INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9108180

A nonlinear identification method for modelling the three-dimensional structure of velocity storage in the vestibulo-ocular reflex (VOR)

Sturm, Deborah Dorit, Ph.D.

City University of New York, 1990

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

A

A Nonlinear Identification Method for Modelling the Three Dimensional Structure of Velocity Storage in the Vestibulo-ocular reflex (VOR)

by

Deborah Sturm

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

1990

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

June 18, 1990
date

Theodore Raphan
Chairman of Examining Committee

June 18, 1990
date

Executive Officer

Professor Michael Anshel

Professor Bernard Cohen

Professor Robert Fanelli

Professor Theodore Raphan (Chairman)

Supervisory Committee

The City University of New York

Abstract

A Nonlinear Identification Method for Modelling the Three Dimensional Structure of Velocity Storage in the Vestibulo-Ocular Reflex (VOR)

by

Deborah Sturm

Advisor: Professor Theodore Raphan

Velocity storage is present in vertical and roll components of compensatory eye movements if the head is oriented appropriately with regard to gravity (Raphan & Cohen, 1983; Matsuo & Cohen, 1984; Raphan & Cohen, 1987). To explain this, it has been postulated that gravity orients the eigenvectors of the three dimensional representation of velocity storage towards the space vertical (Raphan & Cohen, 1987; Sturm & Raphan, 1988). The thesis devised a computational procedure to identify the parameters of the system matrix from OKAN data. This procedure should help test this hypothesis in behavioral paradigms.

Velocity storage which is expressed by OKAN (Raphan et al, 1979), has been represented as a dynamical system, $x' = Hx$, where x is a three dimensional vector representing the state of

the system and H is the matrix containing the parameters that govern its dynamical behavior. Because of the assumption of linearity, the dynamical system was solved to give the pitch, roll, and yaw eye velocities in terms of the eigenvalues and eigenvectors of the system matrix. These parameters appear nonlinearly in the solution as a function of time. A nonlinear least squares fitting procedure (Marquardt, 1963) was adapted which compares data from cross- coupling experiments to the model outputs in order to identify the eigenvalues and eigenvectors of the system matrix. The eigenvalues and eigenvectors were determined using OKAN data from a monkey for a few tilt angles. When the parameters were set appropriately in the model, it closely predicted the cross-coupling effects.

Thus, a computational procedure has been developed which uses nonlinear identification techniques to identify parameters of a model of the three dimensional representation of velocity storage. This should be a useful tool in further theoretical and experimental studies on the vestibular ocular reflex.

Acknowledgements

I wish to thank my advisor, Professor Theodore Raphan, without whose input, I could never have completed this dissertation. He spent countless hours reading it and revising its organization and form. In addition, his support from various grants allowed me to concentrate fully on the work involved without having to teach.

I would also like to thank the members of my doctoral committe, Professors Michael Anshel, Bernard Cohen, and Robert Fanelli for their time and effort in reading and making useful suggestions.

I am very grateful for the support I received from the departments of Computer Science at Brooklyn College and at The College of Staten Island. In particular I would like to thank Professors Charles Schnabolk, Chaya Gurwitz, Aaron Tenenbaum, Keith Harrow, Kenneth McAloon, Roberta Klibaner, Marsha Moroh, and Charles Giardana.

Finally, I wish to acknowledge the support given to me by the National Institute of Health under grant no. EY04148, and the PSC-CUNY reseach award program no. 668 285, and NASA under grant no. NASA-17720 to Theodore Raphan.

Table of Contents

List of Illustrations

Figure

1 Introduction

1.1 Rationale and Motivation

The purpose of the vestibulo-ocular reflex (VOR) is to stabilize gaze in space. Understanding compensatory eye movements in response to rotational stimuli has proven useful in diagnosing various diseases and effects of drugs. More recently, interest has focused on space travel and how gravity affects eye movements.

Modelling the vestibulo-ocular reflex has provided insight into the central structure of this system and has identified the neural network components important in generating the compensatory eye movements. These models have provided a theoretical basis for devising experiments and interpreting the neurophysiological data. Currently, experimental studies are being planned to study eye movements in space in the absence of gravity. Modelling the system to incorporate the effects of gravity will allow us to compare how the parameters governing eye movements will be modified in the absence of gravity.

Eye movements in one dimension have been explored in depth. Models and mathematical procedures that identify their parameters are available which simulate visual-vestibular interaction over a wide range of stimulus conditions. It is known, however, that roll, pitch, or yaw eye movements involve neural activity in all three planes. Furthermore, when subjects are oriented in a position not aligned with gravity, their eye movements are not isomorphic to the upright case. Modelling the

three dimensional behavior of the vestibulo-ocular reflex has helped to elucidate the underlying mathematical structure of the central processing in the VOR which accounts for these differences. However, procedures for identifying the parameters of the three dimensional model whose responses can be compared to physiological data have not been developed. In this thesis we have developed an algorithm based on a nonlinear least squares parameter estimation procedure (Marquardt, 1963) to identify the parameters of the system matrix. The parameters were then related to the eigenvalues and eigenvectors of the matrix.

1.2 Organization of Dissertation

This dissertation is organized in a top-down manner starting with a background of the one dimensional models of the visual- vestibular interaction and then the exension to a three dimensional model in particular. Finally we show how the parameters of the model are obtained and how the model compares to experimental data.

Various models have been developed for the VOR both in one and three dimensions. They are summarized in chapter 2. In particular the model that the three dimensional model is based upon is described in detail in section 2.2. Then three dimensional models of visual-vestibular interaction are reviewed in section 2.3. It has been shown that the central processing of the VOR is computed in canal coordinates. The physiological basis for the assumption of a three dimensional structure of the semicircular canals and otolith organs is described in section 2.3.2.

The behavioral basis for the dynamic transformations of the system matrix is outlined in section 2.3.3. The relationship between human perception of the spatial vertical and the orientation of the eigenvectors of the system matrix associated with velocity storage is explored in this dissertation. A review of the human visual orientation with respect to gravity is presented in section 2.3.4.

The element of our model which we have examined in depth is the system matrix, H, associated with the velocity storage integrator. Section 2.3.5 describes the structure of H

as a function of gravity.

The identification of the parameters of the system matrix associated with velocity storage is the main focus of this dissertation. How the system matrix is formed from its eigenvectors and eigenvalues is explained in section 3.1. The dynamics of the transformed system and the closed form solutions are given in section 3.2.

We show how its gravity dependent parameters are extracted from data using a nonlinear least squares estimator in section 3.3. The theoretical basis for the Marquardt algorithm is described in section 3.3.1 and 3.3.2. The extension to vector valued functions in order to account for the three components of the velocity vector is discussed in section 3.3.3.

The process developed in section 3.3 is then applied to the paradigm of tilt experiments in section 3.4. The criterion function being minimized by the Marquardt algorithm has many local minima. Therefore the process of finding the best set of parameters greatly depends on the selection of the initial conditions of the parameters. This process is described in section 3.4.1. The computer implementation of the extended Marquardt method and flowcharts describing the program are given in section 3.5. The application of this general method to tilt experiments is explored in section 3.6.

The parameterization procedure developed in section 3.4 is data driven. The eigenvectors and eigenvalues are extracted from the data and then associated with the system matrix. Section 3.7 develops a procedure for choosing the initial parameters for arbitrary orientations for which there may not

be data. To do so we present a procedure for determining the vertical eigenvector and the eigenvalues for any orientation. A discussion and recommendations for future research are given in section 3.9.

2 Models of the VOR and Visual-Vestibular Interaction

2.1 Introduction

Modelling the VOR has led to a greater understanding of visual- vestibular processes. Relating models to clinical studies has allowed the interpretation of the effects of lesions and drugs (Raphan et al, 1979; Waespe et al, 1983; Cohen et al, 1987). In this dissertation we are examining the effects of gravity on OKN and OKAN. This involves studying the three dimensional aspects of the system matrix associated with the velocity storage integrator (Raphan et al, 1979; Raphan & Cohen, 1985). This important component of the central processing is common both to the visual and vestibular subsystems. In order to understand the behavior of the three dimensional integrator, we must first examine the three dimensional structure of the peripheral subsystem and how the vestibular and visual systems couple to it.

The hypothesis suggested by Raphan (Raphan and Cohen, 1988) that the gravitational field imposes a spatial reference onto the body via the velocity storage integrator which is reflected in its characterictic eigenvectors is essential for understanding the nature of compensatory eye movements with regard to gravity. This theory is supported by its similarity to how humans perceive the vertical in tilted orientations (Aubert, 1861; Muller, 1916; Schone, 1984; Mittlestaet, 1986).

The aim of this chapter is to review what is known about the organization of the vestibulo-ocular reflex and visual vestibular interaction and the models which have been developed to explain its behavior. We will show why the three dimensional structure of the velocity storage integrator, and how it is affected by gravity, is important for understanding the behavior of the system in three dimensions.

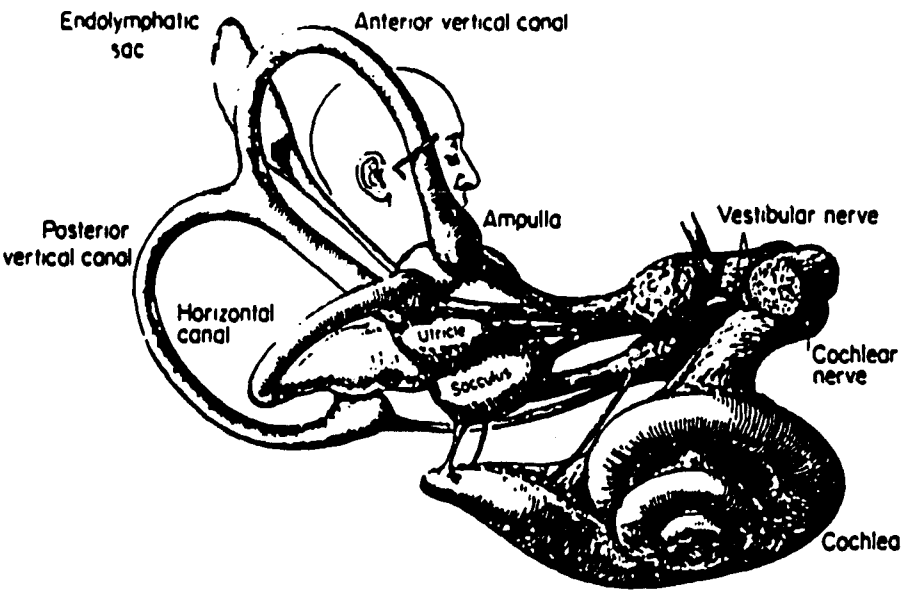
In the next section we consider a one dimensional model of the vestibulo- ocular reflex which focuses on the importance of the velocity storage integrator in mediating visual vestibular interactions.

2.2 One dimensional Models of Vestibular Nystagmus, OKN and Visual-Vestibular Interaction

One of the first models attempting to explain the response of the vestibular system to head rotations was formulated by Steinhausen (1933). He described how the semicircular canals behave as accelerometers in responding to angular movements of the head. Each canal is filled with a fluid called endolymph which when stimulated by an angular acceleration having a component normal to the plane of the ring, deviates a membrane (the cupula) in a direction opposite to head acceleration (Figure 2.1b). This in turn stimulates hair cells which cause the firing rates of the nerve to increase or decrease depending on the direction of rotation (Lowenstein & Sand, 1940; Hudspeth, 1977).

Steinhausen's equations described the fluid dynamics as a second order system and the parameters associated with the fluid were related to eye velocity during head rotations over a band of frequencies from .05 to 1Hz (Steinhausen, 1933). Later work showed that the dominant time constant associated with nerve activity was appoximately 3-5 seconds (Goldberg & Fernandez, 1971) while the ocular response had a time constant of 10-15 seconds (Skavenski and Robinson, 1973; Robinson, 1975). The application of the Steinhausen model to explain the eye velocity response to vestibular stimuli was therefore inadequate. The differences are seen in Figure 2.2a. The eye velocity response which would be generated by the cupula is shown by the solid line drawn on the velocity response. Thus, it was necessary to postulate the existence of a central

(A)



(B)

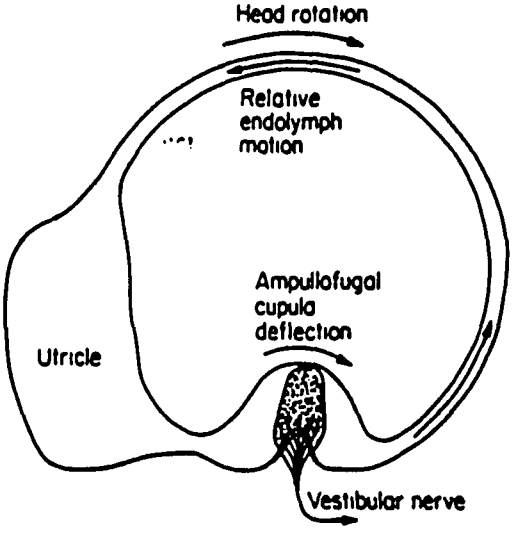


Figure 2.1 The Inner Ear

(A) The semicircular canals, utricle, and saccule together with the vestibular nerve innervating them. The placement of the horizontal canal tipped 15 degrees up in the head is shown.

(B) A vestibular canal. The arrows indicate the head rotation direction (clockwise in the plane of the canal) and the resulting fluid motion in the opposite direction.

Taken from Howard (1982) pp. 342,343.

mechanism that would enhance the dominant time constant of the VOR (Robinson, 1977; Raphan et al, 1977,1979).

A similar kind of "central storage mechanism" was necessary to explain the optokinetic response (Collewijn, 1972; Cohen et al, 1977). When a full field rotating optokinetic stimulus is viewed by a subject, there is a compensatory response of the eyes whose velocity approximates that of the velocity of the stimulus (Mowrer,1937; Ter Braak, 1936). When the lights are extinguished the eyes continue to beat in the same direction as the stimulus, with their velocity gradually decaying to zero. This response is called Optokinetic After-nystagmus (OKAN) (Figure 2.2b).

In the monkey, OKN is characterized by a rapid jump followed by a slow rise to a steady state velocity (Cohen et al, 1977; Raphan et al, 1979). The rapid rise is due to visual pathways to the oculomotor system called the direct pathways (Waespe et al, 1983). The pathways responsible for the slow changes during OKN and OKAN are the indirect pathways. The indirect pathway includes the velocity storage integrator which is common to both the visual and vestibular systems (Raphan et al, 1979; Waespe et al, 1983). During OKN the velocity storage integrator is charged, while OKAN represents the discharge of the integrator.

Human OKN and OKAN have similar characterics to that found in the monkey (Cohen et al, 1981; Jell et al, 1984). However, the details of the response are different. In humans, OKN is characterized by a rapid jump which is maintained for the duration of the stimulation. When the lights are extin-

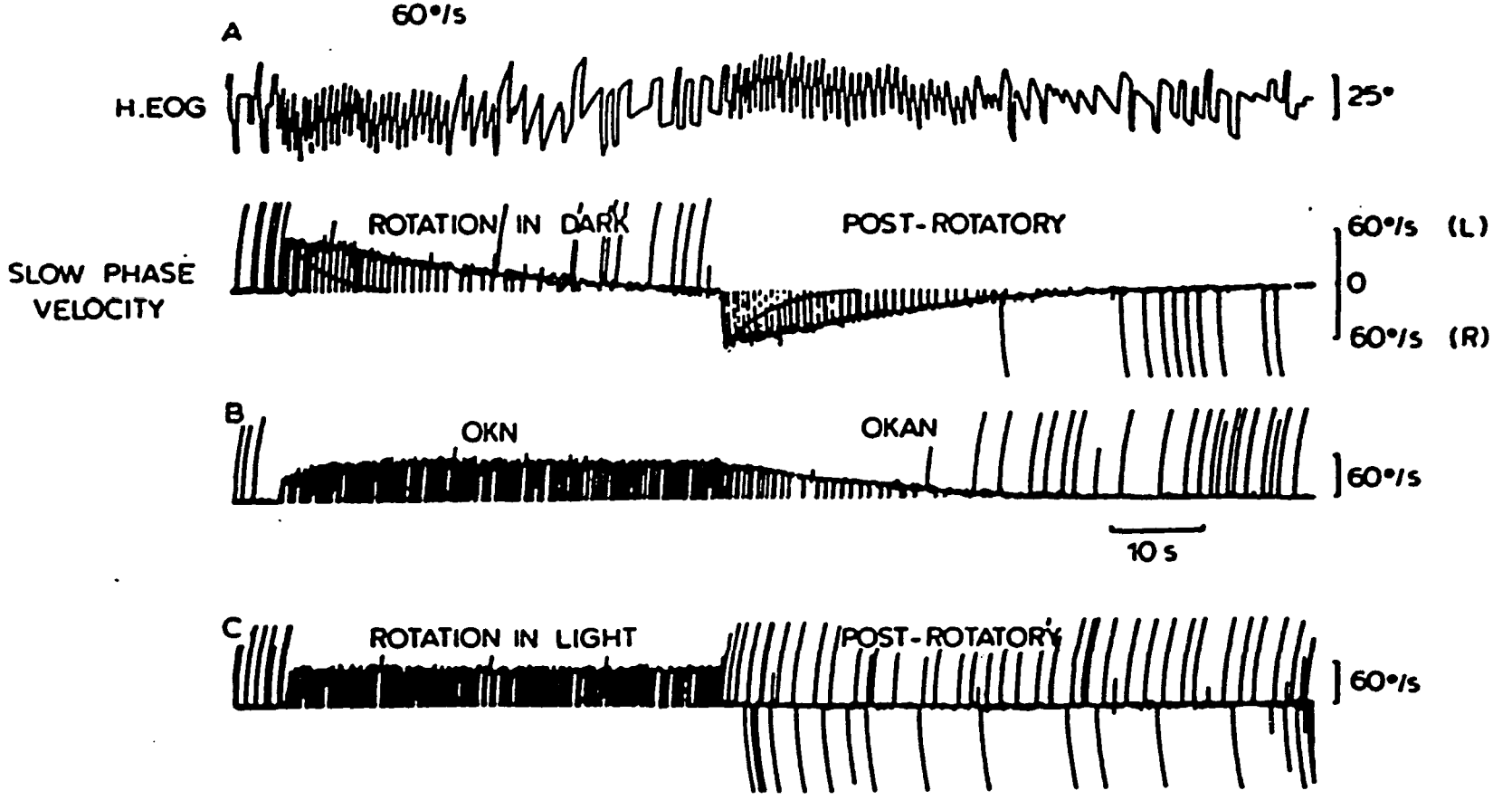
guished eye velocity drops and OKAN ensues (Cohen et al, 1981; Jell et al, 1984). The human OKAN response was modelled using two integrators resulting in a two component decay (Jell et al, 1984). One time constant was very short corresponding to the rapid drop in eye velocity at the start of OKAN (Jell et al, 1984). The other time constant is long and corresponds to the discharge of the velocity storage integrator. At high velocities (greater than 30 deg/sec), they found that the effects of the direct pathway were masked and dominated by the indirect pathway.

The rapid drop in eye velocity at the start of OKAN in humans is equivalent to that in monkey (Cohen et al, 1977; Raphan et al, 1979; Cohen et al, 1981) and is probably due to the inactivation of the direct pathway in darkness. This is supported by the greater gain in the rapid jump of eye velocity at the start of OKN. The greater jump in eye velocity at the start of OKN and the greater drop in eye velocity at the start of OKAN in humans as compared to that in the monkey supports the idea that the direct pathway plays a more dominant role in the OKN of humans (Cohen et al, 1981) than in monkeys. The gain of the OKAN in humans is small compared to that in the monkey. Thus, while human velocity storage as represented by OKAN is weaker than in the monkey, it nevertheless has similar characteristics and is probably affected by gravity in a similar fashion.

Studies involving visual-vestibular interaction have shown that the vestibular and visual systems complement one another to maintain fixed retinal images (Wilson and Melvill-Jones, 1979; Henn et al, 1980) and that the velocity storage integrator

Figure 2.2. Nystagmus induced by a step of platform rotation with eye position followed by eye velocity (A), by a step of surround rotation (B), and by a step of platform rotation in light with a stop in darkness (C). The stimulus velocity for each was 60 degrees per second. When rotary nystagmus and OKN were in the same direction, their after-responses were oppositely directed (A and B). There was only slight post-rotatory response after rotation in light (C). The solid line in the slow phase velocity trace of (A) is the VIII nerve time constant. (taken from Raphan & Cohen, 1980)

Figure 2.2.


A
60°/s
H.EOG
25°
SLOW PHASE VELOCITY
ROTATION IN DARK
POST-ROTATORY
60°/s (L)
0
60°/s (R)
B
OKN
OKAN
60°/s
10 s
C
ROTATION IN LIGHT
POST-ROTATORY
60°/s

is a central focus for this interaction. This is shown in Figure 2.2C. For rotations in light, there is a rapid rise in slow phase velocity similar to Figure 2.2a. This is followed by a maintained steady state velocity for the duration of rotation similar to the steady state velocity in Figure 2.2b. When the rotation is stopped and the animal is in darkness, the post rotatory nystagmus slow phase velocity is zero (Figure 2.2C). This corresponds to a cancellation of what would have been the postrotatory response in darkness (Figure 2.2a). The velocity storage integrator has been shown to play an important role in this cancellation in monkeys (Raphan et al, 1979; Waespe et al, 1983) as well as in humans (Cohen et al, 1981; Jell et al, 1984).

A one dimensional model has been developed to explain vestibular nystagmus, OKN and visual-vestibular interaction. It will now be described in detail since it forms the basis for the extension to three dimensions. (Cohen et al, 1977; Raphan et al, 1979; Waespe et al, 1983).

The equations which govern the dynamics of the integrator during optokinetic and vestibular stimulation are given by (see Figure 2.3):

$$x' = -h_o x + g_o * r_v + n(e) \qquad (2.2.1)$$

$$e = r_o - r_h - y \qquad (2.2.2)$$

$$y = r_v + x + v_f \qquad (2.2.3)$$

Vestibular nystagmus is generated by a head velocity signal r_h, which through the cupula dynamics generates the signal

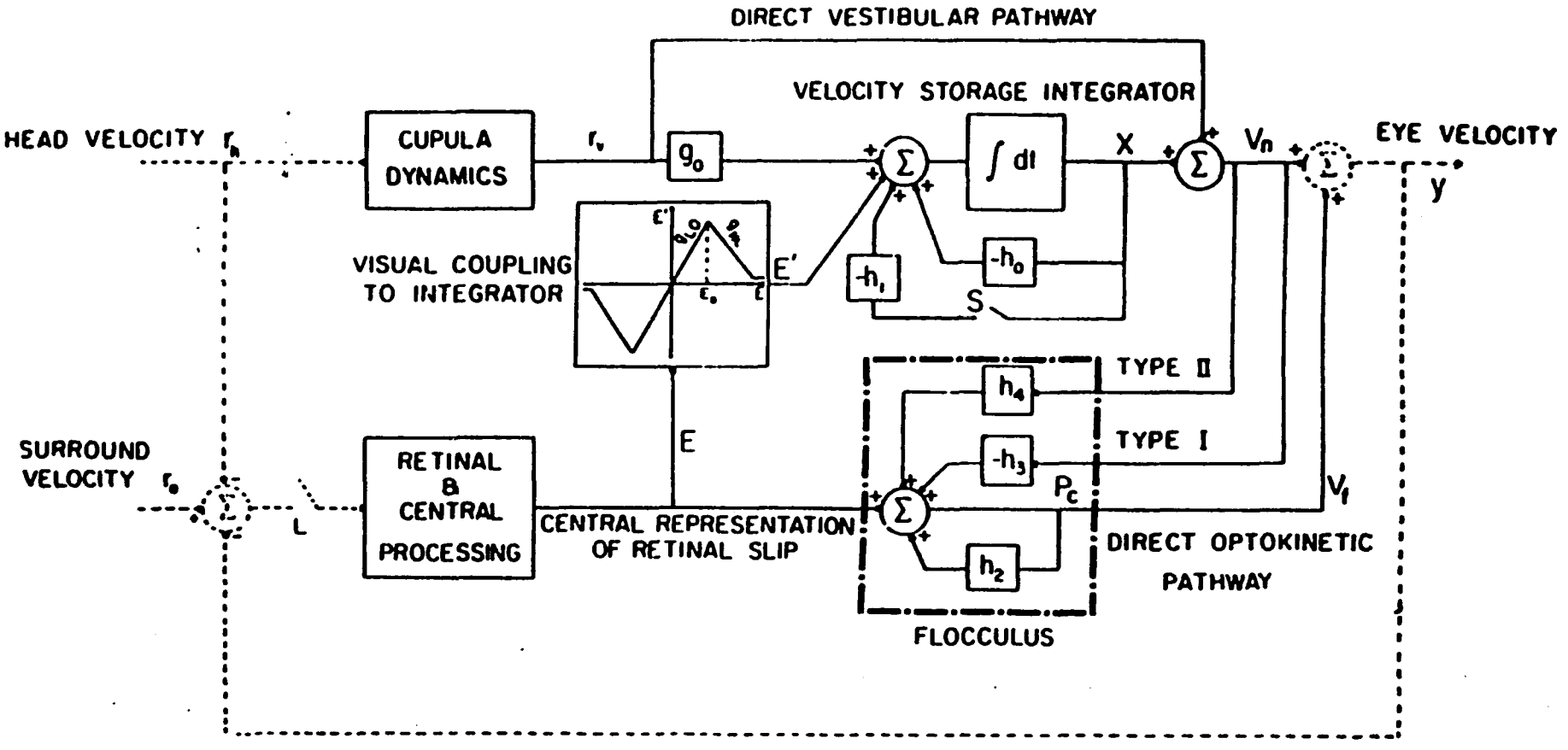


Figure 2.3 Model of OKN, OKAN, per- and post-rotatory nystagmus and visual-vestibular interactions. Solid lines represent neural signals, dotted lines represent mechanical variables. Taken from Waespe et al., 1983.

r_v, that appears in semicircular canal afferents in the vestibular nerve. This information activates the integrator, as well as projecting around it, to form a component of the eye velocity command signal in the vestibular nuclei V_n. The time constant of the integrator is equal to $1/h_o$. OKN is initiated by the velocity signal ro, generated by movement of the visual surround. From this signal is subracted head velocity and eye velocity, whose sum is gaze velocity. This generates the retinal slip signal e. The slip signal can be extinguished by the light switch L, or transmitted centrally to two elements. One is the direct pathway that is responsible for rapid changes in eye velocity. The second is a nonlinear function whose output activates the velocity storage integrator (visual coupling to the integrator). The suppression switch S in the model is utilized to discharge or 'dump' the integrator rapidly during visual or tilt suppression.

Model predictions from the one dimensional model (Raphan et al, 1979) are shown in figure 2.4. The eye velocity predictions corresponding to animal rotation in darkness, drum rotation in light with the subject stationary, and for animal rotation in light are shown in figure 2.4a, 2.4c, and 2.4e, respectively (compare with figure 2.2). Figures 2.4b, 2.4d, and 2.4f show the components which summate to form the velocity, namely the cupula, and integrator activity. These simulations show that the model can simulate the visual and vestibular responses of the oculomotor system to a wide range of stimulus conditions.

