscience* 5:1125-1136.

Press, W, Flannery, B, Teukolsky, S, and Vetterling, W. (1986): *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press.

Raphan, T and Cohen, B. (1983): Gravitational effects on visual vestibulo-oculomotor coordinate transformations generating compensatory eye movements. *Soc. Neurosci. Abstr.* 13:316.

Raphan, T and Cohen, B. (1988): Organizational principles of velocity storage in three dimensions; the effect of gravity on cross-coupling of optokinetic after-nystagmus (OKAN). *Ann. NY Acad. Sci.*.

Raphan, T, Cohen B, and Henn, V. (1981): Effects of gravity on rotatory nystagmus in monkeys. *Ann NY Acad. Sci.* 374:44-55.

Raphan, T, Matsuo, V, and Cohen, B. (1977): A velocity storage mechanism responsible for optokinetic nystagmus (OKN), optokinetic after-nystagmus (OKAN) and vestibular nystagmus. in: *Control of Gaze by Brain Stem Neurons.* Elsevier/North Holland Biomedical Press pp 37-47.

Raphan, T, Matsuo, V, and Cohen, B. (1979): Velocity stroage in the vestibulo-ocular reflex arc (VOR). *Exp. Brain Res.* 35:229-248.

Raphan, T and Cohen, B. (1986): Multidimensional organization of the vestibulo-ocular reflex (VOR). In: Keller, and Zee (Eds.) *Adaptive Processes in Visual and Oculomotor Systems.* pp. 285-292.

Reason, J T. (1969): Motion sickness-some theoretical considerations. *Int J. Man-Machine Studies* 1:21-38.

Reason, J T. Learning to cope with atypical force environments. *Adult Learning*, M. Howe, Ed., London, Wiley pp 203-22 (1977).

Reason, J T. Motion sickness adaptation: a neural mismatch model. *J. Royal Soc. Med.* 71:819-829 (1978).

Robinson, D A. (1977): Vestibular and optokinetic symbiosis: an example of explaining by modelling. In: *Control of Gaze by Brain Stem Neurons, Developments in Neuroscience, Vol. 1.* Elsevier/North-Holland Biomedical Press pp. 49-58.

Robinson, D A. (1982): The use of matrices in analyzing the three-dimensional behavior of the vestibulo-ocular reflex. *Biol. Cybern.* 46:53-66.

Robinson, D A. (1985): The coordinates of neurons in the vestibulo-ocular reflex. In *Adaptive mechanisms in gaze control,* Eds. Bethoz & Melvill Jones. Elsevier Science Publishers BV.

Sills, A W, Honrubia, V, and Kumley, W E. (1975): Algorithm for the multi-parameter analysis of nystagmus using a digital computer. *Aviat. Space and Envir. Med.* July .

Smith, J R, Cronin, M J, and Karacan, I. (1971): A multichannel hybrid system for rapid eye movement detection (REM detection). *Computers and Biomedical Research* 4:275-290.

Stahle, J. (1958): Electro-nystagmography in the caloric and rotatory tests. *Acta Oto. Suppl.* 137.

Steinhausen, W. (1933): Uber die Beobachtung der Cupula in den Bogengansampullen des Labyrinths des lebenden Hechts. *Arch. Ges. Physiol.* 232:500-512.

Steer, R W. (1967): The Influence of Angular and Linear Acceleration and Thermal Stimulation on the Human Semicircular Canal. MIT Sc.D. Thesis.

Sturm, D and Raphan, T. (1988): Modelling the three dimensional structure of velocity storage in the vestibulo-ocular reflex (VOR). Proceeding of the 14th Bioengineering Conference (IEEE).

Ter Braak, J W. (1937): Untersuchungen ueber optokinetischen Nystagmus. *Arch. Neerl. Physiol.* 20:309-376.

Tokita, T, Suzuki, T, Hibi, T, and Tomita, T. (1975): A quantitative test of optokinetic nystagmus and its data processing by computer. *Acta. Oto. Suppl.* 330:159-168.

Tole, J R and Young L R. (1971): MITNYS: A hybrid program for on-line analysis of nystagmus. *Aero. Med.* 42:508-511.

Tomko, D L, Wall, C, Robinson, FR, and Staab, JP. (1986): Gain and phase of cat verticl eye movements generated by sinusoidal pitch rotations with and without head tilt. *Aviat. and Space Environmental Med.*

Tweed, D and Vilis, T. (1987): Implications of rotational kinematics for the oculomotor system in three dimensions. *J. Neurophysiology* 58:832-849.

Voots, R J. (1969): Computerless automatic data reduction for electronystagmography. *Aerospace Medicine.* 40:1080.

Wall, C. (1987): Eye movements induced by gravitational force and by angular acceleration: their relationship. *Acta. Oto.* 104:1-6.

Wall, C and Black FO. (1981): Algorithms for the clinical analysis of nystagmus eye movements. *IEEE Transactions on Biomedical Engineering* Vol BME-28 No. 9 638-646 Sept..

Westheimer, G. (1957): Kinematics of the eye. *J. of the Optical Society of America* Vol. 47 Number 10 pp 967-974.

Young, L R. (1967): Effects of linear acceleration on vestibular nystagmus. 3rd Symposium of the Role of the Vestibular Organs in Space Exploration NASA SP-152.

Young, L R. (1970): On Visual-Vestibular Interaction. Proc. 5th Symposium of the Role of the Vestibular Organs in Space Exploration. NASA SP 314:205-210.

Young, L R. (1978): Man's internal navigation system. *Technology Review* 8:40-45.

Young, L R. (1981): Visual and vestibular influences in human self-motion perception. In: Gualtierotti, T (ed.) *The vestibular system: function and morphology.* Springer, Berlin Heidelberg, New York.

Young, L R. (1983): Perception of the body in space: mechanisms. In: Darian-Smith I (ed.) *Handbook of physiology: The nervous system.* American Physiology Society, Bethesda MD pp 1023-1066.

Young, L R and Oman, C M. (1970): Modelling adaptation in human semicircular canal response to rotation. *New York Academy of Sciences* 32:489-494.

Young, L R and Sheena, D. (1975): Survey of eye movement recording methods. *Behavior Research Methods & Instrumentation*, Vol 7 (5) 397-429.