In summary, we have shown the significance of the velocity storage integrator in mediating the vestibular nystag-

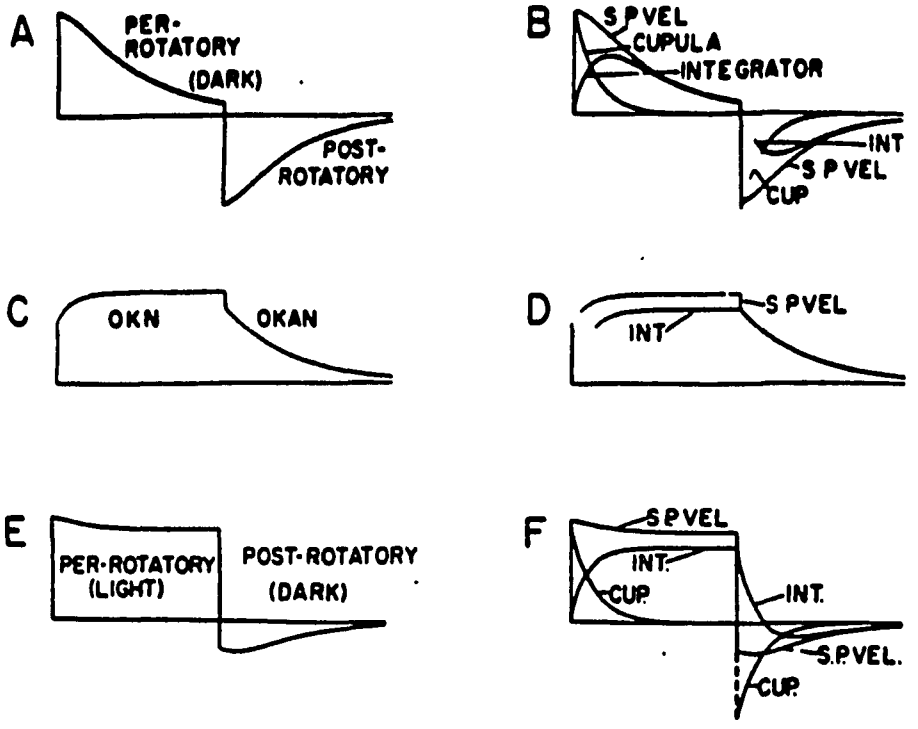


Figure 2.4 Model predictions of slow phase eye velocity for a step of angular velocity in darkness (A), for a step of surround velocity (C) and for a step of angular velocity in light (E). B, D, F, Comparative changes in slow phase velocity, cupula deflection, and output of the integrator for the responses shown in A, C, and E, respectively. Taken from Raphan, et al., 1979.

mus, OKN, and OKAN in one dimension. The next section considers the role of the velocity storage integrator in three dimensional models of the VOR and the coupling to it by the peripheral labyrinth.

2.3 Three Dimensional Models of Visual Vestibular Interaction

2.3.1 Introduction

Three dimensional models of the VOR and visual-vestibular interaction have considered both the static and dynamical aspects of the transformations involved in generating compensatory eye velocity. Robinson (1982) used vectors and matrices to represent the rotations and transformations that occur in three dimensions. He assumed that the brainstem calculates in canal coordinates, transforming the input which is in head coordinates into a canal based system (Simpson et al, 1981). The next transformation applied is the brainstem function. Finally, the signals are translated into muscle coordinates. Thus, the entire calculation is reduced to three matrix multiplications.

A similar approach is presented by Pellionisz (1985). He deals with the transformations as tensor transformations to account for the possible non-orthogonalities of the coordinate bases of the semi-circular canals. Although the approaches taken by Robinson (1982) and Pellionisz (1985) deal with the coordinate transformations that are necessary to generate compensatory eye movements in three dimensions, they do not address the question of how the dynamics of the system are considered in a three dimensional construct.

Hain (1986) utilized the Robinson model and added dynamics into the three dimensional system. However, the formulation does not lend itself to comparisons with experimental data and does not give a clear picture of how the parameters of

the system are related to spatial orientation. All of these approaches to modelling the three dimensional behavior of the vestibulo-ocular reflex do not clearly identify the relationship between the dynamic structure of the system and the orientation of the head with regard to gravity.

A generalized state space model representing the structure and function of visual-vestibular interaction describing optokinetic nystagmus (OKN), optokinetic after-nystagmus (OKAN), and visual-vestibular interactions has been developed by Raphan (Raphan & Cohen,1985; Raphan & Cohen, 1988). It is a natural extension of the one dimensional model with the system parameters being represented by matrices. In addition to performing the static transformation that converts input in head coordinates to semicircular canal coordinates and back to head coordinates, the model provides a basis for the dynamical transformations that occur when the head and the visual surround are moved in three dimensions. In this thesis we will develop algorithms that relate the parameters of a three dimensional representation of the velocity storage integrator to eye velocity components as a function of gravity. First we will consider the static and dynamical transformations of the overall model and their physiological basis.

2.3.2 Physiological Basis for Static Transformations

The inner ear contains 3 essentially orthogonal semicircular canals and 2 otolith organs, the utricle and saccule (see figure 2.1a). The semicircular canals respond to angular accelerations and the otoliths sense linear accelerations and static head position relative to gravity.

The semicircular canals are composed of three rings, the anterior, posterior, and lateral canals. Their nearly orthogonal structure allows them to detect angular accelerations about any axis. As reported by Steinhausen (1933), they function as approximate integrators whose output reflects angular velocity.

The semicircular canals form nearly parallel complementary pairs on either side of the head (Reisine et al, 1988). Each of the canals at its intersection with the utricle has an ampulla which contains the crista ampullaris (Figure 2.5). The multiciliated sensory cells of the ampulla combine and project into the cupula. The cupula is a membrane which seals the the lumen of the ampulla.

As the head is rotated in the plane of a canal, movement of the semicircular canal fluid will be accompanied by movement of the cupula which bends the bundles of cilia. Each bundle consists of many stereocilia located approximately to one side of the one long kinocilium (Hudspeth, A.J & D.P. Corey, 1977). The vector with direction from the short stereocilia to the kinocilium represents the polarization vector. In a given ampulla essentially all the morphological polarization vectors are parallel. As the bundle of cilia bends in the

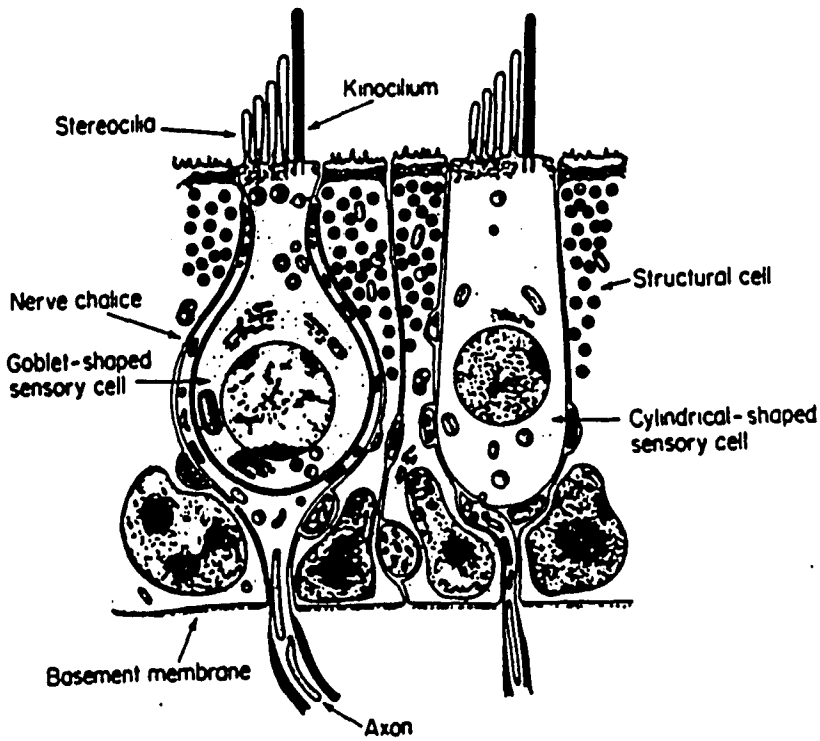


Figure 2.5 A crista ampullaris

Detailed structure of a cross-section of the crista ampullaris showing the stereocilia and the kinocilium. Taken from Howard (1982) p. 345.

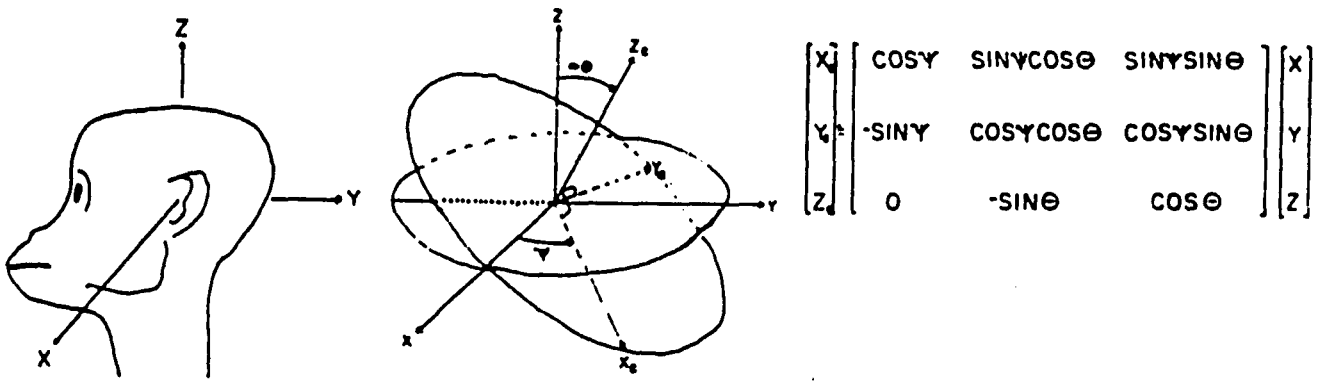
direction of this polarization vector, there is an excitation of the primary afferent signal whereas an inhibition of the primary afferent signal is associated with the bundle of cilia bending away from the polarization vector.

The signals arising in the peripheral organs are projected to the vestibular nuclei. The VIIIth nerve carries the canal and otolith afferents to the medial, superior, and lateral vestibular nuclei. Not only are these signals coded in a canal based coordinate frame, there is evidence that the visual system also processes signals in canal coordinates (Simpson & Graf, 1981). Therefore, this coordinate frame is of significance in determining the function of the VOR and visual vestibular interaction. Since it is the velocity storage integrator which is the focus of visual vestibular interaction, it is likely that this mechanism also processes information in a canal based coordinate frame.

In the upright position, the canal planes are tipped up approximately 15 degrees and are rotated 45 degrees from the frontal plane of the head. The measurement of eye movements is generally done in a head based coordinate frame. The orientation of the canal based coordinate frame with respect to the head is given by the euler angles phi = 0, theta = -15, and psi = 45 (Appendix). The matrix transformation is then given by (Raphan & Cohen, 1985; See figure 2.6):

$$T_{can} = \begin{vmatrix} 0.707 & 0.683 & -0.183 \\ -0.707 & 0.683 & -0.183 \\ 0.0 & 0.259 & 0.966 \end{vmatrix} \qquad (2.3.1)$$

While there is evidence that the semicircular canals estab-



XYZ - HEAD BASED CO-ORDINATES
$X_cY_cZ_c$ - CANAL BASED CO-ORDINATES

X_c - NORMAL TO ANTERIOR CANAL PLANE
Y_c - NORMAL TO POSTERIOR CANAL PLANE
Z_c - NORMAL TO HORIZONTAL CANAL PLANE

Figure 2.6 Orientation of the semicircular canals with respect to the head system coordinate frame. Taken from Raphan & Cohen, 1985.

lish the coordinate frame for combining signals in the VOR, gravity plays an important role in controlling the parameters of the system (Raphan & Cohen, 1988). This is done through the otolith organs as well as through somatosensory input (see Wilson & Melvill Jones for review). While the somatosensory control of the VOR is not well understood, more is known about how the otoliths control the VOR.

The otolith organs are responsible for detecting orientations with respect to gravity, as well as linear acceleration. Each side of the head contains a utriculus and sacculus which are situated at the intersection of the three semicircular canals (Figure 2.1a). It is the hair cells within the utricles and saccule, which respond to the magnitude and direction of gravity. The main part of the utricular macula is approximately parallel to the horizontal semicircular canal while its anterior end curves upward. The saccular macula is approximately perpendicular to the utricular plane (Figure 2.7a). The utriculus responds to accelerations in the horizontal plane while the sacculus responds to accelerations in the vertical plane. In addition, there is evidence that there is otolith activity in the central nervous system which interacts with neural circuits which are in a canal based coordinate frame (Melvill-Jones & Milsum, 1969).

Each macula contains a layer of bundled ciliated cells similar to the cristae of the semicircular canals which extend through the otolith membranes. These membranes contain calcite crystals called the otoliths (Figure 2.7b). Each bundle of cilia consists of stereocilia and one kinocilium. The polarization vector for each bundle is determined by the vector from

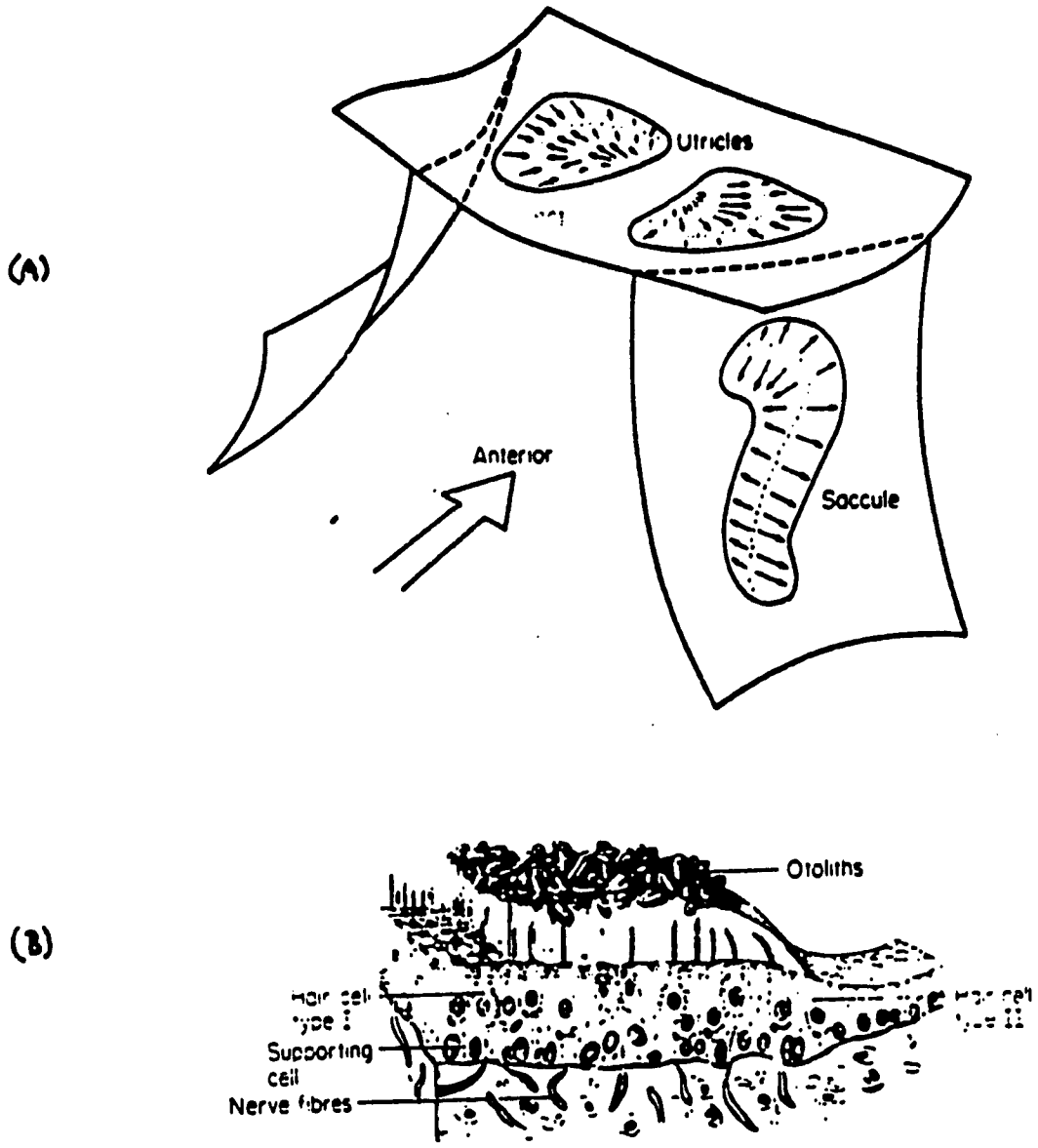


Figure 2.7 The Otolith Organs

(A) Relative positions of the utricles and saccule with the polarization maps. The arrows point in the direction of the kinocilium.

(B) Detailed structure of the macular epithelium showing the placement of the otoliths
Taken fromHoward (1982) p. 355.

the stereocilia to the kinocilium. Forces along such polarization vectors maximally excite the afferents projecting from the bundle (Hudspeth & Corey 1977). These projections give the central nervous system information about head orientations with regard to gravity (see Melvill Jones and Wilson, 1979). They provide the information that modifies the dynamical properties of motion processing in the velocity storage integrator (Raphan et al, 1988) and in estimating head velocity in gravitational environments (Raphan & Schnabolk, 1988; Fanelli et al, 1990).

In summary, the semicircular canals establish a coordinate basis for the velocity storage integrator in which computations during visual vestibular interaction can take place. The otoliths modify the coupling between neural circuits coding the states of the velocity storage integrator to form the relationship between the dynamics of integrator function and gravity.

We next review the experiments that relate the dynamical properties of velocity storage in three dimensions to the processing of gravity information by the otoliths.

2.3.3 Behavioral Basis for Dynamic Transformations

The main studies that have brought into evidence the three dimensional structure of the velocity storage integrator are those showing that there is coupling between storage associated with nystagmus in the horizontal plane and vertical planes (Raphan & Cohen,1983; Raphan & Cohen, 1988). When the subject is upright, horizontal OKN and OKAN are produced in the same direction as the surround and there are no significant vertical or roll components generated during OKN or the subsequent OKAN (Figure 2.8). In contrast, when monkeys are on their sides and given pure horizontal stimulation, a vertical component of OKAN appears, whereas no vertical optokinetic nystagmus was induced (Raphan & Cohen, 1983; Raphan & Cohen, 1988) (Figure 2.9a, see downward arrow). The rate of decline of the slow phase velocity of the OKAN was that of the velocity storage integrator. In addition, roll eye movements are generated due to pure horizontal stimulation when animals are prone or supine (Raphan & Cohen, 1988) (Figure 2.9b, see upward arrow). In addition, there are secondary OKAN responses (upward arrow, Figure 2.9a, downward arrow, Figure 2.9b). These secondary responses are not encompassed by the three dimensional model (Raphan & Cohen, 1985) and are not considered in this dissertation.

Cross-coupling is also found in cats with similar characteristics to that of the monkey (Harris, 1987). Harris confirmed Raphan's finding that the vertical component of OKAN velocity is a function of the direction and tilt of the horizontal velocity. In addition, Harris showed that there was also cross-

Figure 2.8. OKN and OKAN- Upright

Surround movement is to the left at 60 degrees per second about the animal yaw axis. The traces are the horizontal, vertical, and roll eye position followed by the horizontal, vertical, and roll eye velocity followed by the movement of the visual surround and the status of the light. The OKN and OKAN are horizontal and there is no vertical or roll component. Taken from Raphan & Cohen, 1988.

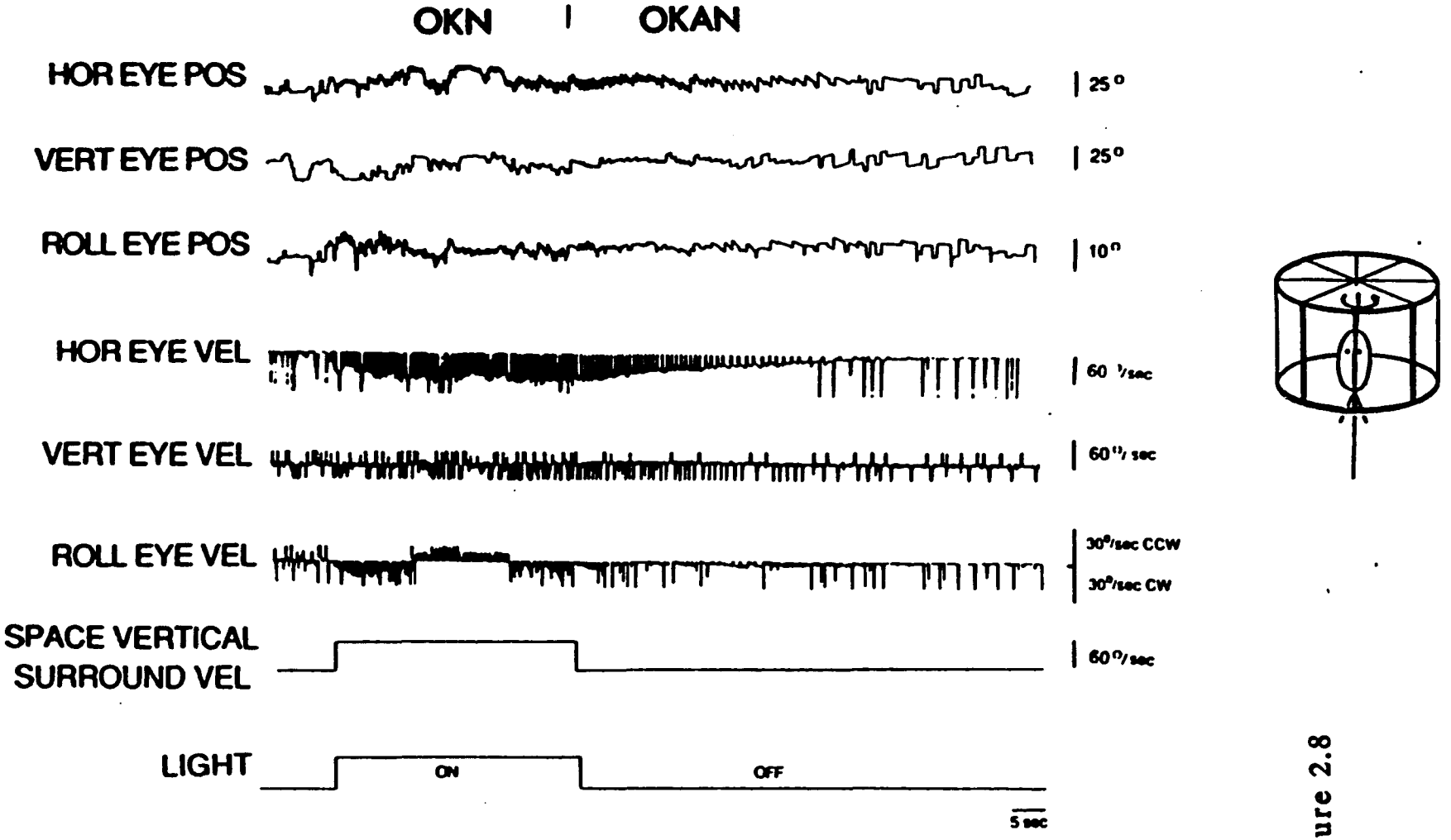

OKN
OKAN
HOR EYE POS
25°
VERT EYE POS
25°
ROLL EYE POS
10°
HOR EYE VEL
60 °/sec
VERT EYE VEL
60°/ sec
ROLL EYE VEL
30°/sec CCW
30°/sec CW
SPACE VERTICAL
SURROUND VEL
60°/sec
LIGHT
ON
OFF
5 sec

Figure 2.8

Figure 2.9 (A)Cross-coupling from yaw to pitch axis. OKN and OKAN induced by rotation of the visual surround to the right in the animal's yaw axis with the animal in the 90-degree roll position, right side down. There was a component of upward slow phase velocity during the OKN that increased during primary OKAN (downward arrow). The duration of the horizontal component was briefer than when the animal was upright. (Figure 2.8).(B) Cross-coupling from yaw to roll axis. OKN and OKAN induced by rotation of the visual surround to the right in the animal's yaw axis with the animal supine. The horizontal component of OKAN was brief, and a counterclockwise roll component developed. Taken from Raphan & Cohen, 1988.

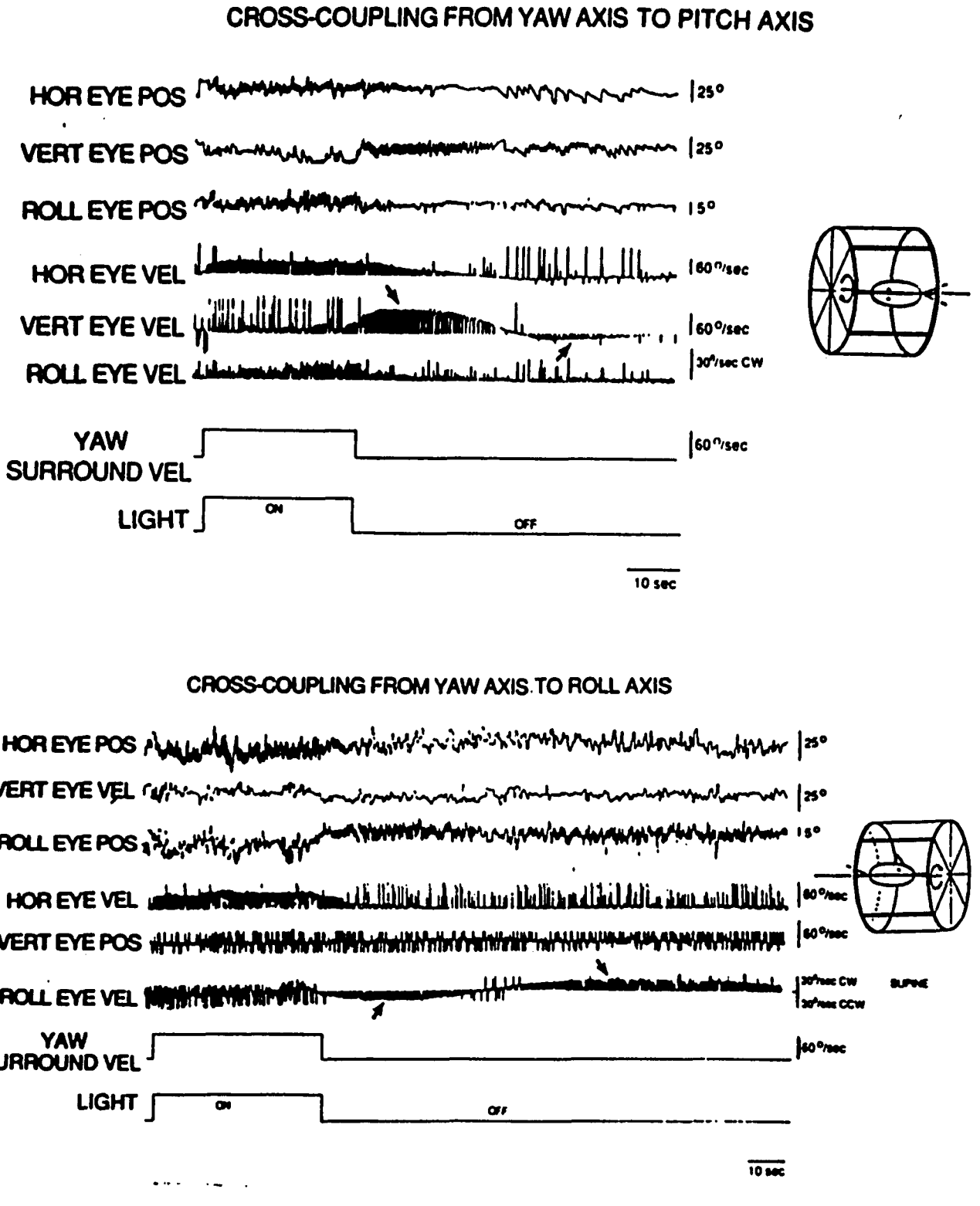

CROSS-COUPLING FROM YAW AXIS TO PITCH AXIS
HOR EYE POS
VERT EYE POS
ROLL EYE POS
HOR EYE VEL
VERT EYE VEL
ROLL EYE VEL
YAW SURROUND VEL
LIGHT
ON
OFF
10 sec
CROSS-COUPLING FROM YAW AXIS TO ROLL AXIS
HOR EYE POS
VERT EYE VEL
ROLL EYE POS
HOR EYE VEL
VERT EYE POS
ROLL EYE VEL
YAW SURROUND VEL
LIGHT
ON
OFF
10 sec

Figure 2.9

coupling of horizontal to vertical slow phase velocity during postrotatory nystagmus after off-vertical axis rotation when animals were stopped on their sides. This again points to the commonality of the velocity storage integrator in the generation of OKN and vestibular nystagmus. It also focuses the cross-coupling phenomenon as a gravity dependent property of a three dimensional velocity storage integrator. Thus, a better understanding of the velocity storage integrator and how it is affected by gravity will help to understand these phenomena. In addition, the coordinate frame that forms the basis for the central processing is also important in understanding the coupling to the velocity storage integrator in three dimensions.

2.3.4 Effects of Gravity on Human Perception of the Spatial Vertical

Raphan (1988) pointed out a relationship between the eigenvector alignment with the spatial vertical and human perception of the visual vertical. This relationship was explored and the original observations were borne out. Here we review what is known about human perception. In chapter 4, we show the similarities with the eigenvector hypothesis.

Man's central body axis, sometimes referred to as his idiotrophic vector (Mittelstaedt, 1986) is in line with gravity. Gravity is our main frame of reference and in conjunction with the horizon define our spatial coordiate frame. Investigators have studied variances from this paradigm including tillted frames of reference and shifting body orientations.

How body tilt affects perceptions of the spatial vertical and horizontal has been studied in depth from the late 1800's until the present (Aubert, 1861, Meuller, 1916, as reported in Schöne, 1984; Mittelstaedt, 1986). It has been found that the subjective or perceived vertical is usually close to the true vertical.

When a human subject is upright, he is able to judge the true vertical (gravity vector) within one degree (Witkin and Asch 1948; Mann et al 1949; Kaufman et al 1949). As the subject tilts, his perception of the vertical is much less accurate.

Aubert (1861) discovered that as one tilts, the spatial vertical appears to tilt in the same direction as ones head. Meuller (1916) refined this theory by analyzing responses to

small tilts versus large tilts. He found that for small tilts (less that 40 degrees) subjects often overcompensated and set the spatial vertical to the side opposite to head tilt close to the spatial vertical. For large tilts they undercompensated, with the perceived vertical following the direction of the head. The Aubert effect is referred to as the A-effect and the Meuller as the E- effect (Figure 2.10).

One outcome of our analysis of the velocity storage integrator was to suggest that there is a common neural basis for the generation of the perception of the true vertical and the orientation of characteristic vectors or eigenvectors associated with velocity storage. Further investigation is necessary to uncover the neural basis for these gravitational phenomena.

We next show how the dynamics of velocity storage can be characterized by the eigenvalues and eigenvectors of the system matrix. This representation will form the basis for the identification scheme to identify the parameters of the system matrix associated with velocity storage.

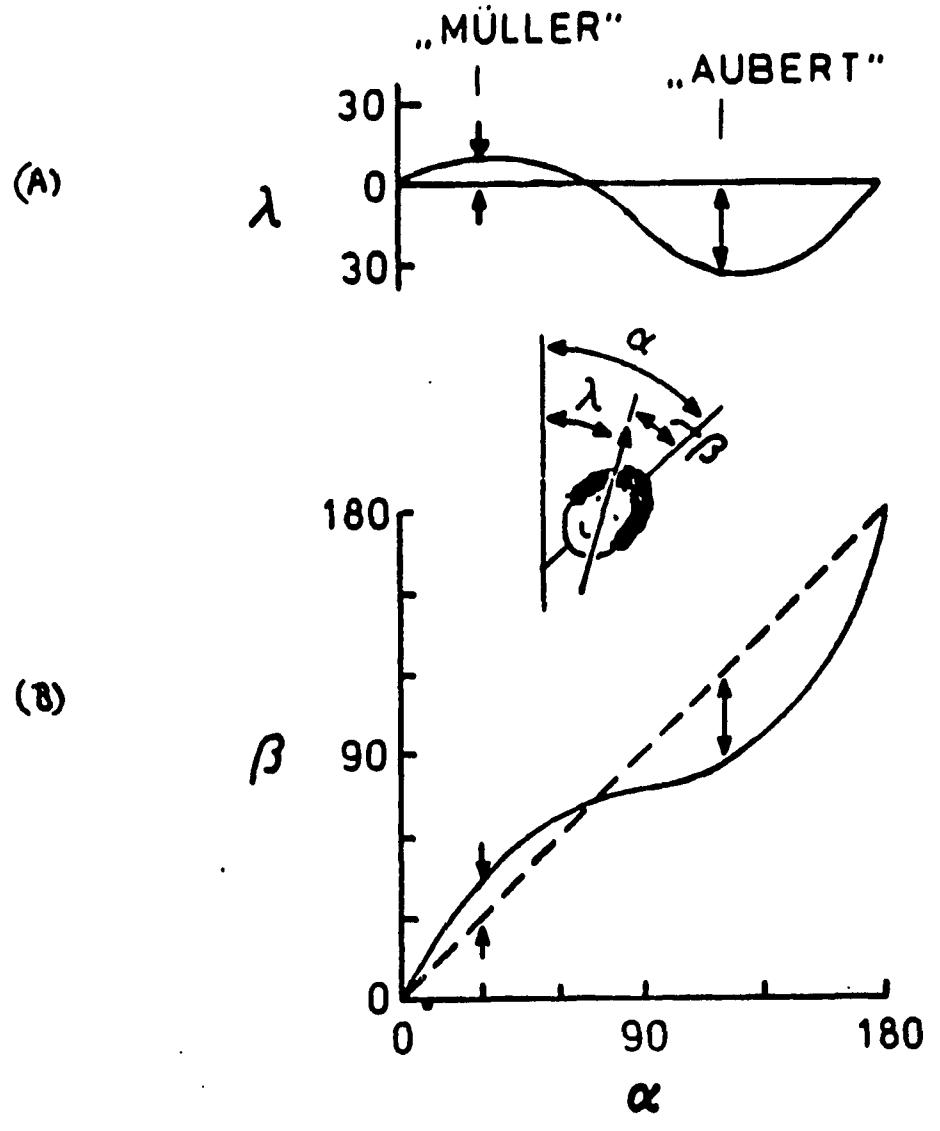


Figure 2.10 The Aubert-Muller Effect: The Muller effect is when $\beta > \alpha$ and the Aubert effect is when $\beta < \alpha$

(A) The angle between the subjective vertical and gravity (λ) as a function of body tilt relative to gravity (α).

(B) The angle between the the subjective vertical and the idiotrophic vector as a function of body tilt.

Taken from Schone (1984), p. 116.

2.3.5 Dynamic Aspects of the Velocity Storage Integrator in Three Dimensions

A model which is an extension of the one dimensional model to three dimensions is shown in Figure 2.11 (Raphan & Cohen, 1985). It represents the velocity storage integrator as a dynamical system with the system matrix H_o. This model structure has been shown to be capable of explaining the cross-coupling phenomena because of the off diagonal terms in the system matrix.

The dynamics of the velocity storage integrator in three dimensions is represented by the following matrix differential equations: (see Raphan & Cohen, 1985; see figure 2.11)

$$x' = H_o\, x + G_o\, rv + N\,(e) + r_{int} \qquad (2.3.2)$$

$$y_h = T_{can}^{-1} x \qquad (2.3.3)$$

$$r_v = D_{can}\, T_{can}\, r_h \qquad (2.3.4)$$

$$r_{int} = D_{oto}\, T_{oto}\, r_l \qquad (2.3.5)$$

$$e = L\,(T_{can}\,(r_o - r_h - y)\,) \qquad (2.3.6)$$

where x is the three dimensional state of the integrator, e is the retinal slip, N(e) is a nonlinear matrix operator on the central representation of retinal slip, G_o is the coupling matrix from the eighth nerve to the integrator, and H_o is the matrix representing the dynamics associated with the integrator.

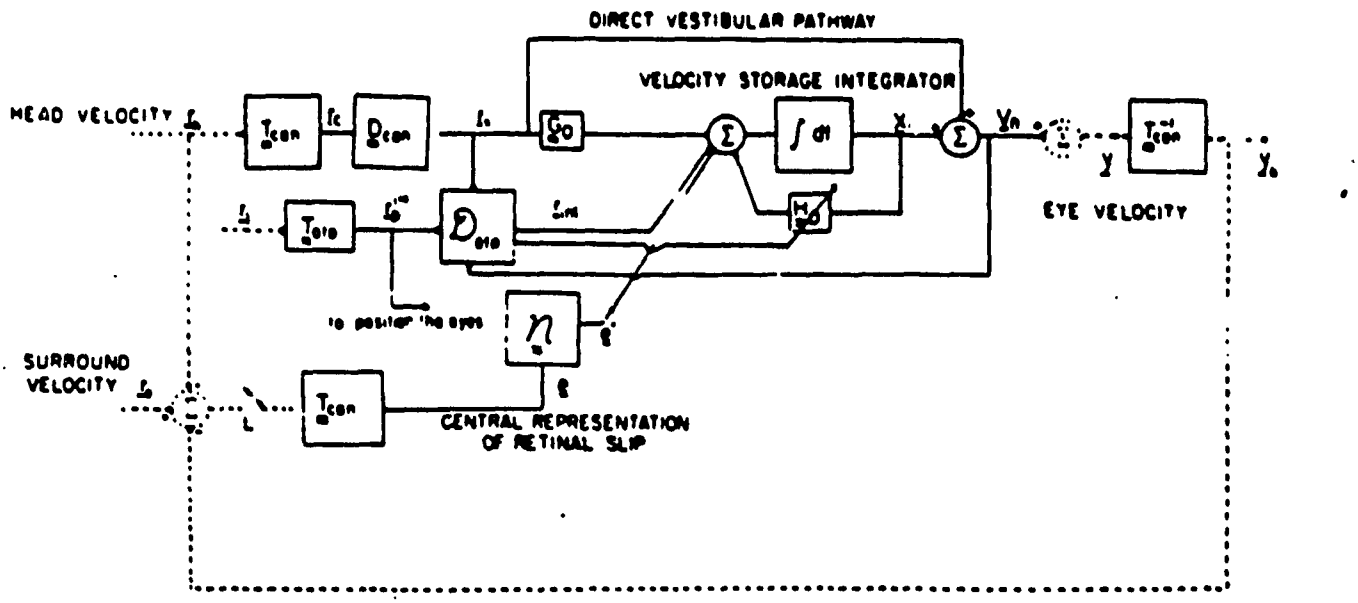


Figure 2.11 Three dimensional model of the vestibulo-ocular reflex. Taken from Raphan & Cohen, 1985.

The model has three inputs each of which is a three dimensional vector relative to space: a head angular velocity, r_h, a surround angular velocity, r_o, and a force, r_1. The output is the eye velocity relative to the head, y_h. The head coordinate system is defined by three mutually perpendicular unit vectors (Figure 2.6): a unit vector pointing up out of the top of the head (z-head direction); a unit vector pointing back out of the back of the head y-head direction); and a unit vector pointing to the left out of the left ear (x-head direction). The z, x, y components of yh in the head based coordinate frame correspond to horizontal, vertical and roll eye movements, respectively. The central nervous system signal processing is assumed to be done in a semicircular canal based coordinate frame. This is represented by the kinematic transformation T_{can} and T_{can}^{-1} which are transformations from head based to canal based coordinates and back to head based coordinates. The dynamics of the semicirular canals was discussed above (in section 2.3.2).

The model also contains various dynamical components: 1) a cupula dynamical system, D_{can} is used to transduce the head angular acceleration into a vestibular signal which is an internal representation of the change in head angular velocity, 2) a velocity storage integrator state (memory), and its system matrix H. The vector x is its output, 3) a nonlinear visual coupling to the integrator N(e), 4) an otolith system operator transforming patterns of sequential activations of otolith hair cells by the force, r_1, into an estimate of head angular velocity (Raphan & Schnabolk 1988; Fanelli, Raphan & Schnabolk, 1990). These components transform the inputs (rh, ro, r1) and

the present state (x) into the output (y_h) and a next state (see Raphan & Cohen, 1985 for a detailed description).

The internal coordinate system of the model is defined by three mutually perpendicular directions, each of which is normal to one of the three planes of the semicircular canals. It is assumed that the coordinate basis vectors are mutually orthogonal. The inputs, head and surround velocity, to the system are transformed into the canal coordinates by T_{can} and contribute to the canal coordinate representation of the output via two paths; first via an indirect pathway that goes through the velocity storage integrator and second via a direct pathway that goes around it. The direct pathway allows for a very rapid response of the eye to the inputs while the indirect pathway allows for a much slower response based on the state of the integrator, x and its system matrix H. The model then tranforms the canal coordinate representation of the eye velocity into the head coordinate representation of the eye velocity via Tcan inverse.

The conceptual basis for the structure of the velocity storage integrator in three dimensions was suggested by Raphan (Raphan & Cohen, 1985; Raphan et al, 1988). The velocity storage component of the model has been represented as a dynamical system, which can be described using a state space approach.

In order to gain insight into the structure of the system matrix as a function of gravity, we will examine the zero input response and the modes inherent in the solution (Zadeh & Desoer, 1963). The zero input response is important since it is

precisely the characteristic response of the integrator and is analogous to the condition during OKAN. This is because during OKN the integrator is charged to some value which becomes the initial condition for the response during OKAN. It is also important because the head is fixed with regard to gravity during this condition. The zero input response is obtained by setting the input head velocity $r_h = 0$, the optokinetic input $ro = 0$, opening the switch L in the model (Figure 2.11), and solving the equation for a given initial condition. This reduces the model to that shown in Figure 2.12. Therefore, the operative equations for modelling the subsequent OKAN become:

$$x'(t) = H\,x(t) \qquad (2.3.7a)$$

and,

$$y(t) = T_{can}^{-1}\,x(t) \qquad (2.3.7b)$$

subject to the initial state, $x(t_o) = x_o$

If one examines the measured output y representing eye velocity (see model) for the zero input response an equation can be derived which governs its dynamic characteristics. From equation 2.3.7b we obtain,

$$x = T_{can}\,y \qquad (2.3.8)$$

Therefore,

$$x' = T_{can}\,y' \qquad (2.3.9)$$

Velocity Storage Integrator

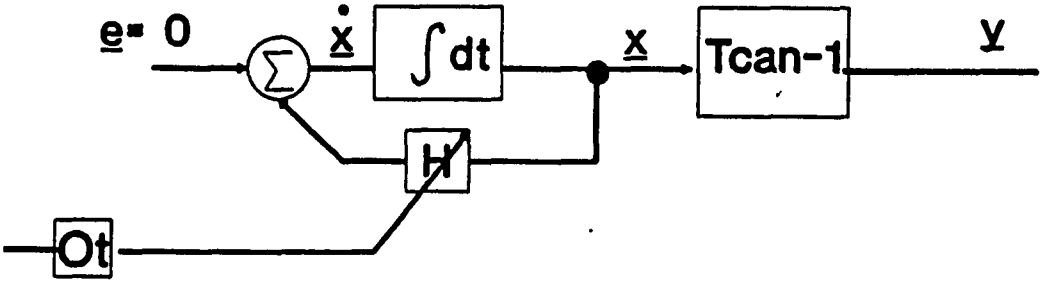


Figure 2.12 System configuration for computing the zero input response.

Substituting equations (2.3.8) and (2.3.9) into equation (2.3.7b), the dynamic equation for the output y is given by:

$$y' = (Tcan^{-1}\, H\, Tcan)\, y \qquad (2.3.10)$$

Thus, a matrix H_s can be defined which governs the zero input response for the output measured in head coordinates. It is a similarity transformation of the integrator system matrix H in canal coordinates and is given by:

$$H_s = T_{can}{}^{-1}\, H\, T_{can} \qquad (2.3.11)$$

The effect of gravity on the parameters can best be explored by examining the structure of the matrix H

$$H = \begin{vmatrix} h11 & h12 & h13 \\ h21 & h22 & h23 \\ h31 & h32 & h33 \end{vmatrix} \qquad (2.3.12)$$

If H has distinct eigenvalues $\lambda 1$, $\lambda 2$, $\lambda 3$, then Hs which is similar to H will have identical eigenvalues (Lang, 1966; Zadeh & Desoer, 1963). The solution to equation (2.3.7a) is given by:

$$x(t) = S\, e^{\Lambda(t-to)}\, S^{-1}\, x_o \qquad (2.3.13)$$

and the output y is given by,

$$y(t) = (T_{can}{}^{-1}\, S\, e^{\Lambda(t-to)}\, S^{-1}\, T_{can})\, y_o \qquad (2.3.14)$$

where S is the modal matrix, Λ is the diagonal matrix whose

elements are the eigenvalues of H, S^{-1} = (n1,n2,n3) such that $<ni,nj> = 0, i \neq j$, $<ni,ni> = 1$ and S = (u1,u2,u3) (Director and Rohrer, 1972; Zadeh and Desoer, 1963;)

Since u_1, u_2, u_3 are unit eigenvectors associated with the eigenvalues of the matrix H then the operation of H on these eigenvectors are along the eigenvector, i.e.:

$$H u_i = \lambda_i u_i \qquad i = 1,2,3 \tag{2.3.15}$$

If the intitial state, x_o, is chosen along an eigenvector, then,

$$x_o = au_i \qquad \text{where a is a constant} \tag{2.3.16}$$

and,

$$y_o = T_{can}^{-1} au_i \tag{2.3.17}$$

Substituting equation (2.3.16) and (2.3.17) into equations (2.3.13) and (2.3.14) gives a solution

$$x(t) = a S e^{\lambda i(t-to)} S^{-1} u_i$$

$$= a e^{\lambda i(t-to)} u_i \tag{2.3.18}$$

and,

$$y(t) = a (T_{can}^{-1} S e^{\lambda i(t-to)} S^{-1}) u_i$$

$$= a e^{\lambda i(t-to)} (T_{can}^{-1} u_i) \tag{2.3.19}$$

In summary, if the initial state of the system is along an eigenvector the solution will lie along the eigenvector in the state space for all time (Zadeh & Desoer, 1963).

The approach we have taken in this research is to associate the orientation of the eigenvectors of the integrator's state space with that of three dimensional space as oriented by gravity. We have mathematically characterized the three dimensional structure of the system matrix, H, as a function of orientation and have identified transformations so as to maintain the approximate invariance of the yaw eigenvector in state space close to the spatial vertical. Algorithms were devised to identify the parameters of the system matrix from experimental measurements, which maintain this invariance. The system matrix was then incorporated into the three dimensional model of the vestibulo- ocular reflex and we were able to simulate the cross-coupling during OKN and OKAN.

2.3.6 Summary

1. The vestibular system generates accurate compensatory eye movements of the head over a wide range of angles and frequencies. As shown above, the transformation of the head velocity signal into the neural signals that code head movement are known for the various sensors such as the semicircular canals and otoliths. The dynamic transformations that take place in the central vestibular system to combine visual and vestibular information in order to produce compensatory eye movements for head rotations about any axis have not been explored in depth.

2. Raphan & Cohen (1988) have shown that the implications of the cross coupling from the horizontal to the vertical and roll planes, is that velocity storage is a gravity dependent three dimensional system. A generalized three dimensional model representation of the VOR and visual vestibular interaction has been developed by Raphan and Cohen (1985).

3. In order to understand the performance of the system in three dimensions, the matrices associated with the model are identified as a function of gravity. Of special interest is the matrix associated with the velocity storage integrator. A mathematical structure has been developed that relates the matrices associated with the three dimensional model to the orientation of the head with regard to gravity.

4. From this work it has been suggested that the yaw axis of storage, or the yaw eigenvector, tends to align with gravity (Raphan & Cohen, 1987; Sturm & Raphan 1988) as the head is

tilted.

In the next chapter we parameterize the representation of velocity storage in terms of its eigenvalues and eigenvectors and develop the identification procedures to relate the parameters of the velocity storage matrix to data.

3. Identification of System Matrix associated with Velocity Storage

3.1 Structure of the System matrix H and Formation of the System Transformation as a Function of Gravity

A detailed representation of the three dimensional velocity storage integrator and the parametric control the otoliths have on the integrator's parameters is shown in figure (3.1). The three dimensional integrator is represented by three integrators whose outputs are the states of the system and are represented by $\underline{x} = (x_1, x_2, x_3)$. The parameters h_{ij} of the system are the coupling elements between the states and correspond to the elements in the system matrix in a canal based coordinate frame. The parameters are functions of gravity as sensed by the central otolith projections. These values at any given head orientation determine the cross coupling between states of the integrator that will be induced during OKAN. This models the effects of gravity in modifying OKN and OKAN.

In the upright position, the system matrix, in a head based coordinate frame, i.e. the frame of reference in which eye movements are measured, is denoted by H_o. It and its similar counterpart in a canal based coordinate frame, have 3 eigenvectors, one associated with the body vertical or yaw axis, another with the pitch axis, and the third with the roll axis. The system matrix in the canal frame of reference is given by $H_{co} = Tcan\ H_o\ Tcan^{-1}$. The eigenvalues associated with these eigenvectors correspond to the time constants of yaw, pitch and roll OKAN

Figure 3.1 Implementation of the three dimensional velocity storage integrator. The three dimensional integrator is represented by three integrators whose outputs are the states of the system, x_1, x_2, x_3. The inputs to the system are e_a, e_p, e_l which represent central retinal slip signals along anterior, posterior and lateral canal coordinates. The other inputs are the accelerations due to gravity, r_{la}, r_{lp}, r_{ll}. The parameters of the system are the coupling elements h_{ij}, among the states and correspond to the elements in the system matrix H. The parameters h_{ij} are functions of gravity as sensed by the central otolith projections. Their values at any given head orientation determine the cross-coupling among states of the integrator that will be induced during OKAN.

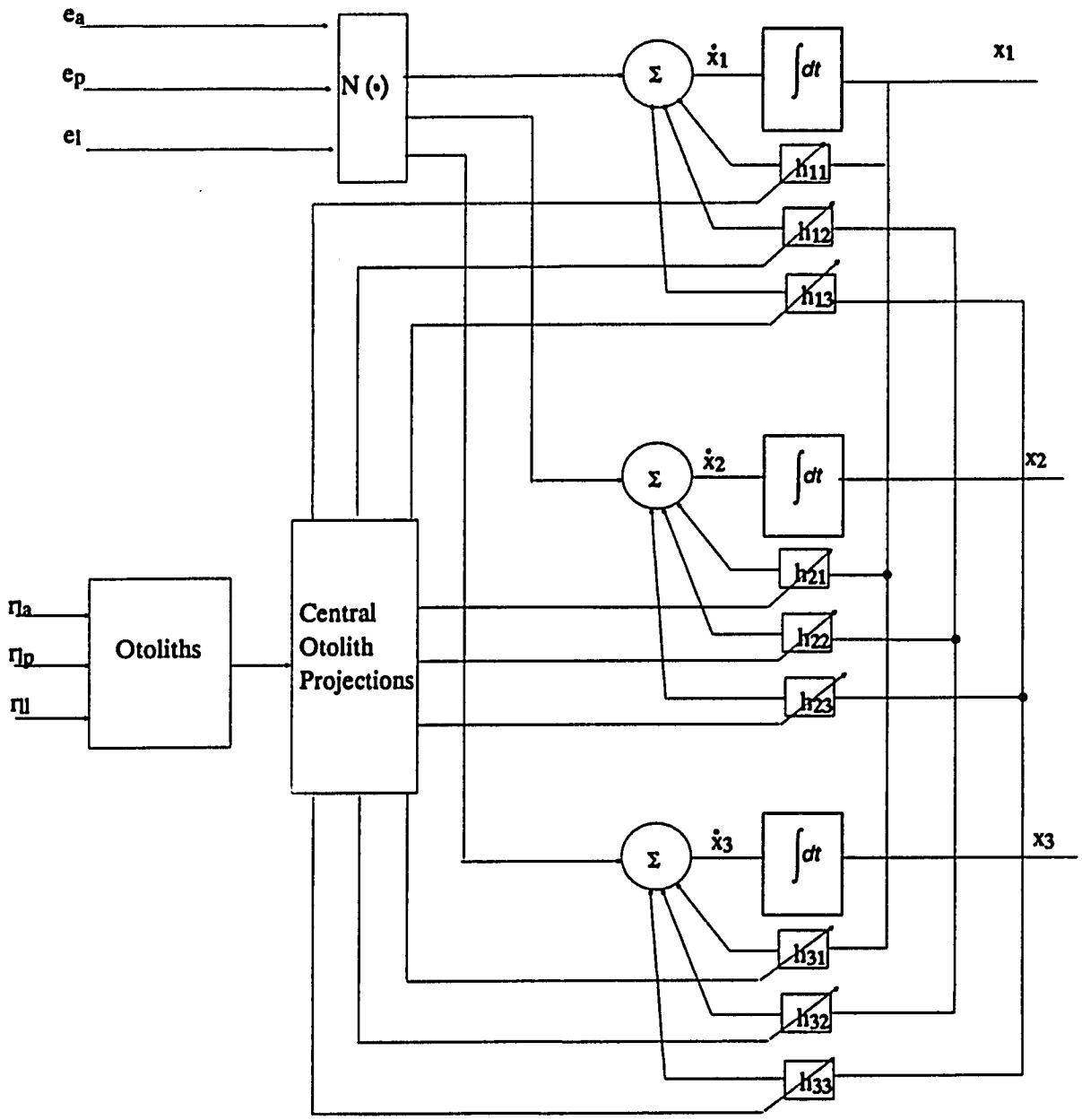

e_a
e_p
e_l
N (•)
Σ
$\dot{x}_1$
$\int dt$
x_1
h_{11}
h_{12}
h_{13}
Σ
$\dot{x}_2$
$\int dt$
x_2
h_{21}
h_{22}
h_{23}
Σ
$\dot{x}_3$
$\int dt$
x_3
h_{31}
h_{32}
h_{33}
η_a
η_p
η_l
Otoliths
Central Otolith Projections

Figure 3.1

with the head in the upright position. Because there is little or no cross-coupling in the upright position, the eigenvectors of H_o are assumed to be in line with the spatial coordinate unit vectors, as determined by the gravity vector.

Based on the experimental finding that the long time constant of OKAN or velocity storage is mainly associated with axes close to the spatial vertical, we must find a transformation of H_o that rotates the pitch and yaw eigenvectors with the head but maintains the yaw axis eigenvector approximately spatially invariant. We will consider the effects of the rotation on matrices H_o and its similar matrix H_{co} as the head is tilted using equations (2.3.10) and (2.3.11) to obtain the system matrix for the tilted positions. As the subject is tilted, the head frame of reference changes relative to the spatial coordinate frame. In addition, only two of the eigenvectors associated with the corresponding system matrix are altered although the subject is rotated. The vertical eigenvector approximately maintains spatial invariance. Therefore, a series of transformations will be applied to H_o to obtain the system matrix for the tilted position in a frame of reference in which eye velocity is measured i.e. head coordinates.

Let P be a mapping from 3 space to 3 space such that

$$\begin{aligned} P(e_1) &= R_1(e_1) = a_{11}e_1 + a_{21}e_2 + a_{31}e_3 \\ P(e_2) &= R_2(e_2) = b_{12}e_1 + b_{22}e_2 + b_{32}e_3 \\ P(e_3) &= R_3(e_3) = c_{13}e_1 + c_{23}e_2 + c_{33}e_3 \end{aligned} \qquad (3.1.1)$$

where R_1, R_2, R_3 are rotation mappings and e_1, e_2, e_3 are the usual basis associated with the spatial coordinates, i.e., e_1 =

$(1,0,0), e_2 = (0,1,0)$, and $e_3 = (0,0,1)$, where e_3 is opposite to gravity. Therefore, the matrix representing the transformation P is given by

$$\begin{vmatrix} a_{11} & b_{12} & c_{13} \\ a_{21} & b_{22} & c_{23} \\ a_{31} & b_{32} & c_{33} \end{vmatrix} \qquad (3.1.2a)$$

$$\text{or } P = (R_1e_1, R_2e_2, R_3e_3) \qquad (3.1.2b)$$

The matrix P can also be considered as a basis changing transformation where R_1e_1, R_2e_2, and R_3e_3 are the new basis vectors. Therefore, H_o the system matrix corresponding to the upright condition becomes transformed under this mapping by a similarity transformation related to the angle of tilt (Lang, 1963; Appendix).

In addition to changing the basis of the system matrix and its eigenvectors, gravity also modifies the time constants or eigenvalues of the system matrix. This entails multiplying the matrix H_o by a matrix Q which is a function of gravity. Since the multiplication is done in the eigenvector basis, both H_o and Q are diagonal matrices.

Thus, the new system matrix can be given as

$$H_p = PQH_oP^{-1} \qquad (3.1.3)$$

where P and Q are gravity dependent, and Hp is the system matrix relative to the upright spatial basis, which is aligned with the spatial coordinate axes, e_1, e_2, e_3.

In order to express eye velocity in the coordinate basis in which it is usually measured i.e. head coordinates, a transformation must be made to the subject reference frame. This is a similarity transformation by a rotation matrix R, where R rotates the subject head frame into the spatial frame. Therefore the system matrix relative to the head coordinate basis is given by:

$$Hs = R^{-1} Hp R \qquad (3.1.4)$$

By substituting equation (3.1.3) into equation (3.1.4) and using equation (2.3.11), the system matrix is given by:

$$H = T_{can} (R^{-1}(PQH_oP^{-1}) R) T_{can}^{-1} \qquad (3.1.5)$$

Note that H and H_o are not representations of the same linear transformations in different coordinate frames. Rather, P and Q explicitly change the eigenvalues and eigenvectors of the original system matrix.

Figure (3.2) shows an example of the system matrix calculated for three different orientations. In the upright condition (Figure 3.2a) the eigenvectors are the spatial coordinates (e_1, e_2, e_3) and the system matrix is diagonal. No cross coupling is induced. The associated eigenvalues were set approximately equal to -1/12 for the yaw axis, -1 for the roll axis and -.75 for the pitch axis. These correspond to the 12 second time constant of OKAN about a vertical axis and the approximate one second time constant for vertical and roll OKAN when the animal is upright. For a tilt of 45 degrees (Figure 3.2b), the yaw eigenvector is approximately -15 degrees while the pitch

and roll eigenvectors rotate with the head. The off diagonal component produces the cross-coupled horizontal to vertical component. When the subject in the 90 degree right ear down position (Figure 3.2c), the vertical eigenvector rotates approximately 10 degrees in the direction of tilt. The off diagonal component is larger, and therefore the cross-coupling is more pronounced.

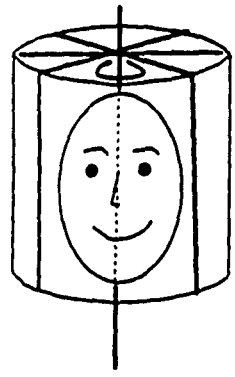
A summary of the various system matrices in their coordinate bases is given in table 3.1.

Matrix	Coordinate Basis	
H_o	Spatial	System matrix in spatial coordinates for upright condition
H_{co}	Canal	System matrix in canal coordinates for upright condition
H_p	Spatial	System matrix for arbitrary orientations in spatial basis P^{-1}:spatial --> eigenvector
QH_o	Eigenvector	System matrix for arbitrary orientations in eigenvector basis (Diagonal matrix)
H_s	Subject head	System matrix for arbitrary orientations in head coordiates R^{-1}:spatial --> head

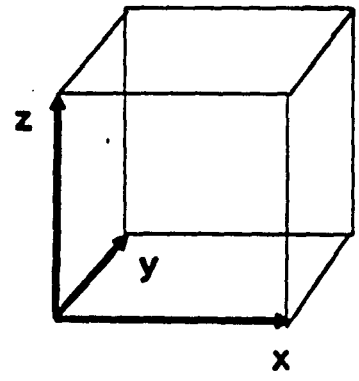
Table 3.1 Matrices and their coordinate bases

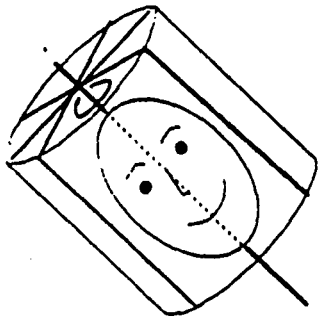
Figure 3.2 Calculating the system matrix. The system matrix for any head orientation is computed by performing a series of transformations on the original system matrix for the upright condition. The new system matrix, in a head coordinate frame is, $H = R^{-1}(PQH_0P^{-1})R$. It is assumed that there is an internal transformation to canal coordinates and back. Q is the transformation which changes the eigenvalues of the system. P is the transformation that changes the eigenvectors, and R transforms the system into head coordinates by performing a pure rotation.

Position	System Matrix	Eigenvectors

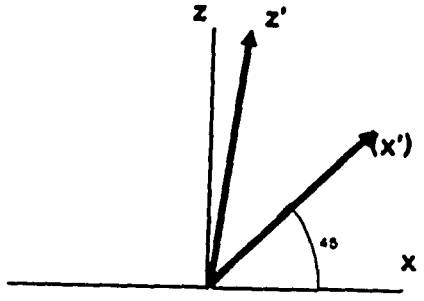


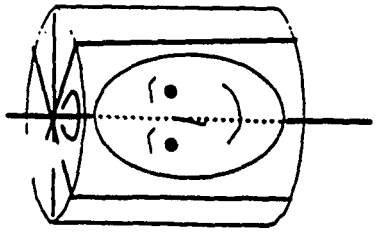
$$\begin{bmatrix} .75 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & .075 \end{bmatrix}$$





$$\begin{bmatrix} -.23 & 0 & .17 \\ 0 & -1 & 0 \\ 0 & 0 & -.09 \end{bmatrix}$$





$$\begin{bmatrix} -.2 & 0 & .416 \\ 0 & -1 & 0 \\ 0 & 0 & -.134 \end{bmatrix}$$

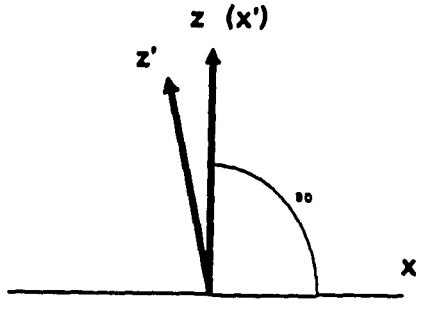


Figure 3.2

3.2 Dynamics of Transformed System

Because eye movements are measured in head coordinates, it simplifies the computation of the system matrix in terms of its eigenvalues and eigenvectors if the system matrix is represented in head coordinates. For the OKAN response, expressing the system matrix in head coordinates implies that the ouput velocity states will equal the velocity storage states ($y = x$). Equations (3.1.3) and (3.1.4) can be combined to give the following relationship between H_s and H_o:

$$H_s = R^{-1}PQH_oP^{-1}R \qquad (3.2.1)$$

It should be noted that P represents a transformation that maintains the yaw axis eigenvector spatially invariant in spatial coordinates while R^{-1} is the rotation that changes the spatial basis into the head frame. Thus, we can define a matrix T which is a transformation that maintains the yaw axis eigenvector spatially invariant and changes the basis to head coordinates. It is given by,

$$T = R^{-1}P \qquad (3.2.2)$$

We can express the relationship between the system matrix in the upright condition and the tilted condition by:

$$H_s = TQH_oT^{-1} \qquad (3.2.3)$$

The matrix differential equation governing this system, with the vector $\underline{x}$ also expressed in head coordinates is therefore

$$\begin{vmatrix} x_p' \\ x_r' \\ x_y' \end{vmatrix} = TQH_oT^{-1} \begin{vmatrix} x_p \\ x_r \\ x_y \end{vmatrix} \qquad (3.2.4)$$

where x_p, x_r, x_y are the pitch, roll, and yaw components of the state vector.

Multiplying each side of equation (3.2.4) by T^{-1} , we obtain:

$$T^{-1} \begin{vmatrix} x_p' \\ x_r' \\ x_y' \end{vmatrix} = Q\, H_oT^{-1} \begin{vmatrix} x_p \\ x_r \\ x_y \end{vmatrix} \qquad (3.2.5)$$

where,

$$QH_o = \begin{vmatrix} -\lambda 1 & 0 & 0 \\ 0 & -\lambda 2 & 0 \\ 0 & 0 & -\lambda 3 \end{vmatrix} \qquad (3.2.6)$$

and $\lambda 1$, $\lambda 2$, $\lambda 3$ are the eigenvalues of the system matrix H. Note that QH_o is a diagonal matrix since it is expressed in the eigenvector basis.

Equation (3.2.4) can be rewritten as,

$$\begin{vmatrix} X_p' \\ X_r' \\ X_y' \end{vmatrix} = \begin{vmatrix} -\lambda 1 & 0 & 0 \\ 0 & -\lambda 2 & 0 \\ 0 & 0 & -\lambda 3 \end{vmatrix} \begin{vmatrix} X_p \\ X_r \\ X_y \end{vmatrix} \qquad (3.2.7)$$

where,

$$\begin{vmatrix} X_p \\ X_r \\ X_y \end{vmatrix} = T^{-1} \begin{vmatrix} x_p \\ x_r \\ x_y \end{vmatrix} \qquad (3.2.8)$$

The solution to this equation is given by:

$$X_p = Ae^{-\lambda 1 t}$$

$$X_r = Be^{-\lambda 2 t} \qquad (3.2.9)$$

$$X_y = Ce^{-\lambda 3 t}$$

multiplying each side of the equation by T, we obtain:

$$\begin{vmatrix} x_p \\ x_r \\ x_y \end{vmatrix} = T \begin{vmatrix} Ae^{-\lambda 1 t} \\ Be^{-\lambda 2 t} \\ Ce^{-\lambda 3 t} \end{vmatrix} \qquad (3.2.10)$$

The matrix P can now be obtained as the column vectors given by $P = (p_1,p_2,p_3)$. We will find solutions to the state equations given by (3.2.10) constraining p_3 to be a space invariant vector. This agrees with our hypothesis that the yaw axis eigenvector remains approximately spatially invariant with regard to head tilts.

The rotation transformation from spatial to head coordinates is given as $R^{-1} = (q_1,q_2,q_3)$. Therefore,

$$R^{-1}P = \begin{vmatrix} 1 & 0 & * \\ 0 & 1 & * \\ 0 & 0 & * \end{vmatrix} = \begin{vmatrix} 1 & 0 & \\ 0 & 1 & q_3' \\ 0 & 0 & \end{vmatrix} = T \qquad (3.2.11)$$

where, $q_3' = R^{-1}p_3 = (s_1,s_2,s_3)^t$, and where s_1 ,s_2 ,and s_3 are the components of the yaw axis eigenvector in head coordinates.

The simplified form of equation (3.2.11) results from expressing the eigenvectors in head coordinates. The first two

colums of T match that of the identity matrix because the first two eigenvectors rotate with the head.

We can now solve for the state vector:

$$\begin{vmatrix} x_p \\ x_r \\ x_y \end{vmatrix} = \begin{vmatrix} 1 & 0 & s_1 \\ 0 & 1 & s_2 \\ 0 & 0 & s_3 \end{vmatrix} \begin{vmatrix} A\, e^{-\lambda 1 t} \\ B\, e^{-\lambda 2 t} \\ C\, e^{-\lambda 3 t} \end{vmatrix} \qquad (3.2.12)$$

$$= \begin{vmatrix} A\, e^{-\lambda 1 t} + s_1\, C e^{-\lambda 3 t} \\ B\, e^{-\lambda 2 t} + s_2\, C e^{-\lambda 3 t} \\ s_3\, C\, e^{-\lambda 3 t} \end{vmatrix} \qquad (3.2.13)$$

For an initial condition of $(x_1,x_2,x_3)^t$, the coefficients A,B,C are:

$$A = x_1 - s1/s3 * x_3$$

$$B = x_2 - s2/s3 * x_3 \qquad (3.2.14)$$

$$C = x_3/s3$$

Thus, substituting (3.2.14) into (3.2.13) and using the fact that y =x, yields the solutions,

$$y_p = (x1 - s1/s3 * x3) * e^{-\lambda 1 t} + (s1/s3) * x_3 * e^{-\lambda 3 t}$$

$$y_r = (x2 - s2/s3 * x3) * e^{-\lambda 2 t} + (s2/s3) * x_3 * e^{-\lambda 3 t} \qquad (3.2.15)$$

$$y_y = x_3 * e^{-\lambda 3 t}$$

Thus, we have obtained the time domain solution for the three eye velocities in head coordinates as a function of the

eigenvalues, eigenvectors and initial conditions.

We can simplify these expressions by setting,

$$\begin{aligned} c_{11} &= x_1 - (s_1/s_3) * x_3, \\ c_{13} &= (s_1/s_3) * x_3, \\ c_{22} &= x_2 - (s_2/s_3) * x_3, \\ c_{23} &= (s_2/s_3) * x3, \\ c_{33} &= x_3, \end{aligned} \qquad (3.2.16)$$

Then,

$$\begin{aligned} y_p &= c_{11}e^{-\lambda 1t} + c_{13}e^{-\lambda 3t} \\ y_r &= c_{22}e^{-\lambda 2t} + c_{23}e^{-\lambda 3t} \\ y_y &= c_{33}e^{-\lambda 3t} \end{aligned} \qquad (3.2.17)$$

In order to compute the components of the eigenvectors in head coordinates at a specific orientaions of the head with regard to gravity, we form R the rotation matrix in terms of the euler angles (Goldstein, 1954; Appendix). Phi is the angle of rotation about the spatial vertical and therefore is not affected by gravity. That is, the initial direction of the subject does not influence the outcome of a given experiment. Thus, for our purposes it suffices to set phi = 0. Therefore,

$$R = \begin{vmatrix} \cos\psi & -\sin\psi & 0 \\ \cos\theta\sin\psi & \cos\theta\cos\psi & -\sin\theta \\ \sin\theta\sin\psi & \sin\theta\cos\psi & \cos\theta \end{vmatrix} \qquad (3.2.18)$$

The components of the associated yaw axis eigenvector in the subject's frame of reference is the last column of T, $q_3' = (s_1,s_2,s_3)^t$. We now show how to derive the s_i values in terms of

the remaining two euler angles, theta and psi. The associated yaw axis eigenvector, $p_3 = R_3 e_3 = (0,-\sin\theta',\cos\theta')$, where θ' is the angle of rotation of the associated yaw axis eigenvector from the spatial vertical.

Then $T = R^{-1}P =$

$$\begin{vmatrix} 1 & 0 & -\sin\theta'\cos\theta\sin\psi + \cos\theta\sin\theta\sin\psi \\ 0 & 1 & -\cos\theta\cos\psi\sin\theta' + \sin\theta\cos\psi\cos\theta' \\ 0 & 0 & \sin\theta\sin\theta' + \cos\theta\cos\theta' \end{vmatrix} \qquad (3.2.19)$$

Then q_3', in the monkey frame of reference has components:

$$s_1 = -\sin q'\cos\theta\sin\psi + \cos\theta\sin\theta\sin\psi$$

$$s_2 = -\cos\theta\cos\psi\sin\theta' + \sin\theta\cos\psi\cos\theta' \qquad (3.2.20)$$

$$s_3 = \sin\theta\sin\theta' + \cos\theta\cos\theta'$$

We now confirm that the closed form solution is correct, by showing that $\underline{x}' = H_s\underline{x}$:

Following from (3.2.1) we express the system matrix in head coordinates in terms of its eigenvalues and eigenvectors:

$$H_s = \begin{vmatrix} 1 & 0 & s1 \\ 0 & 1 & s2 \\ 0 & 0 & s3 \end{vmatrix} \begin{vmatrix} -\lambda 1 & 0 & 0 \\ 0 & -\lambda 2 & 0 \\ 0 & 0 & -\lambda 3 \end{vmatrix} \begin{vmatrix} 1 & 0 & -s1/s3 \\ 0 & 1 & -s2/s3 \\ 0 & 0 & 1/s3 \end{vmatrix}$$

$$= \begin{vmatrix} -\lambda 1 & 0 & s1/s3(\lambda 1-\lambda 3) \\ 0 & -\lambda 2 & s2/s3(\lambda 2-\lambda 3) \\ 0 & 0 & -\lambda 3 \end{vmatrix} \qquad (3.2.22)$$

The differentiated state vector is x':

$$\begin{vmatrix} x_p' \\ x_r' \\ x_y' \end{vmatrix} = \begin{vmatrix} -\lambda 1(x1 - s1/s3*x3)e^{-\lambda 1t} - \lambda 3*s1/s3*x3\ e^{-\lambda 3t} \\ -\lambda 2(x2 - s2/s3*x3)e^{-\lambda 2t} - \lambda 3*s2/s3*x3\ e^{-\lambda 3t} \\ -\lambda 3x3e^{-\lambda 3t} \end{vmatrix} \qquad (3.2.23)$$

We now multiply the system matrix H_s by the state vector x getting $H_s\underline{x}$:

$$\begin{vmatrix} -\lambda 1(s1/s3*x3-x1)e^{-\lambda 1t} - \lambda 1*s1/s3*x3*e^{-\lambda 3t} + s1/s3(\lambda 1-\lambda 3)x3*e^{-\lambda 3t} \\ -\lambda 2(s2/s3*x3-x2)e^{-\lambda 2t} - \lambda 2*s2/s3*x3*e^{-\lambda 3t} + s2/s3(\lambda 2-\lambda 3)x3*e^{-\lambda 3t} \\ -\lambda 3e^{-\lambda 3t} \end{vmatrix}$$

(3.2.24)

This simplifies to (3.2.23), confirming that $\underline{x}' = H_s\underline{x}$.

We next consider how the eigenvalues and eigenvectors can be determined from experiments and how these results compare to data.

3.3 Identification of Eigenvalues and Eigenvectors from Three Dimensional Eye Velocity: Least Squares Fit

As shown above, we have derived the closed form solutions for the velocity vector output of our model in head coordinates. During OKAN the output y in head coordinates is equal to the state vector. Therefore the solution to the state equations for each component of eye velocity is a linear combination of exponentials having the following form:

$$\underline{y} = \begin{vmatrix} y_p \\ y_r \\ y_y \end{vmatrix} = \begin{vmatrix} c_{11}e^{-\lambda 1 t} + c_{12}e^{-\lambda 2 t} + c_{13}e^{-\lambda 3 t} \\ c_{21}e^{-\lambda 1 t} + c_{22}e^{-\lambda 2 t} + c_{23}e^{-\lambda 3 t} \\ c_{31}e^{-\lambda 1 t} + c_{32}e^{-\lambda 2 t} + c_{33}e^{-\lambda 3 t} \end{vmatrix}$$

$$= \begin{vmatrix} c_{11} \\ c_{21} \\ c_{31} \end{vmatrix} e^{-\lambda 1 t} + \begin{vmatrix} c_{12} \\ c_{22} \\ c_{32} \end{vmatrix} e^{-\lambda 2 t} + \begin{vmatrix} c_{13} \\ c_{23} \\ c_{33} \end{vmatrix} e^{-\lambda 3 t} \qquad (3.3.1)$$

where y_p, y_r, and y_y are the pitch, roll and yaw components of eye velocity during OKAN.

The vectors,

$$\begin{vmatrix} c_{11} \\ c_{21} \\ c_{31} \end{vmatrix}, \begin{vmatrix} c_{12} \\ c_{22} \\ c_{32} \end{vmatrix}, \text{ and } \begin{vmatrix} c_{13} \\ c_{23} \\ c_{33} \end{vmatrix} \qquad (3.3.2)$$

are the eigenvectors of the system matrix, H, and the λi are the corresponding eigenvalues. Therefore, the solution is a nonlinear vector function of the eigenvalues and eigenvectors. To identify the eigenvalues and eigenvectors of the system matrix, a nonlinear identification technique must be used.

An efficient algorithm using a least squares minimization of the error between the data and the model output was developed by Marquardt (1963). By extending the Marquardt algorithm to use three dimensional input vectors, we could identify the system matrix from data from monkeys. In the next section, we consider the Marquardt algorithm, its implementation, and extension to three dimensional data modelling.

3.3.1 Basis for Marquardt Algorithm for estimation of parameters

The Marquardt algorithm combines two common methods for least squares estimation of nonlinear parameters. The first method makes use of a Taylor expansion of the function to be fit which takes advantage of local linearity. The other method is a variation of a steepest descent method. Each of these methods has its limitations. The Taylor expansion tends to diverge on succesive iterations and the region of convergence is small. This implies that the parameters must be fairly close to the final values before this technique is useful. In the problem of identifying the system matrix components, this is not feasible. The steepest descent technique, on the other hand, has a greater region of convergence, but has an long convergence rate. The Marquardt method varies smoothly between these two methods in such a way as to avoid the pitfalls of each while utilizing their advantages.

In deriving the algorithm we will assume that the function to fit the data is given by:

$$y = f(\underline{x},\beta) = f(x_1,x_2,\ldots,x_m;\beta_1,\beta_2,\ldots,\beta_k) \qquad (3.3.3)$$

where xj are the independent variables and βi are the parameters to be calculated. We define Z_i to be the value of the function at the ith data point.

Yi are the data points at the points $\underline{X}_i = X_{1i},X_{2i},\ldots,X_{mi}$ giving the tuples:

$$(Y_i,X_{1i},X_{2i},\ldots,X_{mi}) \quad i=1,2,\ldots,n \qquad (3.3.4)$$

The goal is to find parameters $\beta_i, \ldots, \beta_k$ which minimize the sum of the squared errors given by:

$$\Phi(\beta 1, \beta 2, \ldots, \beta k) = \Sigma_{i=1,n} (Y_i - Z_i)^2 \qquad (3.3.5)$$

As is well known, a necessary condition for finding the minimum is that the gradient of Φ vanishes. This will be satisfied when:

$$\partial\Phi/\partial\beta_j = 0 \qquad i \leq j \leq k \qquad (3.3.6)$$

In general, it is impossible to find a closed form solution to the above system (3.3.6). However, in the case where f is linear, the set of equations obtained from (3.3.6) become linear, and posess a closed form solution. One way to utilize the linearizing property is to constrain the parameters within some k-dimensional sphere.

The constrained problem is to minimize $\Phi(\beta 1, \ldots, \beta k)$ subject to the condition that

$$g(\beta 1, \ldots, \beta k) = ||\beta - \beta^o|| = r \qquad (3.3.7)$$

for some fixed $r > 0$. $||\cdot||$ is the norm of the vector and β^o is the initial value of the vector β.

If such a minimum occurs at a point β', then the pair of level surfaces $g(\beta) = r$ and $\Phi(\beta) = \Phi(\beta')$ are tangent to one another at the point β'. Therefore their respective gradients must be proportional, i.e., $\Phi(\beta') = \mu \nabla g(\beta')$ for some real number μ.

Thus we see that in order to find the minima of the constrained system, it suffices to solve the following system of equations:

$$\nabla\Phi = \mu\nabla g \quad \text{and}$$

$$g(\beta) = r \tag{3.3.8}$$

This is known as the method of Lagrange multipliers (Kaplan & Lewis, 1971).

Again, (3.3.8) is, in general, quite complicated. A key feature of the Marquardt algorithm is the observation that (3.3.8) becomes linear if f is linear. If f is linear it can be expresses as:

$$f(x_1,\ldots,x_m;b_1,\ldots,b_{k+1}) = \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \beta_{k+1} \tag{3.3.9}$$

Φ is therefore given by:

$$\Phi = \Sigma_{i=1,n} \left(Y_i - \Sigma_{k=1,m} \beta_k X_{ki} - \beta_{k+1}\right)^2 \tag{3.3.10}$$

Taking the partial derivatives with respect to the parameters βj and setting it equal to zero we see that (3.3.6) becomes

$$\Sigma_{i=1,n}\, 2\, (Y_i - \Sigma_{k=1,m}\, \beta_k\, X_{ki} - \beta_{k+1}) X_{ji} = 0 \quad j \leq m \tag{3.3.11a}$$

$$-\Sigma_{i=1,n}\, 2\, (Y_i - \Sigma_{k=1,m}\, \beta_k\, X_{ki} - \beta_{k+1}) = 0 \quad j = m+1 \tag{3.3.11b}$$

Thus, as mentioned earlier, we see that these equations are linear in βi and hence can be easily solved via Gaussian elimination.

The above procedure can be interpreted geometrically as follows: When $m = 1$ (one parameter) then the level curves, $\Phi = c$, yield a family of ellipses. Minimizing Φ with regard to b is equivalent to finding the point on a level ellipse for which the c parameter has the smallest possible value. Therefore if f is linearized a solution for $\underline{\beta}$ is obtained directly and does not require an iterative procedure.

Since the local behavior of a smooth function is approximately linear, one can apply this method of "constrained least squares" to minimize Φ in spheres of radius r, and obtain solutions for the parameters if r is chosen appropriately.

3.3.2 Development of Algorithm for Parameter Identification

We now consider how linearization and the gradient techniques are combined in the implementation of Marquardt's algorithm.

In the linearization technique if the function f is expanded in a Taylor series about a point in parameter space $\underline{\beta} = b$, we obtain a linear approximation of f given by:

$$f_i(b+\delta) \sim f_i(b) + \Sigma_{j=1,k} (\partial fi/\partial \beta j) \mid b \ (\delta j) \qquad (3.3.12)$$

where $f_i(b) = f(\underline{X}_i, b)$ $i = 1,..,n$ and $\delta = (\delta 1,...,\delta k)$

The vector δ represents an offset in the parameter vector b which gives a better approximation to the data. Therefore the procedure to find the correct δ is to initially make a guess of the solution, b and improve this guess by changing b to $b+\delta$. We now show how this is done using a standard least squares method. The accuracy with which δ is determined is dependent on the nonlinear characteristics of the function f.

Rewriting (3.3.12) in matrix notation we obtain:

$$\langle Y \rangle = f_o + P\delta \qquad (3.3.13)$$

where, $\langle Y \rangle$ is the linearized function about $\beta = b$:

$$\begin{vmatrix} \langle Y(X_1, b+\delta) \rangle \\ \langle Y(X_2, b+\delta) \rangle \\ \cdots \\ \cdots \\ \langle Y(X_n, b+\delta) \rangle \end{vmatrix} \qquad (3.3.14)$$

and,

$$f_o = \begin{vmatrix} f_1(b) \\ \cdots \\ f_n(b) \end{vmatrix} \qquad (3.3.15)$$

and $P = \partial f_i/\partial b_j \quad i = 1,..,n \quad j = 1,..,k$

Note that P is an n x k matrix:

$$\begin{vmatrix} \frac{\partial f_1}{\partial b_1} & \frac{\partial f_1}{\partial b_2} \cdots & & \frac{\partial f_1}{\partial b_k} \\ & \cdots & & \\ & \cdots & & \\ \frac{\partial f_n}{\partial b_1} & \frac{\partial f_n}{\partial b_2} \cdots & & \frac{\partial f_m}{\partial b_k} \end{vmatrix} \qquad (3.3.16)$$

The criterion function is therefore given as:

$$\Phi = \Sigma_{i=1,n} (Y_i - \langle Y_i \rangle)^2 \qquad (3.3.17)$$

Taking the partial derivatives of Φ with respect to the δ_j, and setting them equal to zero, we obtain:

$$\partial\Phi/\partial\delta_j = -2 \Sigma_{i=1,n} (Y_i - \langle Y_i \rangle) * \partial f_i/\partial\beta_j = 0 \qquad (3.3.18)$$

where from equation (3.3.13),

$$\langle Y_i \rangle = f_i + \Sigma_{l=1,k} \partial f_i/\partial b_l * \delta_l \qquad (3.3.19)$$

Substituting (3.3.19) into (3.3.18) we have,

$$\Sigma_{i=1,n} [Y_i - f_i - (\Sigma_{l=1,k} \partial f_i/\partial b_l * \delta_l)] * \partial f_i/\partial\beta_j = 0 \qquad (3.3.20)$$

Rewriting (3.3.20),

$$\Sigma i=1,n\ (Yi - fi)^* \partial fi/\partial \beta j =$$

$$\Sigma i=1,n \Sigma l=1,k\ \partial fi/\partial bl\ ^*\ \partial fi/\partial \beta j\ ^*\ \delta l \qquad (3.3.21)$$

By switching the two summations (in 3.3.21), we get,

$$\Sigma l=1,k(\Sigma i=1,n\ \partial fi/\partial \beta l^*\ \partial fi/\partial \beta j)\ ^*\ \delta l \qquad (3.3.22)$$

By setting

$$gi\ =\Sigma i=1,n\ (Yi - fi)^*\ \partial fi/\partial \beta j \qquad (3.3.23)$$

in equation (3.3.21) and noting that,

$$P^TP\ =\Sigma i=1,n\ \partial fi/\partial \beta l\ ^*\ \partial fi/\partial \beta j \qquad (3.3.24)$$

equation (3.3.21) reduces to the matrix equivalent:

$$P^TP\ \delta\ =\ g \qquad (3.3.25)$$

where $g = (g_1,...,g_k)$.

By evaluating (3.3.18) at $\delta = 0$ (that is, when $\beta = b$) we see that $g = (-1/2)\ \nabla\Phi\ (b)$, the gradient of the criterion function.

We can further simplify equation (3.3.25) by letting $A = P^TP$. We thus obtain the linear system,

$$A\ \delta\ =\ g \qquad (3.3.26)$$

The linearization technique reduces to solving a system of linear equations for δ, where A and g are obtained from the non-linear function that is to fit the data. This applies when the parameters are close to a solution.

This linearization method will produce a meaningful value for δ only if f is approximately globally linear. Since the generic function is only locally linear, we are led to the constrained least squares problem: Here we try to find a δ which minimizes the function Φ defined by (3.3.17) subject to the constraint

$$|| \delta || = r \qquad (3.3.27)$$

Here r is chosen sufficiently small, so as to guarantee that the linear approximation given by (3.3.13) is valid. Marquardt shows that the Lagrange multiplier method leads to the following linear system:

$$(A + \lambda I)\, \delta = g \qquad (3.3.28)$$

where A and g are as before, and λ is some real number which depends on r and I is the identity matrix. Furthermore, he shows that λ and r are inversely related: λ is a decreasing function of r. It approaches zero as r approaches infinity, and approaches infinity as r approaches zero.

The other aspect of Marquardt's method is to use the gradient technique when the parameters are far from the solution. This method will produce a δ which is guaranteed to approach a local minimum. Here we take the partial derivatives

of Φ with respect to each of the parameters.

$$\Phi = \Phi (b_1,\ldots,b_k)$$

$$\nabla\Phi = (\partial\Phi/\partial\beta_1,\ldots,\partial\Phi/\partial\beta_k)\ (b) \qquad (3.3.29)$$

In order to decrease Φ we move in the direction of the negative gradient.

The Marquardt method dynamically combines the Taylor series method with the gradient technique. As λ increases, δ decreases. When λ is relatively small, δ is large and the Taylor series method is implemented with big steps towards minimizing Φ. When λ is large, δ is small and small steps are taken towards minimizing the criterion function, Φ. Note that as λ decreases, equation (3.3.28) approaches the linear least squares method (3.3.26). In contrast, as λ increases, the matrix A in equation (3.3.28) becomes diagonally dominant and the equation reduces to $(\lambda I)\delta = g$ or $\delta = (1/\lambda)g = (-1/2\lambda)^* \nabla\Phi$. This is equivalent to the gradient descent method. The algorithm can be summarized as follows:

Given an initial parameter vector b; and some arbitrarily small ε:

Compute the criterion function $\Phi^{(1)}(b)$

REPEAT

Solve $(A + \lambda I)\,\delta = g$ for δ and

set $\Phi^{(i+1)} = \Phi(b + \delta)$

If $\Phi^{(i)} < \Phi^{(i+1)}$

$\lambda := \lambda * 10$

else { $\lambda := \lambda * .1$

$b := b + \delta$ }

$i := i + 1$

UNTIL $\Phi^{(i)} - \Phi^{(i+1)} < \varepsilon$

The Marquardt algorithm is not globally convergent. That is, if the initial parameters are far from a minimum, the process might diverge. Variations of this method have been introduced which are globally convergent (Moré, 1977; Powell, 1975; Osborne, 1976). In order to ensure convergence, they employ variations of a line search technique where when a step is taken in the direction of the negative gradient, a convergence test is performed. If the step is found to be unacceptable, the step is shortened by some factor and the process continues. This guarantee of convergence is advantageous if one is concerned with finding any minimum. However, it still does not address the problem of arriving at an absolute minimum. If the parameters are chosen sufficiently close to the solution, this method is adequate. Thus, finding good initial parameters is crucial to convergence to accurate solutions. How to accomplish this is related to the specific problem and is discussed in section 3.4.1 The next section describes an extension of the Marquardt method to vector valued functions, necessary for our purposes as we are fitting data corresponding to a three dimensional model.

3.3.3 Extension of Marquardt Algorithm to Vector-Valued Functions

Each component of the velocity function has parameters in common with the others (see equation 3.3.1). Rather than treat these functions separately and produce several sets of parameters, we extend the Marquardt method to handle vector-valued functions.

Let the model to be parameterized be given by the vector-valued function:

$$\underline{y} = \underline{f}(\underline{x},\underline{\beta}) = \begin{vmatrix} f^{(1)}(x_1,x_2,\ldots,x_m;\beta_1,\beta_2,\ldots,\beta_k) \\ f^{(2)}(x_1,x_2,\ldots,x_m;\beta_1,\beta_2,\ldots,\beta_k) \\ \cdots \\ f^{(l)}(x_1,x_2,\ldots,x_m;\beta_1,\beta_2,\ldots,\beta_k) \end{vmatrix} \qquad (3.3.30)$$

where xi are the independent variables and βi are the parameters to be calculated. The data points will be denoted:

$$(\underline{Y}_i, X_{1i}, X_{2i}, \ldots, X_{mi}) \quad i = 1,2,\ldots n \qquad (3.3.31)$$

The problem is to compute the βi which minimize:

$$\Phi = \Sigma \, |\underline{Y}_i - \underline{Z}_i|^2 \qquad (3.3.32)$$

where $\underline{Z}_i$ is the value of y at the ith point. Taking the norm of the differences we obtain:

$$\Phi = \Sigma_{i=1,n} \left| \begin{vmatrix} Y_i^{(1)} \\ Y_i^{(2)} \\ \cdots \\ Y_i^{(l)} \end{vmatrix} - \begin{vmatrix} Z_i^{(1)} \\ Z_i^{(2)} \\ \cdots \\ Z_i^{(l)} \end{vmatrix} \right|^2 \qquad (3.3.33)$$

This involves taking the square root of the squares of the differences. Since the entire quantity is squared, the square root is cancelled. This simplifies to:

$$\Phi = \Sigma i=1,n\ (Yi^{(1)}-Zi^{(1)})^2 + (Yi^{(2)}-Zi^{(2)})^2 + \ldots + (Yi^{(l)}-Zi^{(l)})^2$$
$$= \Sigma 1=1,n \Sigma j=1,l (Yi\ (j)-Zi\ (j))^2 \qquad (3.3.34)$$

This Φ is dependent on all the functions contributing to the sum. Following the notation from (3.3.13), the extended Taylor series expansion through the linear terms is:

$$\langle \underline{Y} \rangle = f_o + P\delta \qquad (3.3.35)$$

where f_o (which has 1*n components) =

$$\begin{vmatrix} f^{(1)}(xi^{(1)},b) \\ f^{(2)}(xi^{(2)},b) \\ \cdots \\ f^{(l)}(xi^{(l)},b) \end{vmatrix} \qquad (3.3.36)$$

and the augmented P is:

$$P = \begin{vmatrix} \frac{\partial f_1^{(1)}}{\partial b_1} \frac{\partial f_1^{(1)}}{\partial b_2} \ldots & & \frac{\partial f_1^{(1)}}{\partial b_k} \\ \frac{\partial f_2^{(1)}}{\partial b_1} \frac{\partial f_2^{(1)}}{\partial b_2} \ldots & & \frac{\partial f_1^{(1)}}{\partial b_k} \\ & \cdots & \\ & \cdots & \\ \frac{\partial f_n^{(1)}}{\partial b_1} \frac{\partial f_n^{(1)}}{\partial b_2} \ldots & & \frac{\partial f_n^{(1)}}{\partial b_k} \\ \frac{\partial f_1^{(2)}}{\partial b_1} \frac{\partial f_1^{(2)}}{\partial b_2} \ldots & & \frac{\partial f_1^{(2)}}{\partial b_k} \\ & \cdots & \\ & \cdots & \\ & \cdots & \\ \frac{\partial f_n^{(l)}}{\partial b_1} \frac{\partial f_n^{(l)}}{\partial b_2} \ldots & & \frac{\partial f_n^{(l)}}{\partial b_k} \end{vmatrix} \qquad (3.3.37)$$

The coupling among the various $f^{(j)}$ occurs when $A = P^T P$ is formed (see equation 3.3.24). If however the $f^{(j)}$ have no parameters in common, then P will be a block diagonal matrix and no coupling will result. Once the matrix P is found, the matrix A is determined and equation (3.3.28) can be implemented to search for the parameters as described above.

We will next apply the extended Marquardt algorithm in computing the eigenvectors and eigenvalues of the system matrix associated with velocity storage.

3.4 General Procedure for Finding the Eigenvalues and Eigenvectors from Data for Tilt Experiments

We now show how to extract the eigenvalues and the eigenvectors of the system matrix from pitch, roll, and yaw axis eye velocity data during OKAN. To do so, we must estimate the corresponding parameters as initial conditions for the extended Marquardt algorithm. We must also determine the matrix transformation that maintains the yaw axis close to the vertical in head coordinates.

We can then use the closed form solutions given above (Equation 3.2.17), which are given in head coordinates, to determine the c_{ij} and the λ_i from the data.

For the data under consideration, the subject is tilted about the animal's y axis emanating from the back of the head (see Figure 2.9). This axis then remains unchanged. Under this condition there is cross coupling between the yaw and pitch axis but not to the roll axis (see Figure 2.9a).

The Euler angles that determine the orientation of the subject for a tilted position are theta and psi. Psi is 90 degrees and theta determines the angle of tilt. This constraint together with the fact that phi can be set to zero (see section 3.2.1 and equation (3.2.18)), simplifies the rotation matrix R to:

$$R = \begin{vmatrix} 0 & -1 & 0 \\ \cos\theta & 0 & -\sin\theta \\ \sin\theta & 0 & \cos\theta \end{vmatrix} \qquad (3.4.1)$$

If we apply the inverse rotation to the yaw axis eigenvec-

tor in spatial coordinates, p_3, we obtain its represention in head coordinates, q_3'.

Therefore, $R^{-1}p3 = q3' =$

$$\begin{vmatrix} 0 & \cos\theta & \sin\theta \\ -1 & 0 & 0 \\ 0 & -\sin\theta & \cos\theta \end{vmatrix} \quad \begin{vmatrix} 0 \\ -\sin\theta' \\ \cos\theta' \end{vmatrix}$$

$$= \begin{vmatrix} -\sin\theta'\cos\theta + \cos\theta'\sin\theta \\ 0 \\ \sin\theta'\sin\theta + \cos\theta'\cos\theta \end{vmatrix} = \begin{vmatrix} \sin(\theta-\theta') \\ 0 \\ \cos(\theta-\theta') \end{vmatrix} \quad (3.4.2)$$

where θ is the angle of tilt and θ' is the angle between the spatial vertical and the vertical eigenvector (Figure 3.4.1).

Therefore, the vertical eigenvector $q_3' = (s_1,s_2,s_3)$ has components:

$$s1 = \sin(\theta-\theta'); \; s2 = 0; \; s3 = \cos(\theta-\theta') \quad (3.4.3)$$

Alternatively, we could express q3' in terms of α, the angle between the vertical eigenvector and the monkey's x axis. (see figure 3.4.1) This is computationally more consistent since our computations are relative to head coordinates. Then,

$$\alpha = 90 - (\theta-\theta') \quad (3.4.4)$$

and,

$$\cos\alpha = \sin(\theta-\theta'), \; \sin\alpha = \cos(\theta-\theta') \quad (3.4.5)$$

then,

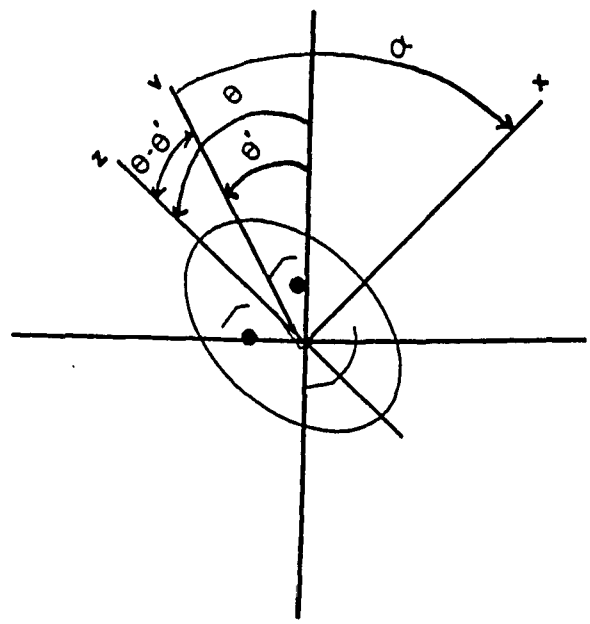


Figure 3.4.1 Yaw axis eigenvector relative to animal head.

v: yaw axis eigenvector

θ: angle of animal tilt from spatial vertical

θ': angle between eigenvector and spatial vertical

α: angle between eigenvector and animal x-axis

$$q_3' = (\cos \alpha, 0, \sin \alpha)^t \qquad (3.4.6)$$

Given the angle of tilt of the animal and the angle of the vertical eigenvector from the spatial vertical, we now have the means to compute the components s_1 and s_3 of q_3'. With this we can calculate c_{11} and c_{13} of the closed form solutions (equation 3.2.17) which are functions of s_1, s_2, s_3 and the initial values of the pitch, roll, and yaw eye velocities during OKAN.

The experimental data provides us with the initial values of velocity for pitch, roll and yaw: x_{po}, x_{ro} and x_{yo}. Therefore by making an initial approximation to the eigenvectors, we have initial vlaues for the c_{ij} that determine the solutions for the velocity profiles. In addition, if the eigenvalues are initialized, experimental velocity profiles can be compared to the functions that predict these velocities and the extended Marquardt algorithm can be used to identify the eigenvalues and eigenvectors for this data set.

For any tilt, the "pitch" eigenvector does not change relative to the monkey's head and is therefore still $[1,0,0]^t$ in head coordinates. That is,

$$\begin{vmatrix} c_{11} \\ c_{21} \\ c_{31} \end{vmatrix} = \gamma \begin{vmatrix} 1 \\ 0 \\ 0 \end{vmatrix} \qquad (3.4.7)$$

where γ is determined by the initial velocities and the tilt of the yaw axis eigenvector.

The "roll" eigenvector remains unchanged relative to gravity and is:

$$\begin{vmatrix} c_{12} \\ c_{22} \\ c_{32} \end{vmatrix} = \begin{vmatrix} 0 \\ 1 \\ 0 \end{vmatrix} \qquad (3.4.8)$$

A flow diagram that shows the process of identifying the eigenvalues and eigenvectors is shown in figure (3.4.2).

We next consider how to estimate the initial parameter values for our problem that drive the Marquardt algorithm.

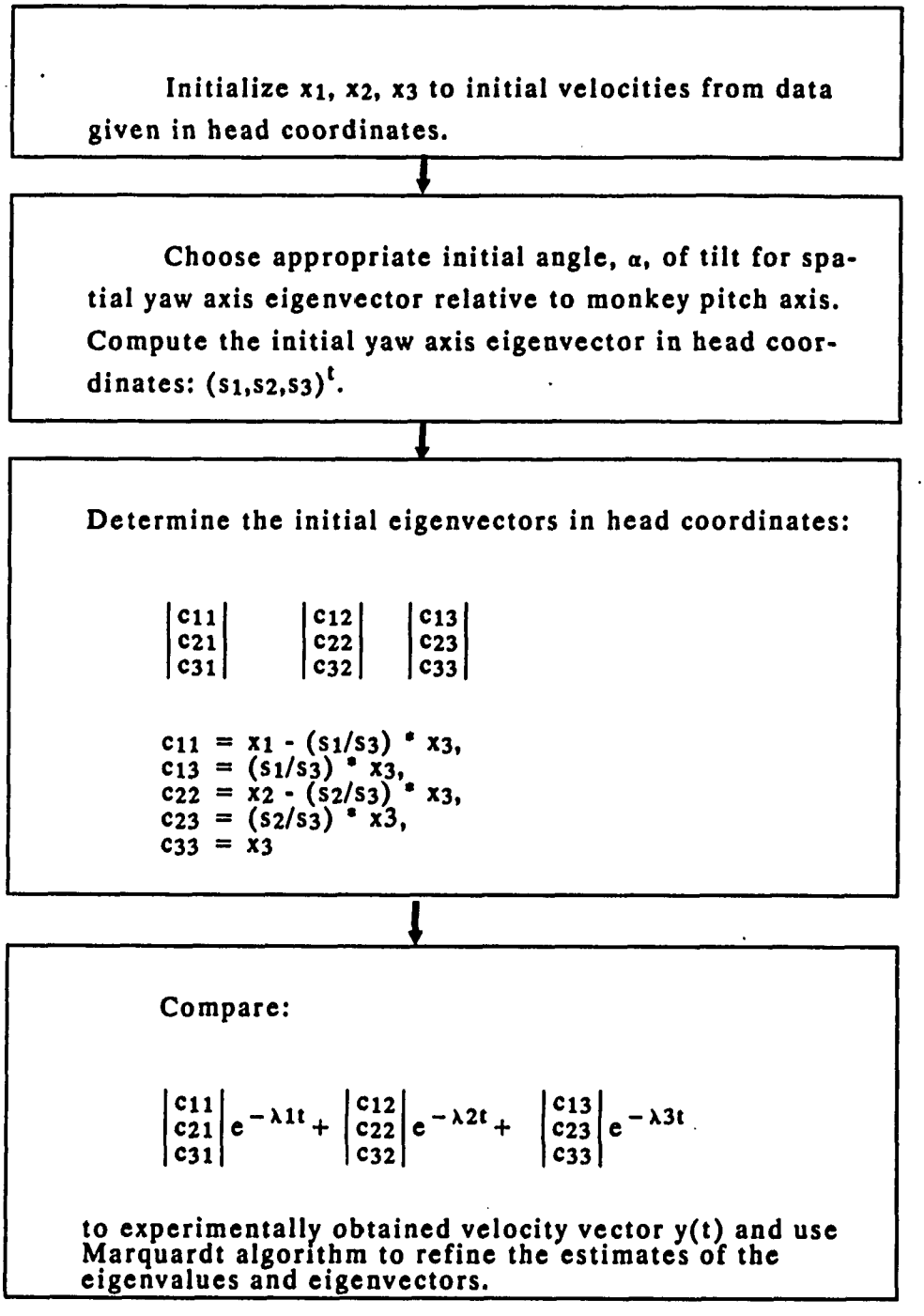


Figure 3.4.2 Flow diagram for identifying the eigenvalues and eigenvectors.

3.4.1 Estimation of the initial parameters

By examining the flow diagram in figure (3.4.2), it is apparent that the choice of initial parameters is important in the identification procedure. Specifically, it is important to choose the initial conditions correctly. Since the extended Marquardt method does not guarantee finding an absolute minimum, it may converge to any local minimum. Therefore, in order to find the best fit we must start with reasonable initial values.

For tilt experiments, the output velocities contain pitch and yaw components (roll is not present and is set to zero). The yaw component is modelled by a single exponential with coefficient c_{33} and exponent λ_3 since there appears to be no cross- coupling from the pitch to the yaw axis. Therefore, c_{31} is set to zero and there is no mode of the response with λ_1 as its eigenvalue. The coefficient c_{33} is set to the initial value for yaw velocity, x_{yo}. The pitch component is the result of cross-coupling between the horizontal and vertical velocities, resulting in the response being the sum of two exponentials. Therefore, it has c_{11} and c_{13} as coefficients and λ_1 and λ_3 as the corresponding eigenvalues in the response modes. Thus, the exponent, λ_3 is the one parameter common to the pitch and yaw velocities.

The strategy for finding an initial set of parameters, is to first fit the yaw component to obtain estimates for c_{33} and λ_3. If pure vertical data is available, we can fit the pitch component for the same tilt angle. This gives us an upper bound

for λ_1. Next, to find c_{13}, we must have a good approximation for the vertical eigenvector. Based on data taken at a zero degree tilt and a 90 degree tilt it was qualitatively estimated that the yaw axis eigenvector ranged from 0 degrees to 12 degrees. Via linear interpolation, we can approximate an initial value for the vertical eigenvector expressed as $q_3' = (c_{13} \;\; 0 \;\; c_{33})^t$. We have shown above how to calculate c_{13} in terms of the yaw initial value and the angle the vertical eigenvector makes with the animal's pitch eigenvector (see Figure 3.4.2). This gives an initial value for c_{13}. Then $c_{11} = x_{yo} - c_{13}$. By fixing the eigenvalues and the yaw coefficients to their input values, we apply the extended Marquardt method to obtain more accurate values for the pitch coefficients.

In the next section, we discuss the program to implement the procedure for identifying the eigenvalues and eigenvectors of the system matrix associated with velocity storage.

3.5 Computer Implementation of extended Marquardt method:

The flow diagram of the program to identify the parameters of the system is shown in figure (3.4.2). The program was implemented in Microsoft Fortran on a PC.

The input consists of two files, one containing the velocity profiles for pitch roll and yaw and the other a parameter file. In the parameter file the user specifies the angle of tilt of the animal, an initial angle of the yaw axis eigenvector relative to the spatial vertical, and the pitch and yaw time constants. The user also provides a binary attribute list corresponding to the parameters, where a one indicates that the parameter is to be changed and a zero signifies that the parameter is to be held at its input value.

The program converts the input parameters into the corresponding eigenvectors and eigenvalues and transforms the yaw axis eigenvector into a head based basis. Using the attribute list, a permutation vector is formed with the indices of the parameters to be adjusted followed by the those that are held constant.

Based on the input parameters and attribute list, the parameters of the Marquardt algorithm are set and arranged into a vector. The order within this vector does not matter. The input data corresponding to the three functions (for pitch, roll, and yaw) is arranged into a single vector, y, as though it were generated by a single function. The values of the independent variables x_i corresponding to each function (in our case, time) are each shifted to a distinct domain. The user

defined procedure of the Marquardt algorithm is provided to return the value of y evaluated at x_i. The procedure distinguishes among the components of the velocity vector and computes the range appropriately. The partial derivatives of the functions relative to the parameters are evaluated and the P matrix is formed (see section 3.3.2).

The output of the program is the set of converged values of the angle of the yaw axis eigenvector and the pitch and yaw time constants. The data points corresponding to the vector function evaluated at the converged parameters are sent to a file specified by the user, in a format that is easily read by a graphics routine.

The flowcharts describing this process follow (Figures 3.5.1 and 3.5.2).

In the next section we show how the program was utilized in obtaining fits to data from monkeys.

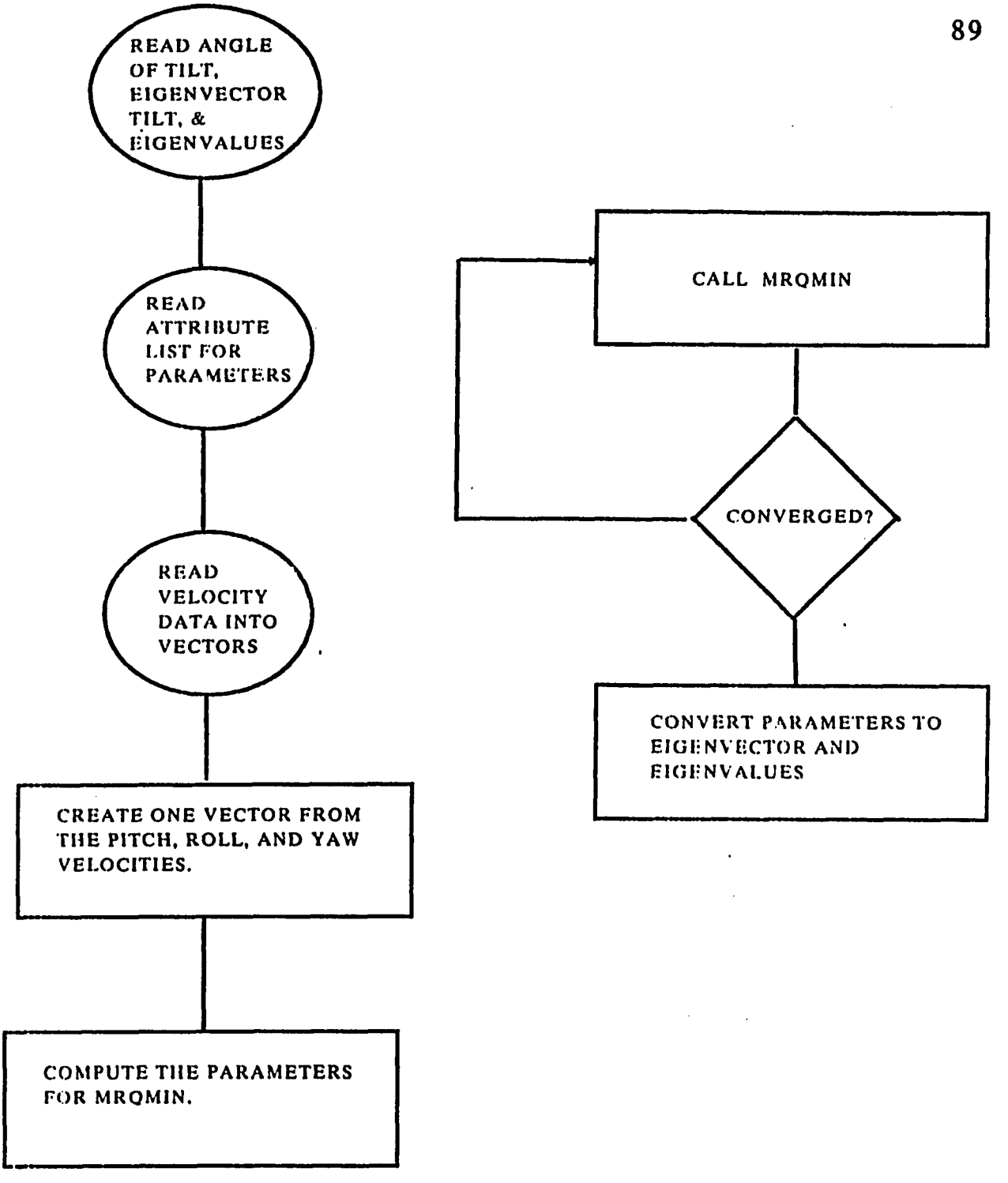


Figure 3.5.1 Flowchart of main routine of extended Marquart algorithm.

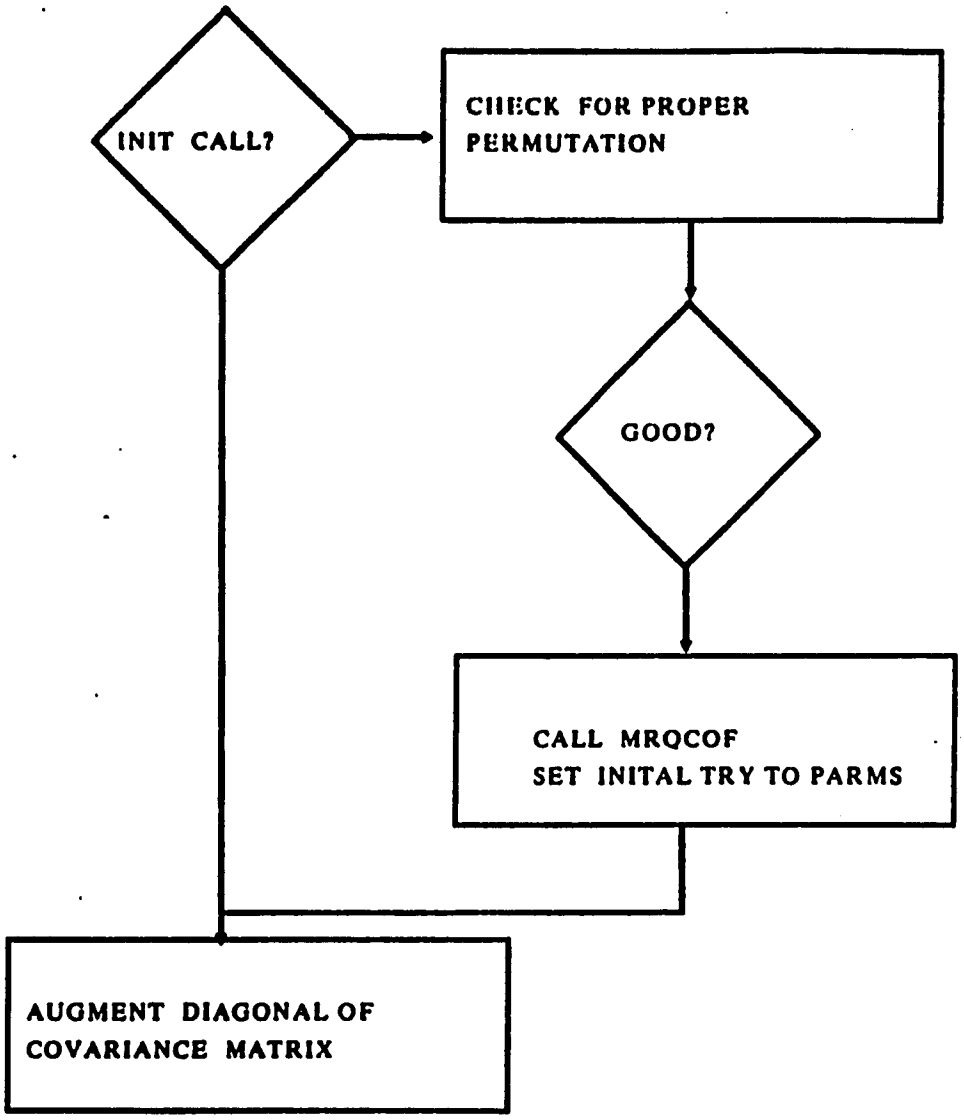
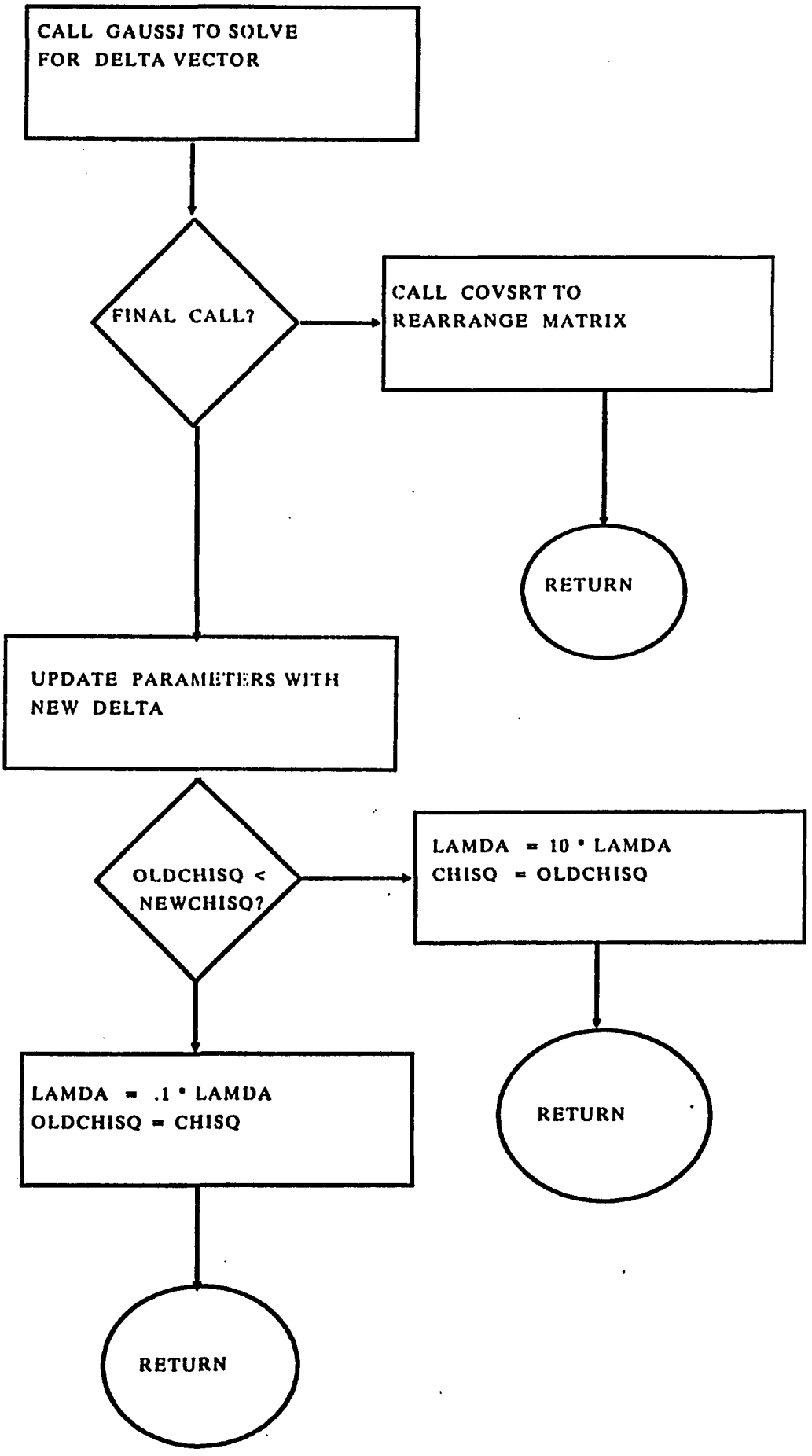


Figure 3.5.2 Flowchart of MRQMIN of extended Marquart algorithm.


CALL GAUSSJ TO SOLVE FOR DELTA VECTOR
FINAL CALL?
CALL COVSRT TO REARRANGE MATRIX
RETURN
UPDATE PARAMETERS WITH NEW DELTA
OLDCHISQ < NEWCHISQ?
LAMDA = 10 * LAMDA
CHISQ = OLDCHISQ
RETURN
LAMDA = .1 * LAMDA
OLDCHISQ = CHISQ
RETURN

3.6 Application to Finding Parameters from Experimental Data

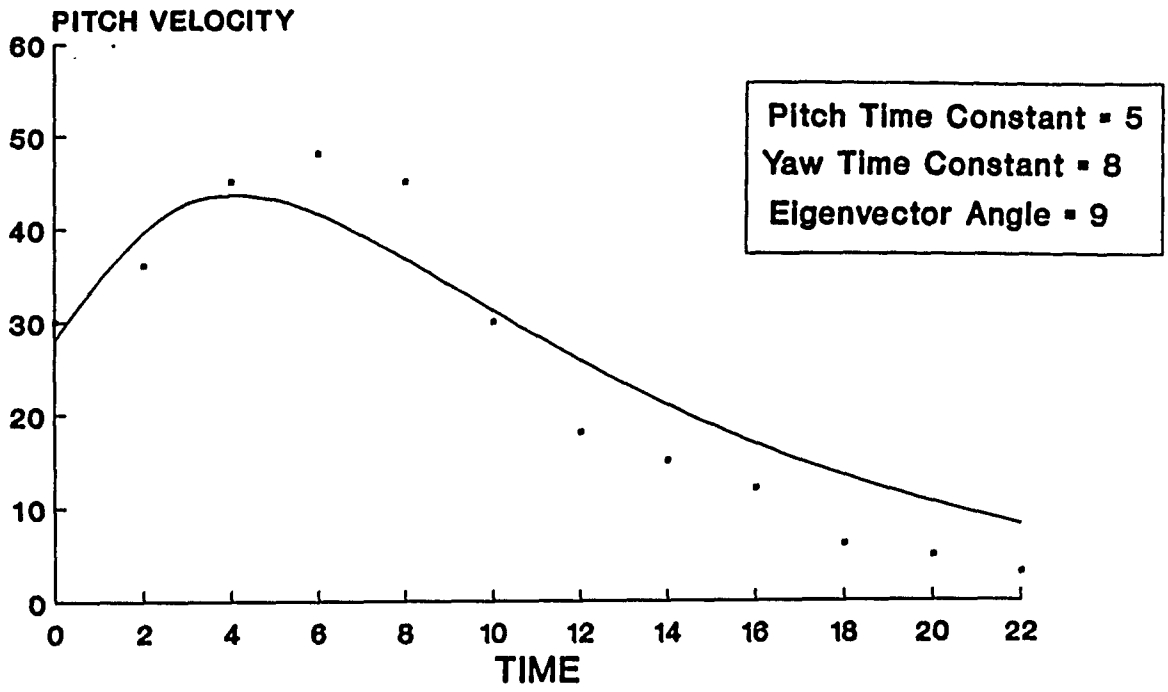
In order to show the applicability of the technique, the identification algorithm was applied to a small data set from experiments done on one monkey (Raphan & Cohen, and personal communication). The animal was right side down and only cross-coupling to the upward pitch component was considered. Despite the small data set, the method developed is shown to be a useful tool in relating the three dimensional model of the vestibulo-ocular reflex to the data. It supports the characterization of the three dimensional velocity storage integrator in terms of its eigenvalues and eigenvectors as a function of gravity. Behavior experiments on cross-coupling to the downward pitch component as well as cross-coupling to roll should further ellucidate the model structure.

When the animal is tilted from the spatial vertical and given a yaw axis optokinetic stimulus (that is the drum is also tilted by the same angle), cross-coupling is produced which generates yaw and pitch components of OKAN (Figure 2.9). The data samples needed for the Marquardt algorithm were obtained by sampling the slow phase eye velocity profiles associated with the yaw and pitch axis of OKAN every 2 seconds. These samples were stored in a file and were later read by the program. The data values are shown as dots in figures (3.6.1, 3.6.3 a-g).

As the first part of the procedure, we obtained the time constant associated with the yaw component. This was accomplished by running the one dimensional Marquardt algo-

Figure 3.6.1 Cross-coupled pitch and yaw OKAN for a 90 degree tilt. The data points are shown as dots. The solid lines are the fits found using the method outlined above. The pitch time constant was 5 seconds and the yaw time constant was 7.5 seconds. The yaw axis eigenvector was found to be 9 degrees from the spatial vertical.

OKAN: 90 DEGREE TILT



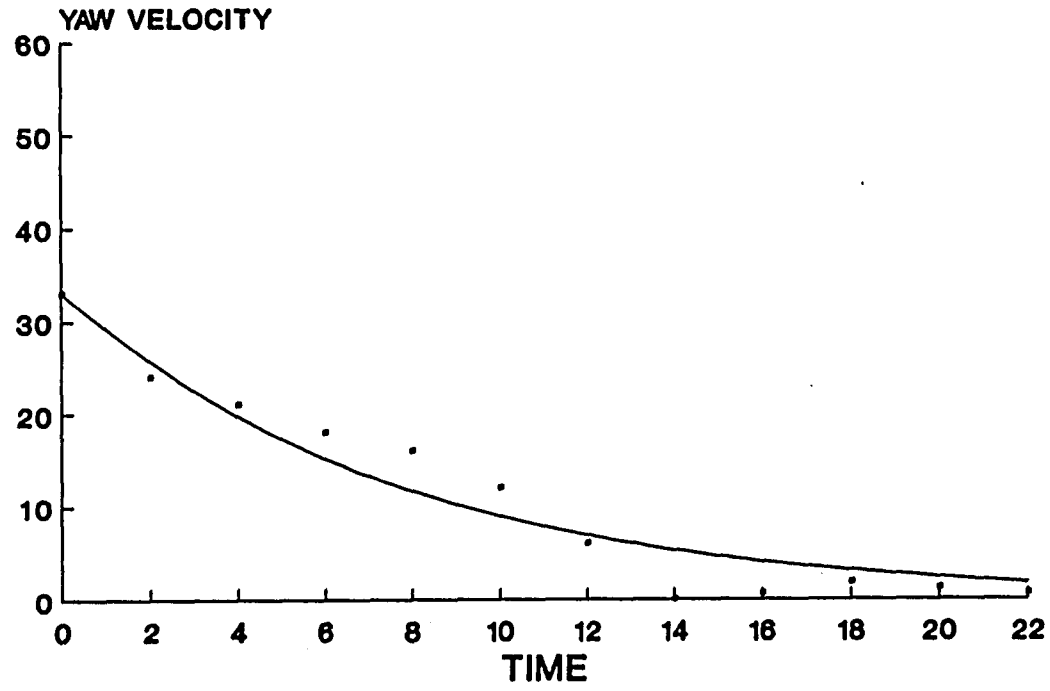


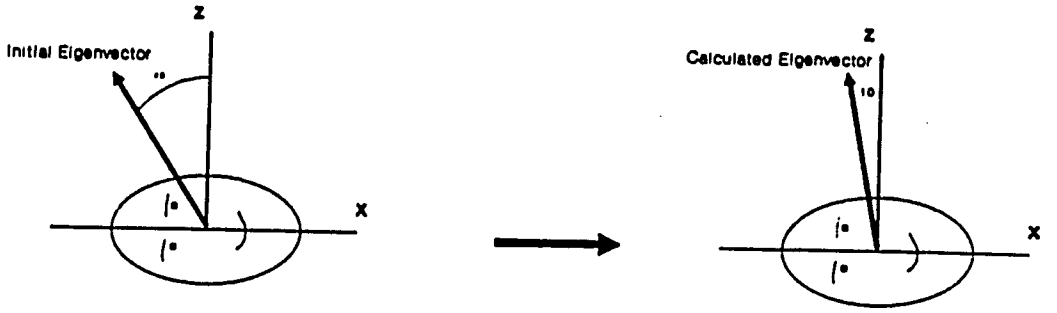
Figure 3.6.1

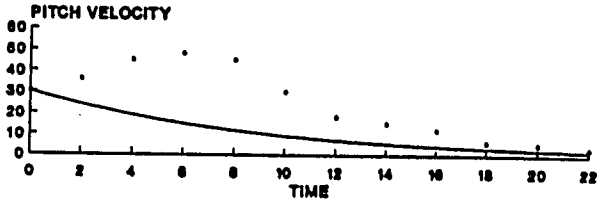
rithm with the yaw component of velocity as input. For the yaw axis velocity there are two parameters, the coefficient c_{33} and the eigenvalue λ_3. For a 90 degree tilt (Figure 3.6.1), we initialize c_{33} to the initial horizontal velocity which is 33. The time constant for the yaw component for the upright is approximately 12 seconds and is known to decrease gradually with tilt angle. Therefore, we set λ_3 initially to .1 (10 seconds). The Marquardt method produces $\lambda_3 = .134$ (corresponding to a 7.4 time constant).

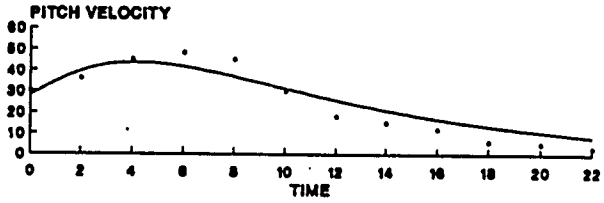
To get an upper bound for λ_1 we run the one dimensional Marquardt algorithm on the pure vertical data for a 90 degree tilt (see figure 3.6.5e for data points). This gives us a time constant of approximately 10 seconds. We run the three dimensional algorithm, this time on the vector valued function holding c_{31} and λ_3 fixed, initializing the the vertical eigenvector to 10 degrees from the spatial vertical. The program computes the initial values of the parameters to be $c_{11} = -157$, $c_{12} = 187$. The attribute list corresponding to the parameters is set so that λ_3, c_{31} and c_{33} are fixed while c_{11}, c_{13}, and λ_1 are allowed to vary. The algorithm then produces $c_{11} = -125$, $c_{12} = 159$, and $\lambda_1 = .206$. This implies that the vertical eigenvector is tilted 11.7 degrees from the spatial vertical with a pitch time constant of 4.8 seconds. The cross-coupled data overlayed by this fit is shown in figure (3.6.1).

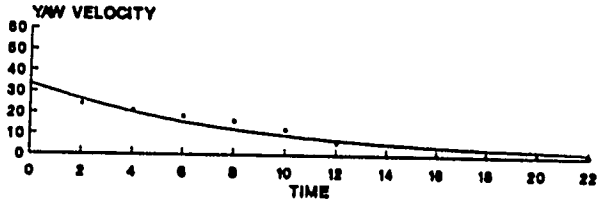
If the eigenvalues are fixed and the yaw axis eigenvector is initialized to 40 degrees from the spatial vertical, this method will converge to a vertical eigenvector of approximately 10 degrees. This is shown in figure (3.6.2). Note that when the eigenvector is far from the true value, the pitch component fits

Figure 3.6.2 This figure demonstrates how the method finds the proper yaw axis eigenvector. In this example, the subject is tilted 90 degrees right ear down and the yaw axis eigenvector is initialized to 40 degrees from the spatial vertical. The pitch and yaw velocities for this case are below the stick figure on the left. The method converges to a yaw axis eigenvector of approximately 10 degrees. The stick figure and velocities corresponding to this case are shown on the right. In both cases the pitch and yaw eigenvalues are held fixed at 5 and 7.5 respectively.


Z
Initial Eigenvector
X
Z
Calculated Eigenvector
X


PITCH VELOCITY
TIME


PITCH VELOCITY
TIME


YAW VELOCITY
TIME

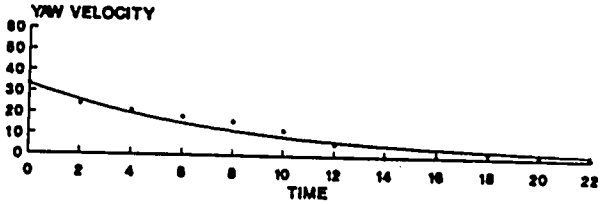

YAW VELOCITY
TIME

Figure 3.6.2

very poorly. Since the yaw time constant is held constant, the yaw velocity fit is the same for both cases.

As we have shown above, our parameter estimation procedure corresponds well with the experimental data. The figures which follow (Figures 3.6.3 a-g), show how the parameters found match the experimental data for tilts of 20, 40, 50, 70, 100, 120 and 130 degrees. In each case, the stimulus velocity during OKN was 60 degrees/sec to the right about the animal's vertical axis. The subsequent cross-coupled pitch and yaw is shown together with the fit found using the method outlined above.

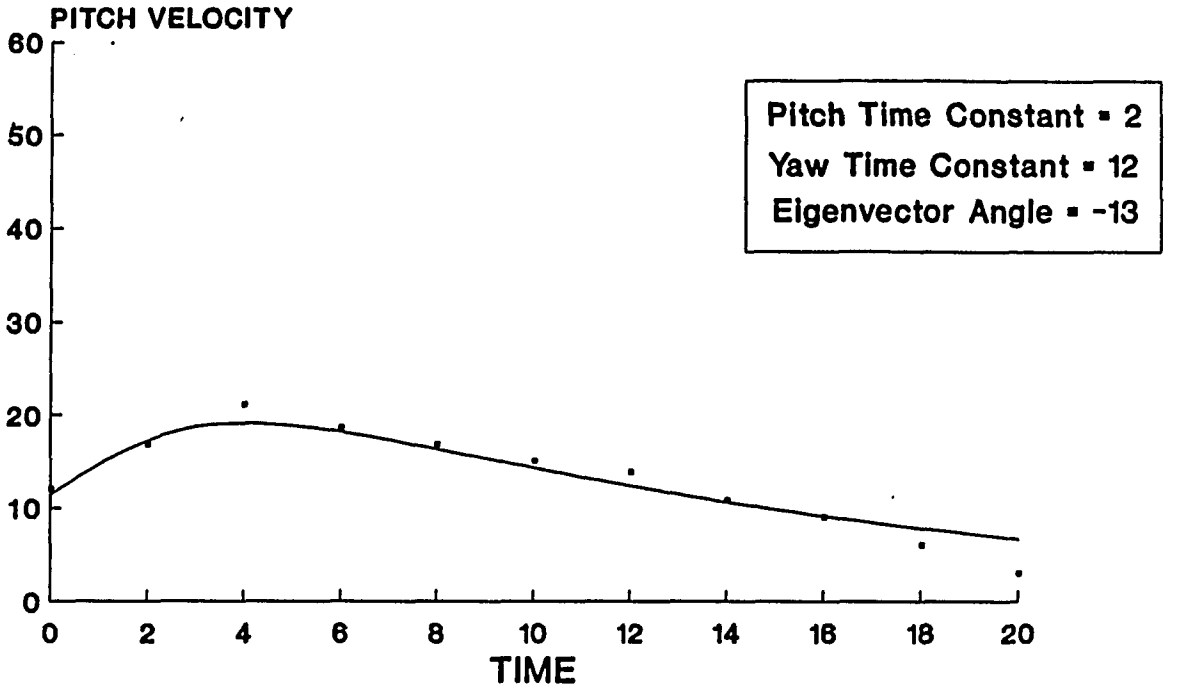
Because the system was assumed to be linear, the pitch component's eigenvalue at each tilt angle is the same as the pitch eigenvalue for a pure pitch stimulus (Figures 3.6.5 a-e). For tilts up to approximately 50 degrees we found this to be true (Figure 3.64).

The pitch axis and roll axis eigenvectors rotated with the head while the "yaw axis" eigenvector stayed close to the spatial vertical regardless of tilt angle. The parameters found, suggest that for small tilts the vertical eigenvector is sometimes tilted in the direction opposite to that of the animal tilt but still close to the spatial vertical (Figure 3.6.6). It will be shown how this function will be utilized in estimating the initial eigenvector for arbitrary orientations of the head (section 3.7)

An evaluation of the fits shows that the first order model fits the data well over a wide range of tilt angles. However, a quantitative approach involves computing a "goodness of fit"

Figure 3.6.3 Cross-coupled pitch and yaw OKAN for tilts of 20, 40, 50, 70, 100, 120, and 130 degrees (a-g). The data points are shown as dots. The solid lines are the fits found using the method outlined above. Each figure shows the pitch and yaw time constants and the yaw axis eigenvector angle found for that tilt.

OKAN: 20 DEGREE TILT



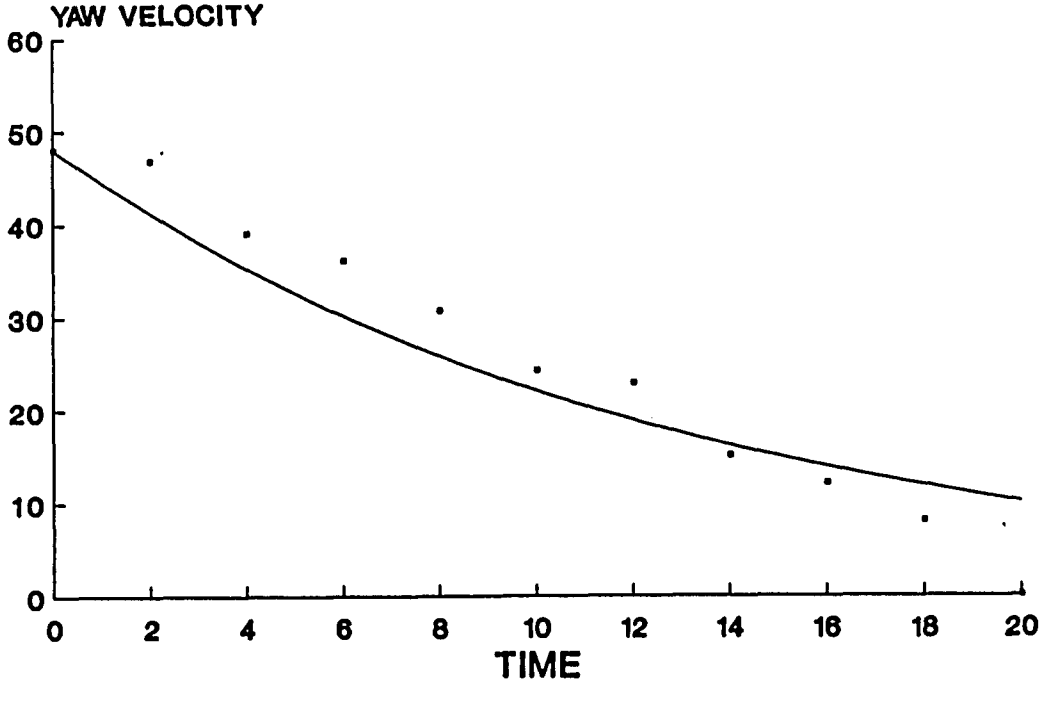
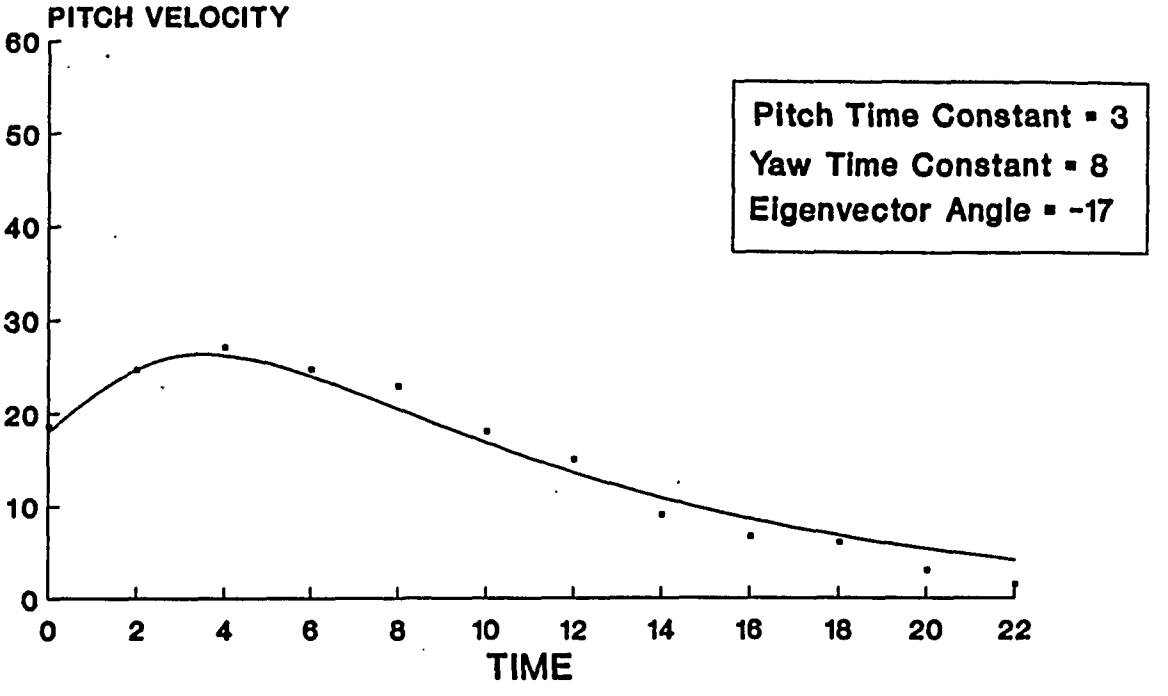


Figure 3.6.3a

OKAN: 40 DEGREE TILT



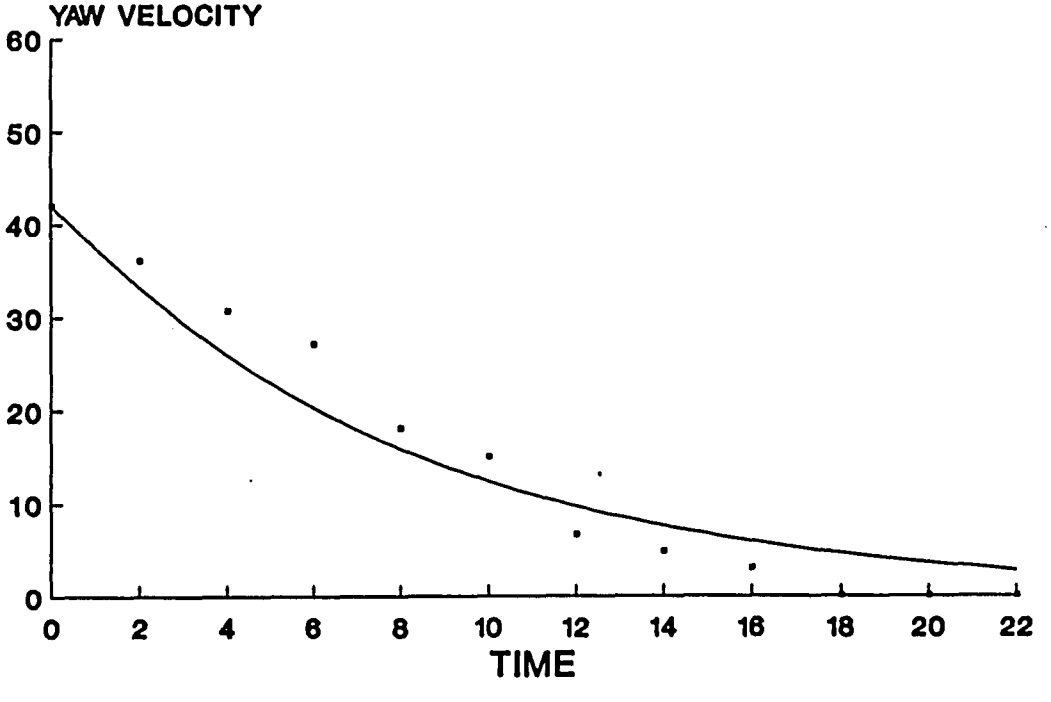


Figure 3.6.3b

OKAN: 50 DEGREE TILT

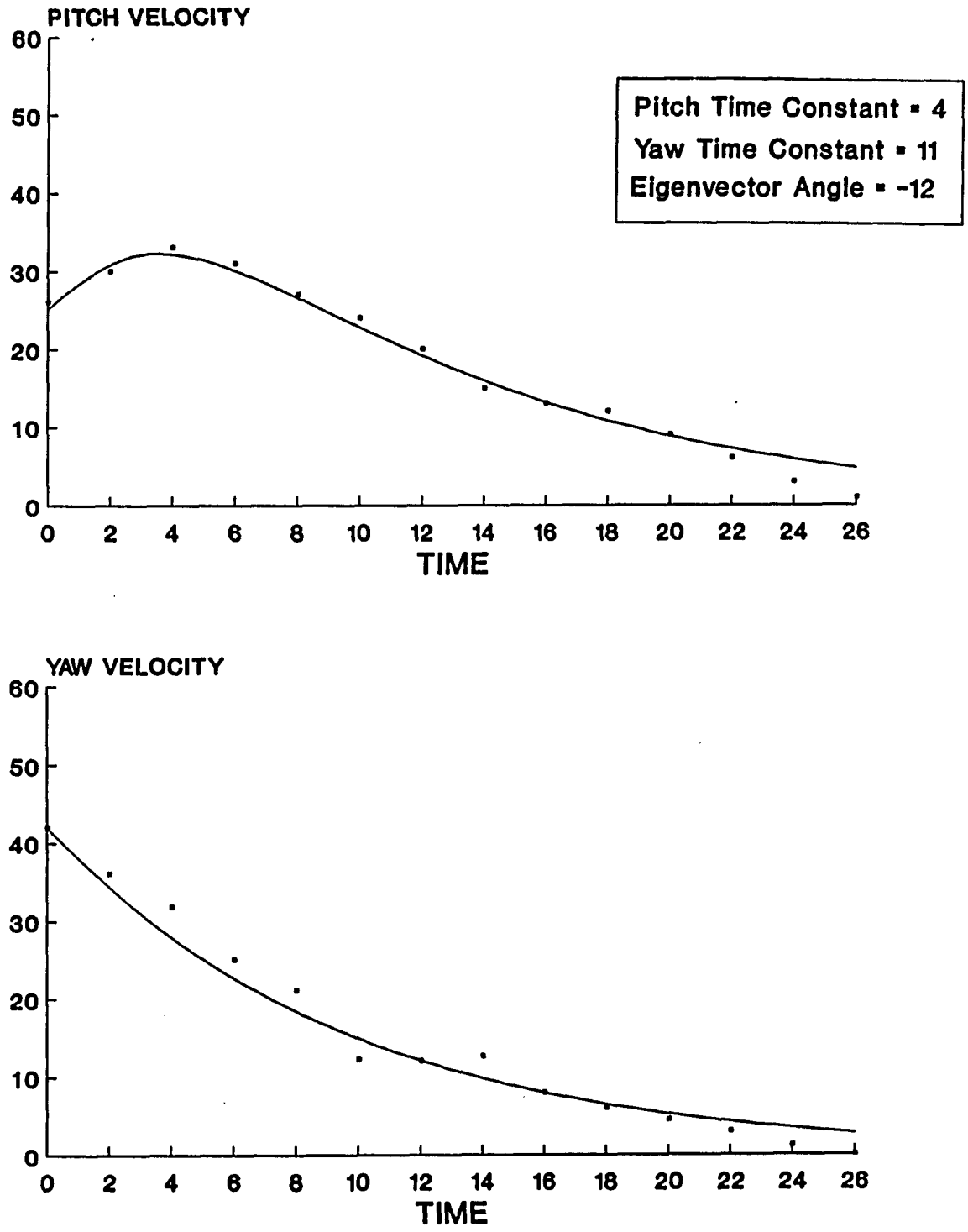
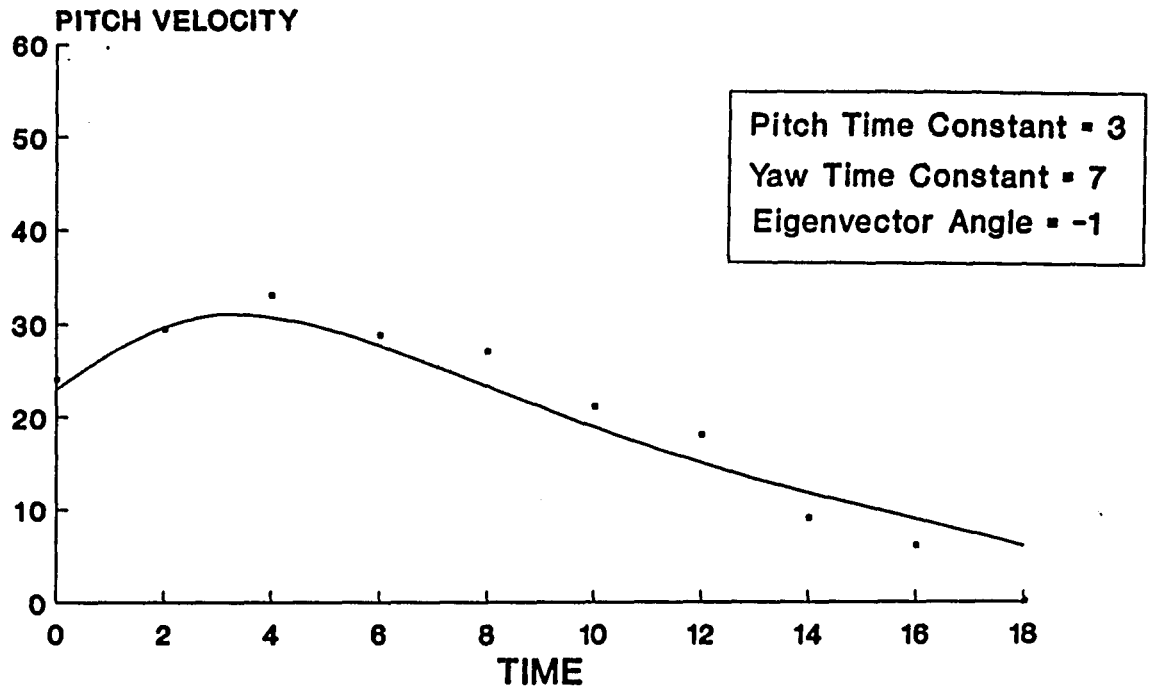


Figure 3.6.3c

OKAN: 70 DEGREE TILT



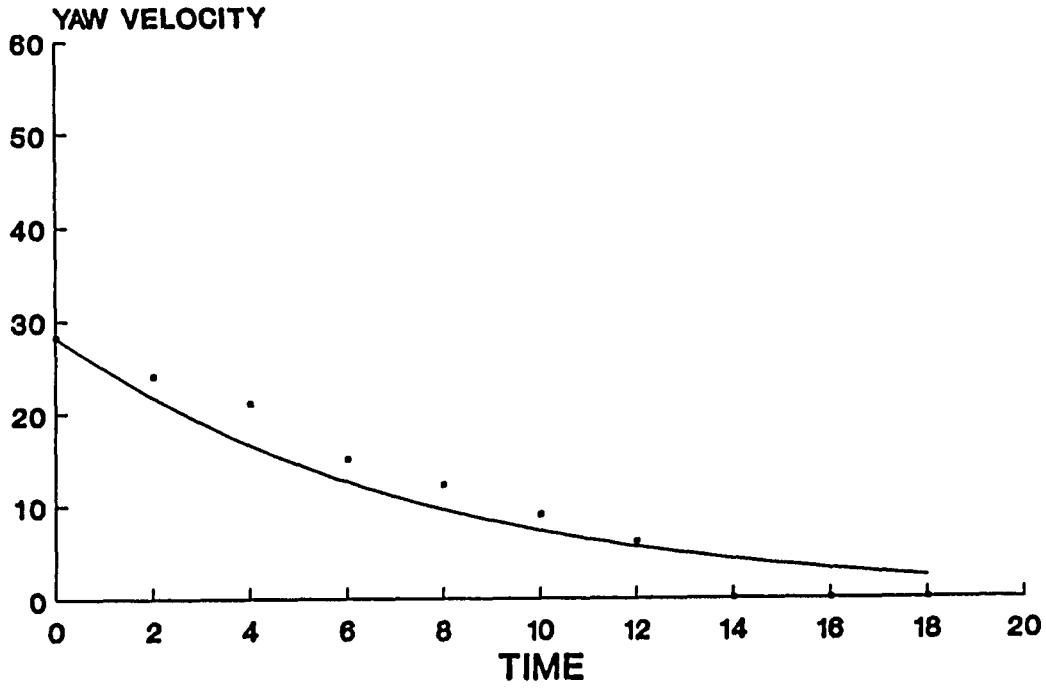


Figure 3.6.3d

OKAN: 100 degree tilt

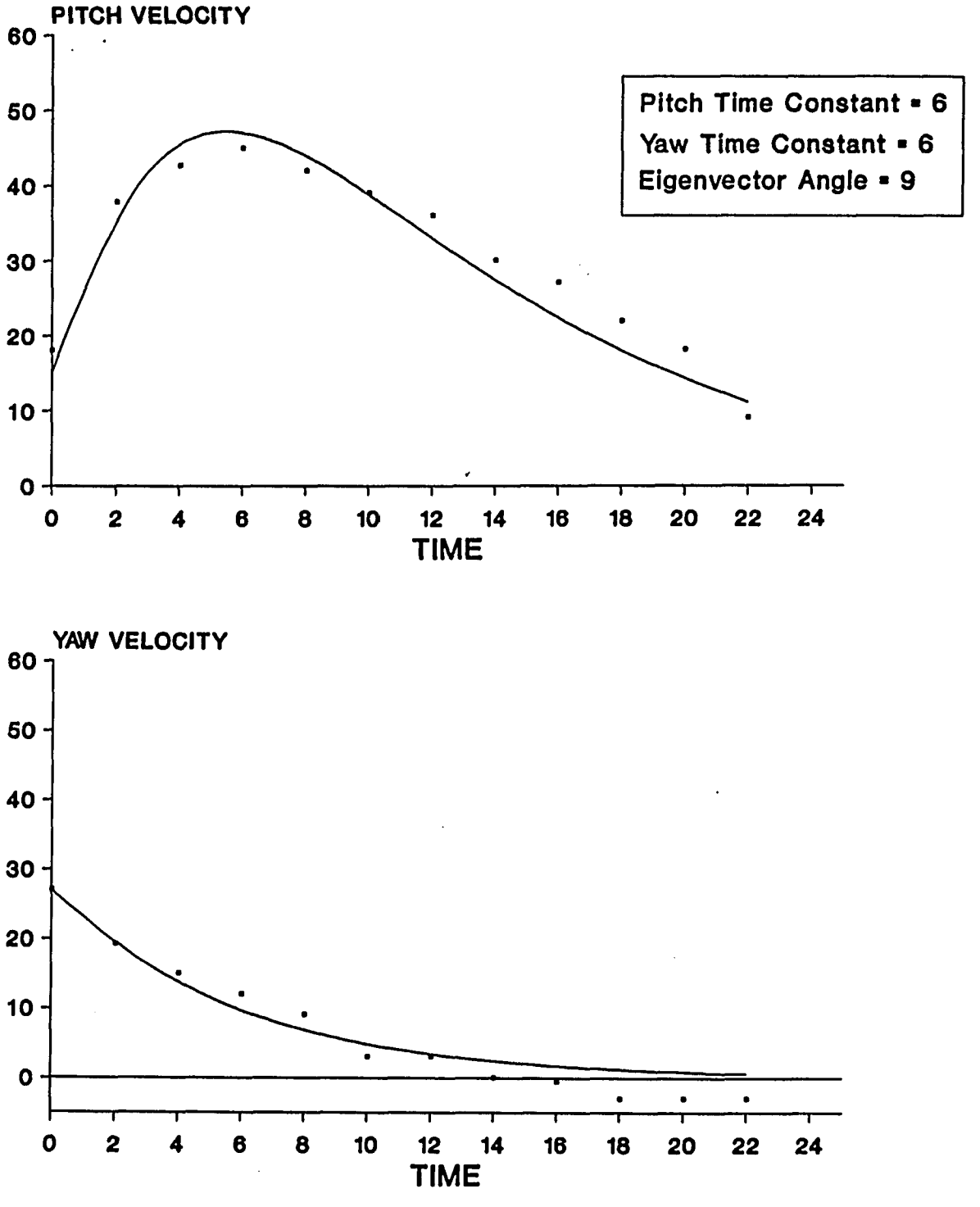
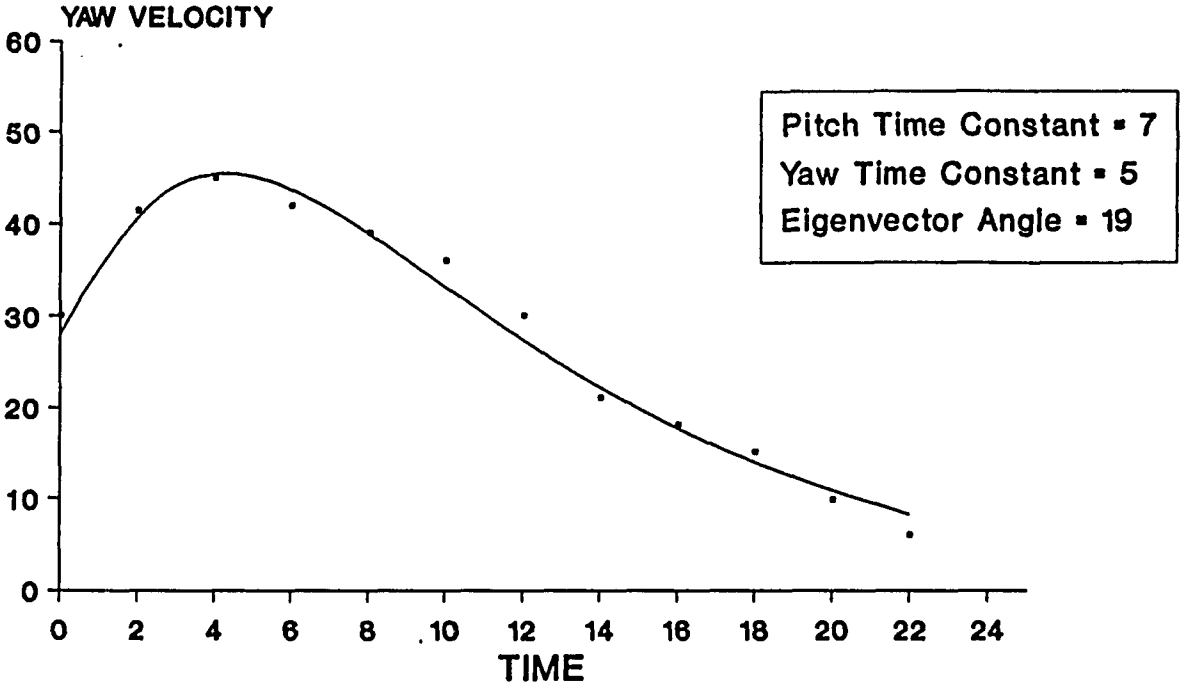


Figure 3.6.3e

OKAN: 120 DEGREE TILT



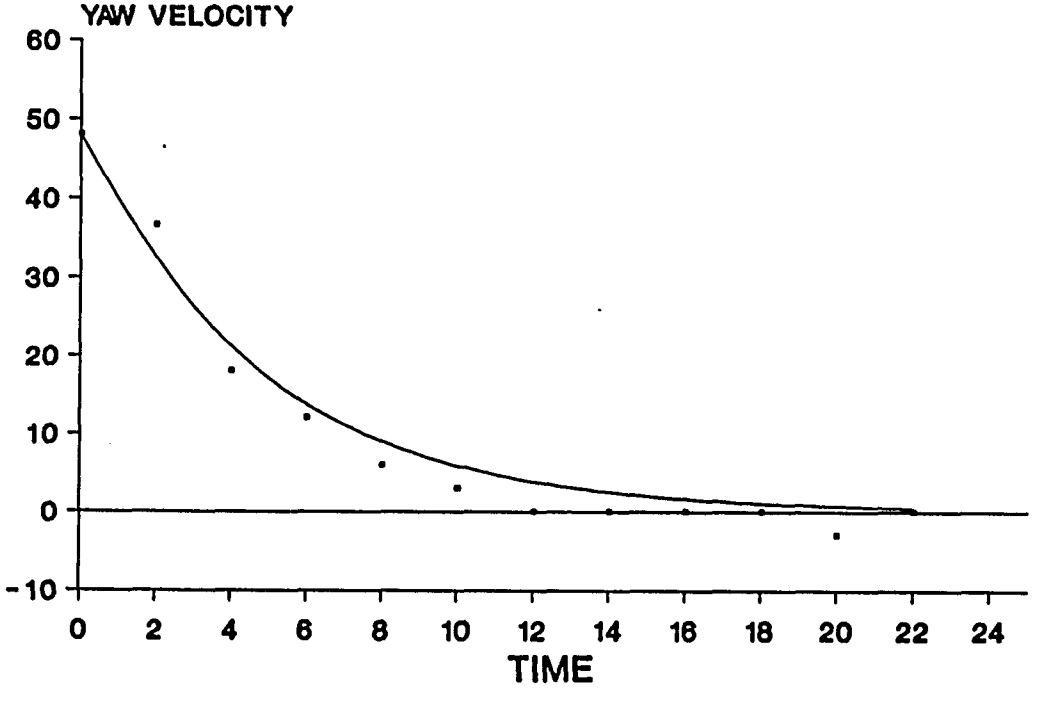


Figure 3.6.3f

OKAN: 130 DEGREE TILT

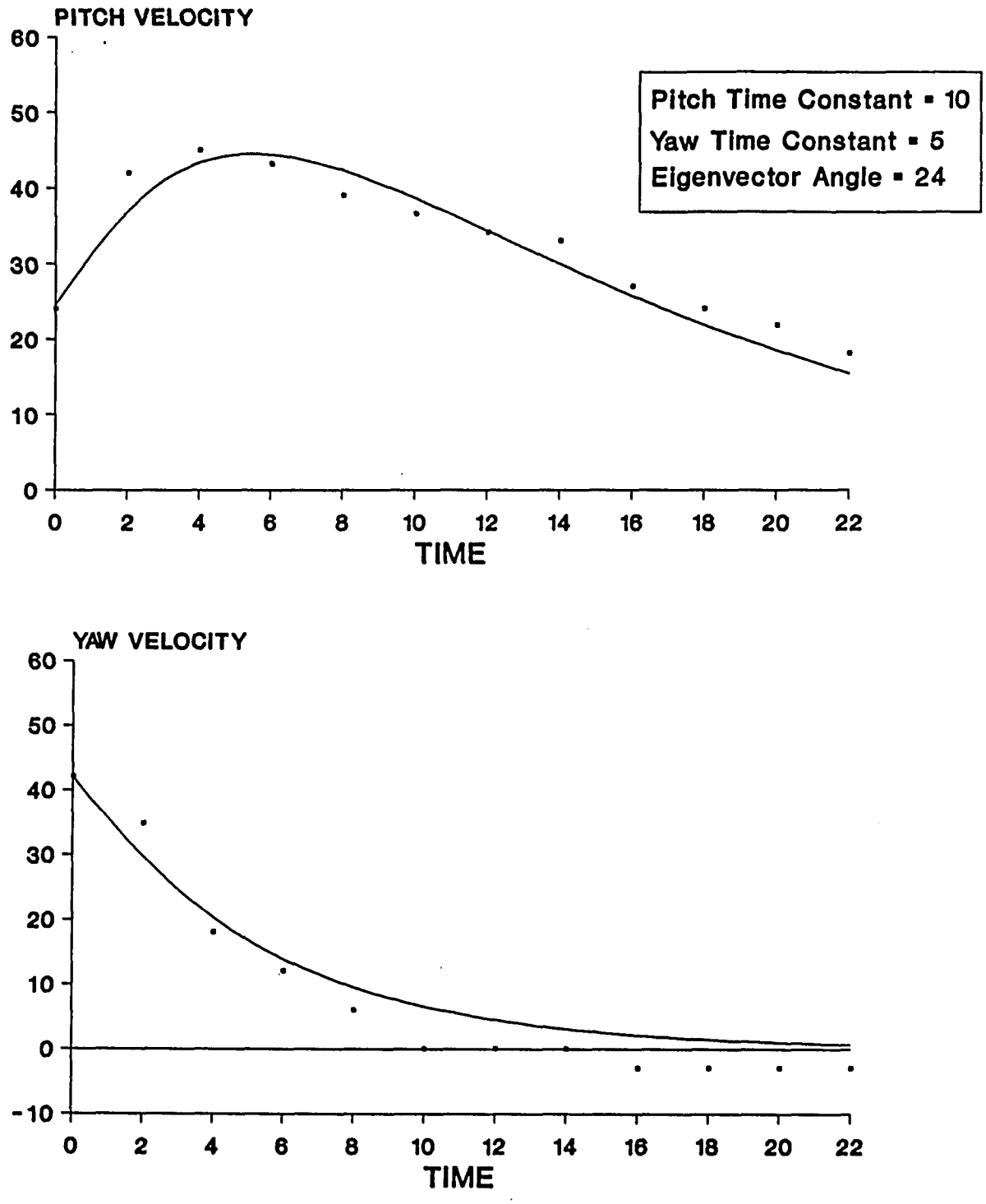


Figure 3.6.3g

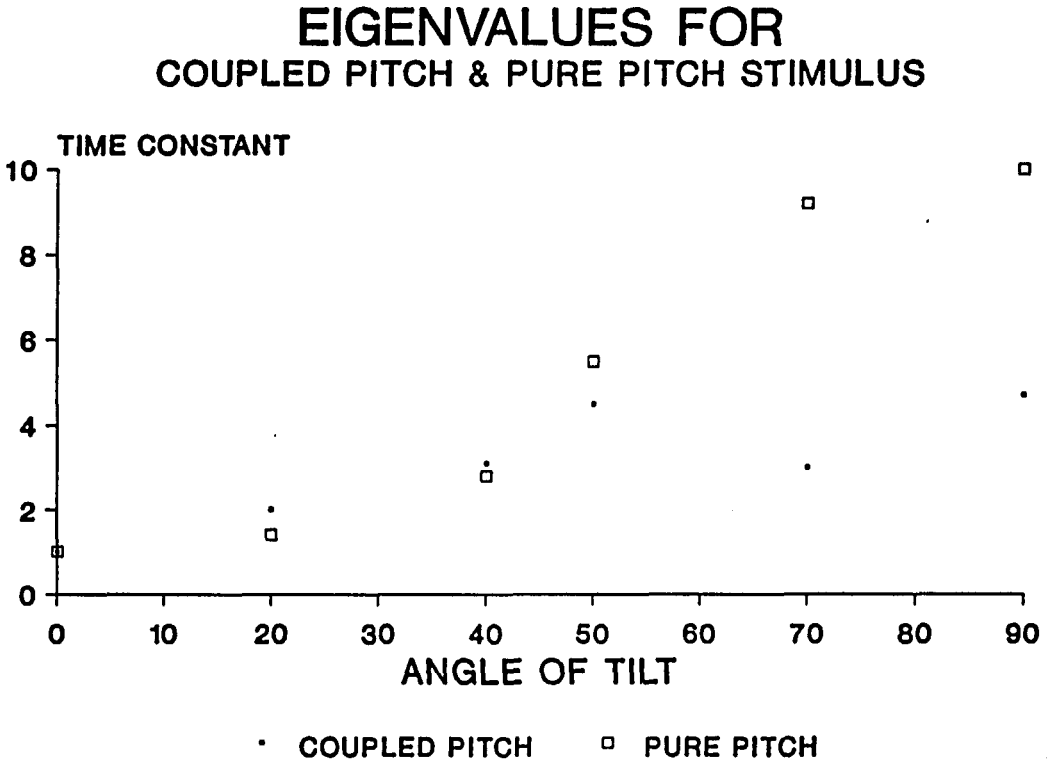
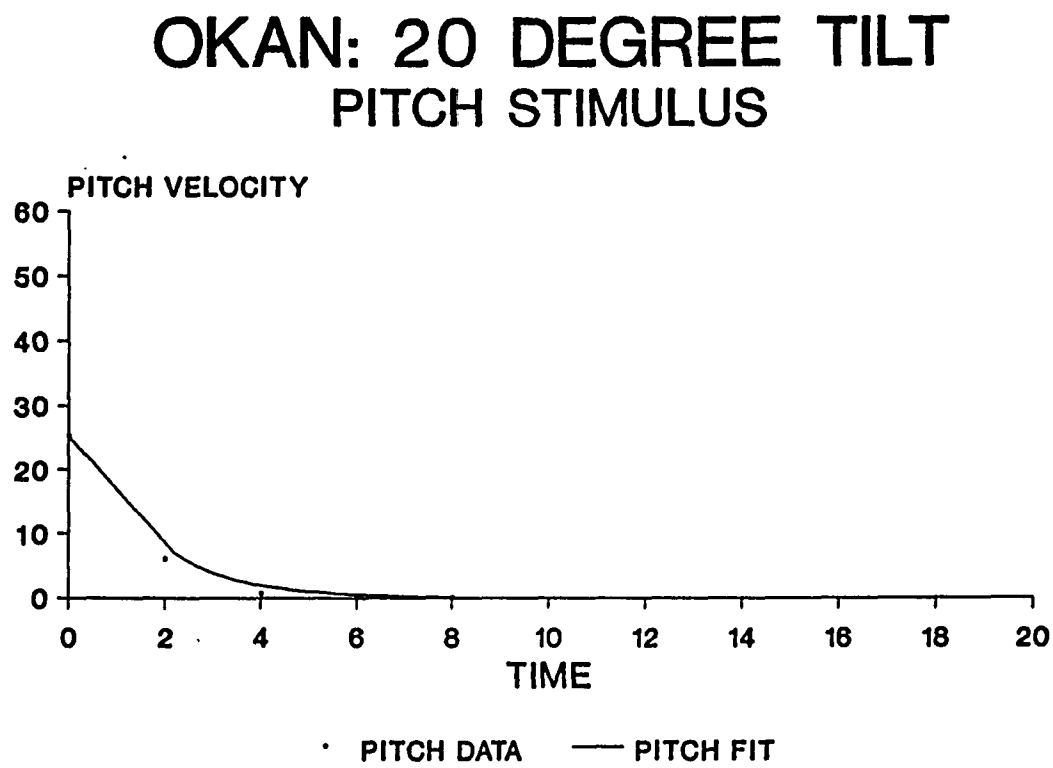


Figure 3.6.4 Time constants (inverse of eigenvalues) associated with upward pitch for cross coupled and pure pitch stimulus as a function of tilt angle.

Figure 3.6.5 Pure pitch OKAN for tilts of 20, 40, 50, 70 and 90 degrees (a-e). The data points are shown as dots. The solid lines are the fits found using the nonvector Marquart algorithm. The pitch time constants for these tilts are:

Upward Pitch axis

	Time constant	Eigenvalue(1/tc)
20:	1.4	.71
40:	2.8	.35
50:	5.5	.18
70:	9.2	1.08
90:	10.0	1.00



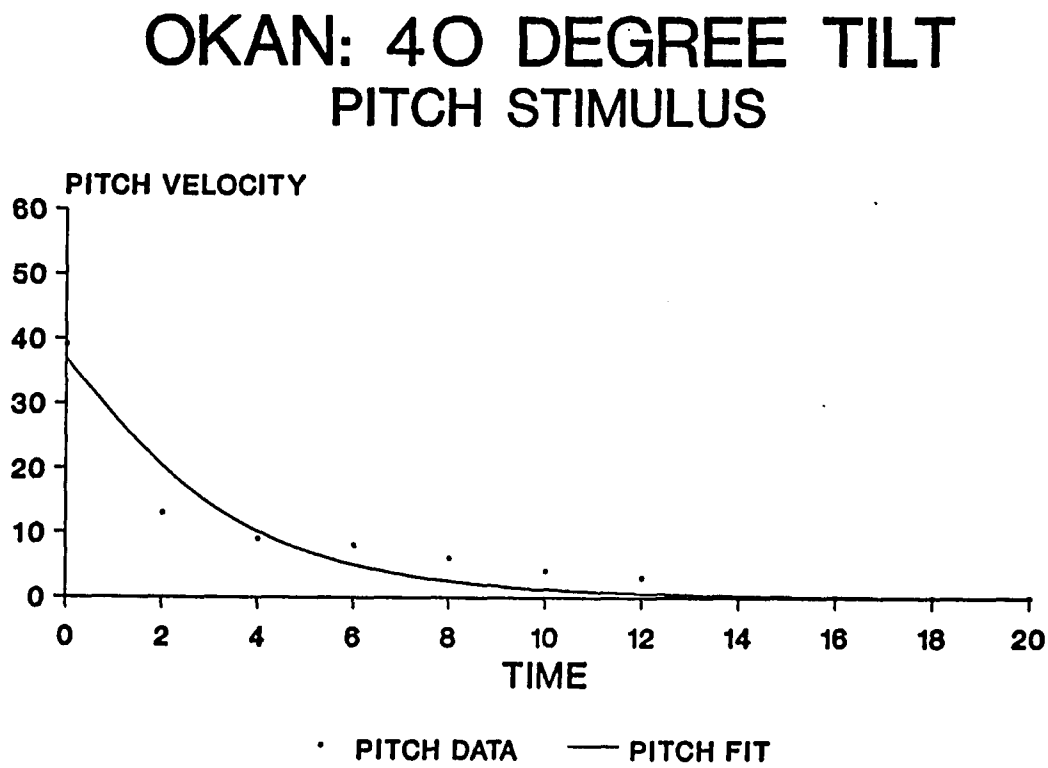
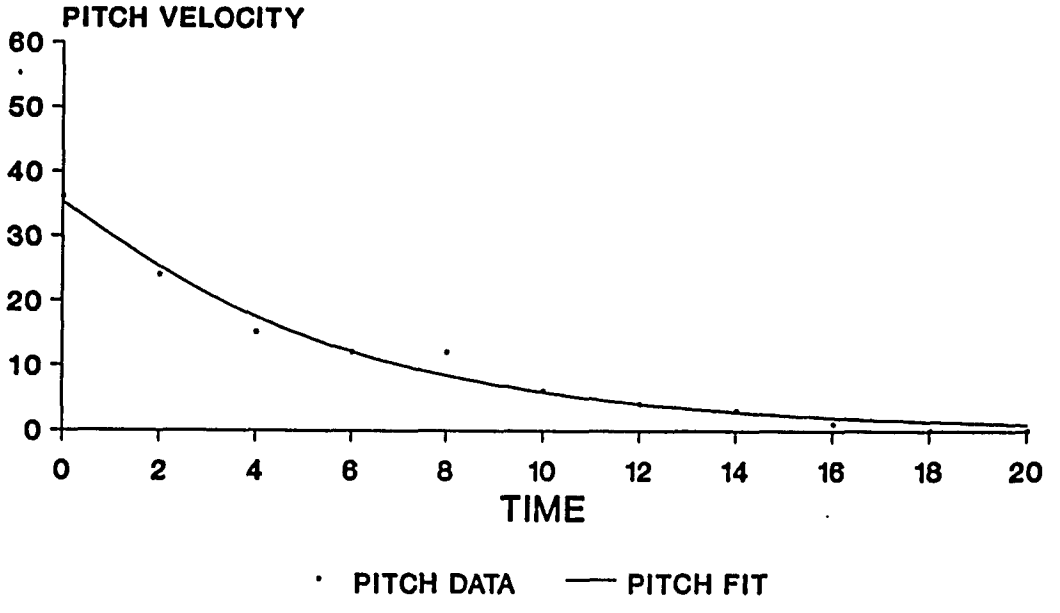


Figure 3.6.5 a,b

OKAN: 50 DEGREE TILT
PITCH STIMULUS



OKAN: 70 DEGREE TILT
PITCH STIMULUS

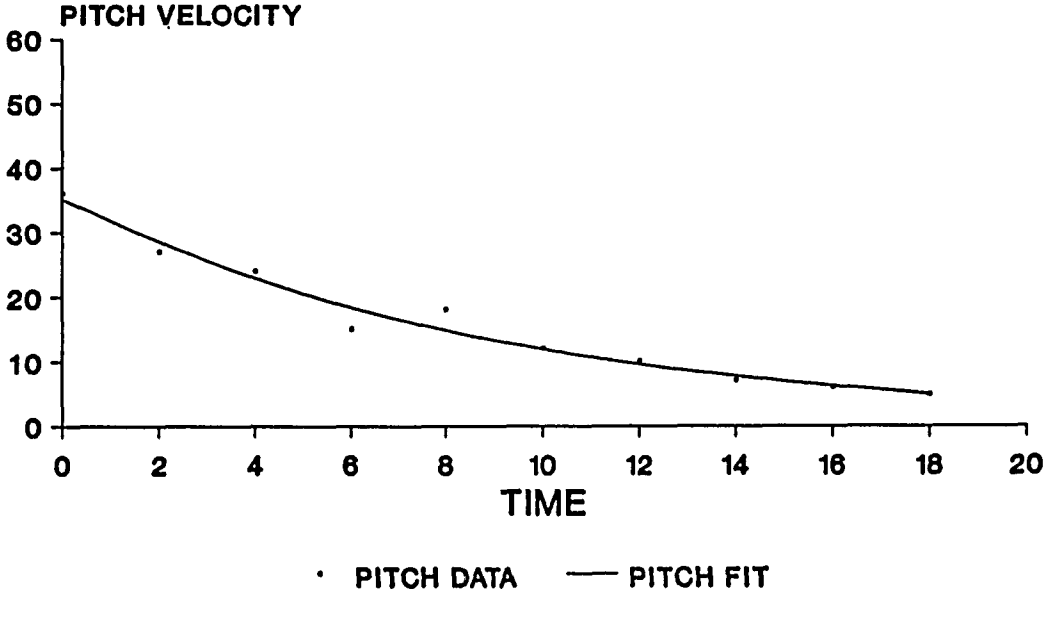


Figure 3.6.5 c,d

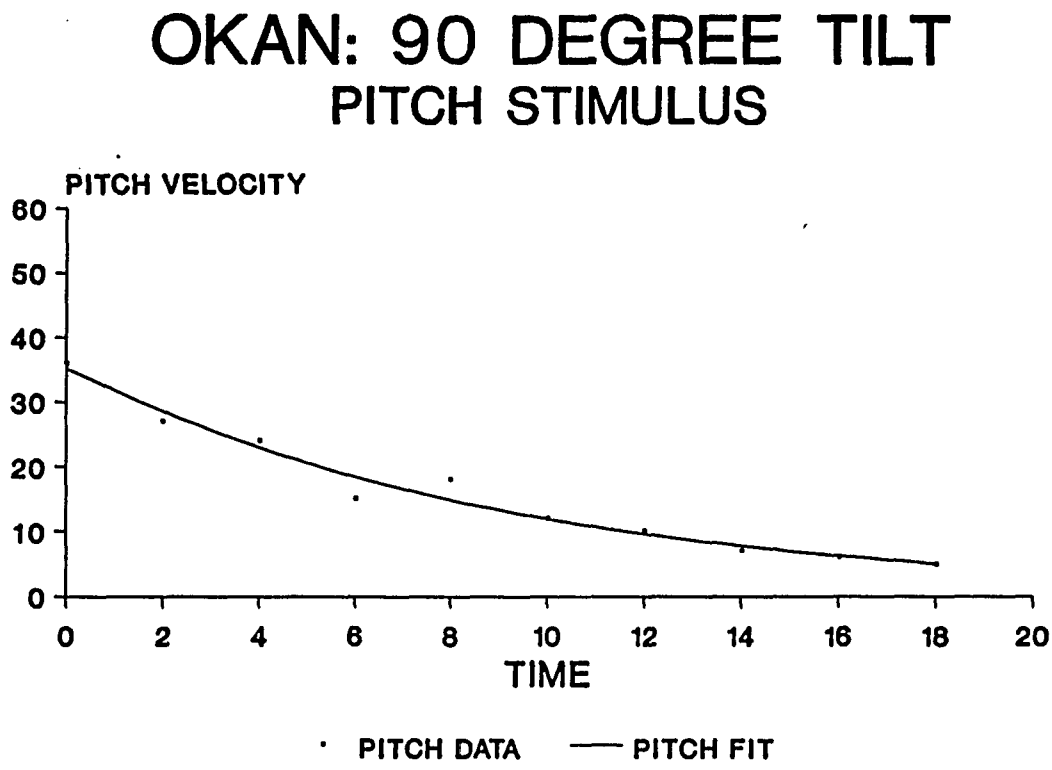

OKAN: 90 DEGREE TILT
PITCH STIMULUS
PITCH VELOCITY
60
50
40
30
20
10
0
0
2
4
6
8
10
12
14
16
18
20
TIME
PITCH DATA
PITCH FIT

Figure 3.6.5 e

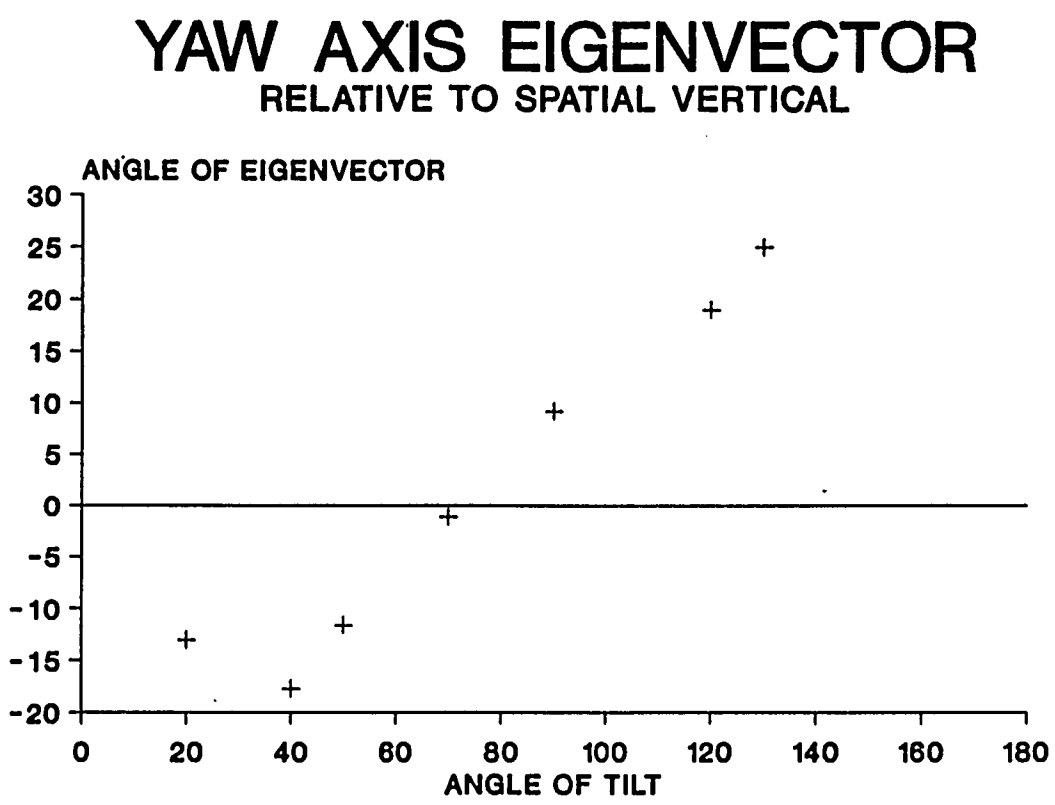


Figure 3.6.6 Yaw axis eigenvector relative to the spatial vertical for tilts of 0, 20, 40, 50, 70, 90, 100, 120, and 130 degrees.

function to express the quality of the data fitting procedure. The chi squared function alone is not sufficient since it is proportionately related to the number of data points. That is, the more points you have, the higher the chi-squared function. Rather than simply dividing by the number of data points, we instead consider the degrees of freedom of the function. The degree of freedom is the difference between the number of data points and the number of parameters used in the algorithm. The ratio of the chi- squared function and the degrees of freedom has been found to be an adequate measure of "goodness of fit" which is independent of input size.

For example, the chi squared value associated with the 90 degree tilt is 42.73. The number of data points used is 24. Although there are 6 parameters, only three are used in the vector valued function fit. Therefore there are 21 degrees of freedom. Thus the "goodness of fit" is found to be 2.

The next section addresses the problem of setting the initial parameters for arbitrary orientations to obtain good convergence characteristics.

3.7 Identification of Parameters for Arbitrary Orientations

In the previous sections we developed a method for determining the eigenvectors and eigenvalues from cross-coupled data and have applied it to various subject tilt angles about the roll axis. The method that has been developed is general and can be applied to arbitrary orientations of the head with regard to gravity. However, a crucial step in the procedure is approximating the initial values for the eigenvectors and eigenvalues.

We have noted previously that the eigenvectors associated with the pitch (x) and roll (y) axes approximately rotate with the head. This is due to the qualitative observation on a small data set that there is little or no cross coupling from the vertical axis to the roll of yaw axis and little or no cross coupling from the roll axis to the pitch and yaw axis. Therefore, for any given orientation their location is approximately along the head pich and roll axis for that orientation. This constitutes a good initial setting for these eigenvectors. However, the vertical eigenvector is known to behave differently. Our hypothesis is that gravity reorients this vector so that it maintains some spatial invariance and is a function of the body vertical.

A reasonable initial approximation to the yaw axis eigenvector is therefore that it lies in the plane determined by the spatial vertical and the body vertical relative to a spatial frame of reference. The initial angle will be determined based on the functial relationship between the angle of the yaw axis eigenvector and tilts about a roll axis (Figure 3.7.1). Although we do

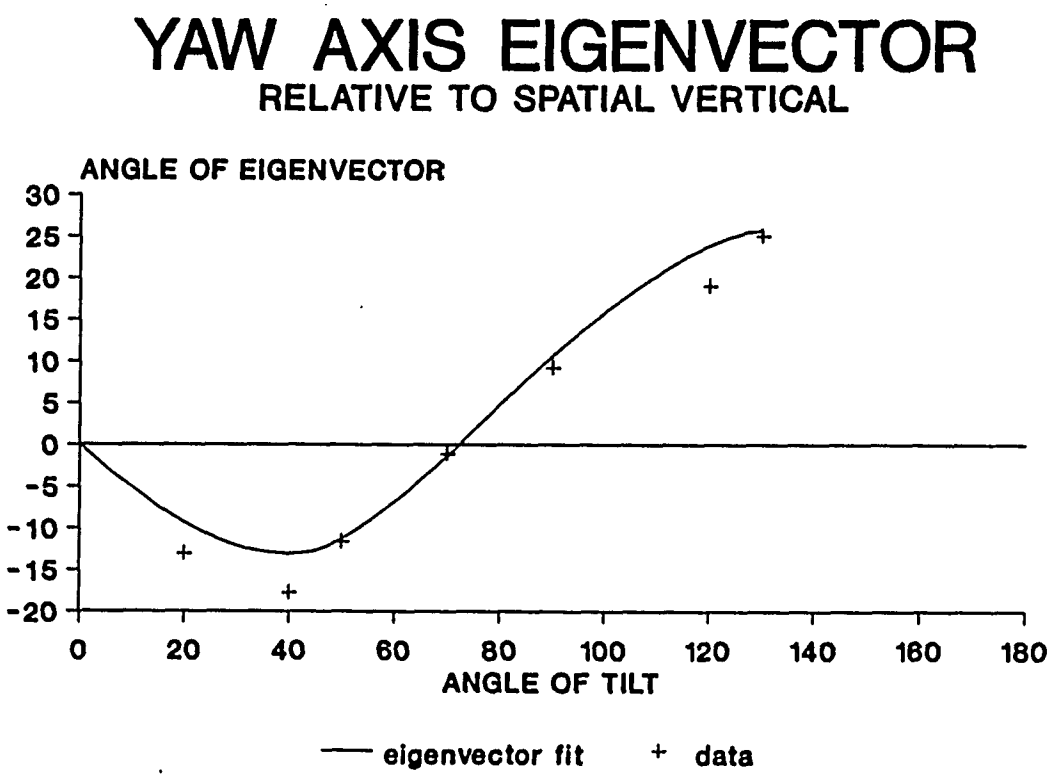


Figure 3.7.1 Relationship of angle of yaw axis eigenvector (relative to spatial vertical) γ_1 to angle of tilt θ. (Fit with sine curve: $f_e = a + b \sin(cx + d)$).

not have data for tilts beyond 120 degrees it is clear that the behavior is a periodic function of the tilt angle. Note the similarity with Figure 2.10. We denote the function which determines the eigenvector in terms of the tilt angle by f_e. By projecting this vector into head coordinates, the initial eigenvector is then suitable for input to the extended Marquardt Algorithm. Specifically this can be seen by examining figure 3.7.2. This assumption is approximately true for tilted and pitched positions of the head (Raphan & Cohen, 1988). The method of finding the initial yaw axis eigenvector is as follows:

Let,

u_1 be the original vertical eigenvector
u_2 be the direction of the body vertical
v be the new vertical eigenvector
γ_1 be the angle between u_1 and $v = f_e(\theta)$
γ_2 be the angle between v and $u_2 = \theta - \gamma_1$
$\theta = \gamma_1 + \gamma_2$, i.e., θ is the angle between u_1 and u_2

Given the assumption that v remains in the plane formed by u_1 and u_2, we conclude that:

$$v = a_1u_1 + a_2u_2 \qquad (3.7.1)$$

for some real numbers a_1 and a_2

We are assuming that u_i and v are unit vectors since we are only considering their direction. We complete the parallelogram, to form the sides a_1 and a_2 (Figure 3.7.3).

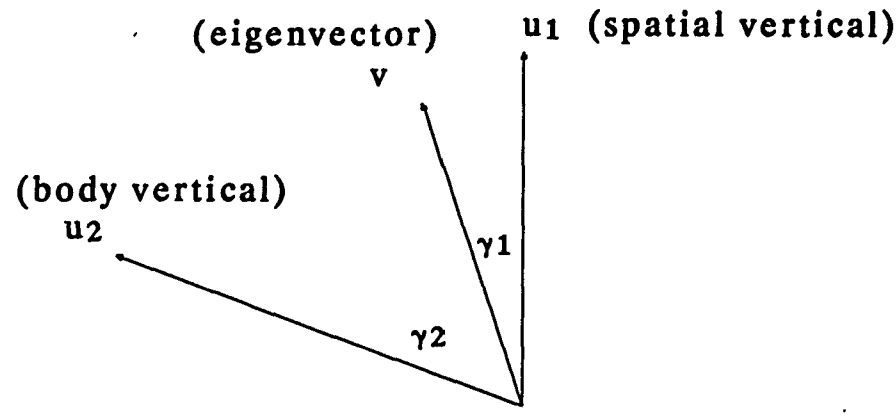


Figure 3.7.2 Diagram of an arbitrary orientation of subject in 3- space. The new vertical eigenvector is shown as v.

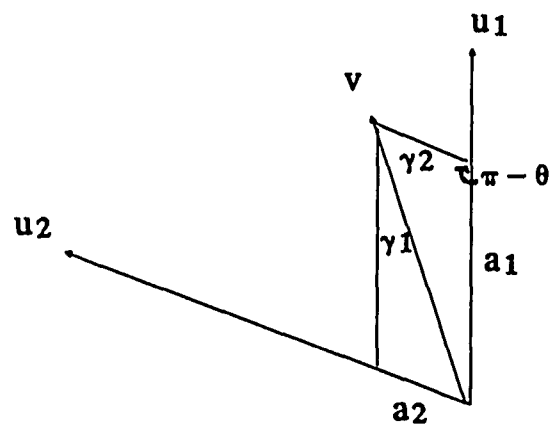


Figure 3.7.3 Diagram showing the method used to find the coordinates of the new vertical eigenvector, v.

Using the law of sines, we obtain:

$$a1/\sin \gamma_2 = 1/(\sin (\pi - \theta)) = 1/(\sin \theta)$$

Solving for a_1, we have:

$$a_1 = \sin \gamma_2 / \sin\theta$$

and similarly,

$$a_2 = \sin \gamma_1 / \sin\theta$$

Substituting back for v in equation (3.7.1), we have:

$$\underline{v} = (\sin \gamma_2/\sin\theta)\, u_1 + (\sin \gamma_1/\sin \theta)\, u_2$$

With this procedure, we can now choose for any orientation of the head, the initial value of the yaw axis eigenvector.

In addition to choosing the eigenvectors for arbitrary orientation, we must also find initial values for the eigenvalues as function of gravity. In the upright position the values for the eigenvalues are known. In this case the Q transformation is simply the identity matrix. As the subject is rotated, the Q matrix modifies the eigenvalues associated with the new eigenvectors.

One of the observations which can be made from the small data set is that the pitch axis is approximately the same for a pure pitch stimulus as for cross-coupling. Thus, the roll and pitch eigenvalues obtained from pure pitch and roll stimuli are good initial values for the Marquardt algorithm.

The algorithm summarizing the steps for finding the system matrix for a given angle is as follows:

1. Determine from experiments the eigenvalues associated with the pitch, roll and yaw axes for a given orientation. This is done by giving a corresponding pure pitch, roll or yaw stimulus and measuring the inverse of the time constant associated with each axis. These values are used as the initial parameters for determining the eigenvalues.

2. Determine the initial value for the yaw axis eigenvector by using the function $\gamma_1 = f_e(\theta)$ (Figure 5.1) to determine its angle relative to the spatial vertical.

3. Compute $R = R(\phi,\theta,\psi)$, the rotation matrix for the position of the head where ϕ,θ,ψ: are the Euler angles associated with the rotation of the head.

4. Determine the angle between the spatial vertical and the body vertical:

$$\gamma_1 = f_e(\theta)$$

$$\gamma_2 = \theta - \gamma_1$$

5. Compute the initial value of the eigenvector:

$$v = (\sin \gamma_2/\sin\theta)\, u_1 + (\sin \gamma_1/\sin\theta)\, u_2$$

4. Form P: $(R(e_1), R(e_2), \underline{v})$

whose columns are the eigenvectors in a spatial reference frame.

5. Run the Marquardt program to obtain the converged values of the yaw axis eigenvector and the eigenvalues. The eigenvalues become the diagonal of the matrix QH_o and the eigenvectors become the columns of the P matrix.

6. Form H: $H = Tcan(R^{-1} (PQH_oP^{-1}) R) Tcan^{-1}$

3.8 Model Simulations and Predictions

In order to test the model under a feedback arrangement and to examine the effects of the direct pathway the computed system matrix was used in the three dimensional model shown in Figure 3.8.1. The program which implements this model was written in Fortran and is given in Appendix C. For the simulations which follow (Figures 3.8.2- 3.8.4) the animal is right side down with a stimulus velocity of 60 degrees/second to the right. The lights are on for 30 seconds and then are extinguished for the duration of the simulation.

For the upright condition (Figure 3.8.2) there is no cross-coupling present and therefore only the yaw eye velocity is shown. The yaw velocity jumps to 25 at the onset of OKN and builds slowly to a steady state value of approximately 47. When the light goes off OKAN is induced and in the absence of the direct pathway there is drop to 42 and a decay to 4 seconds by the end of the simulation.

Figure 3.8.3 shows a simulation for an animal tilted 50 degrees from the vertical. In this case, there is cross-coupling from the yaw to pitch axis. Note that the rate of decay for the yaw OKAN is more rapid than for the upright (Figure 3.8.2). The OKN response for pitch is somewhat suppressed but there is a slight buildup. During OKAN there is rise in velocity to approximately 40 degrees and then a decay. For a 90 degree tilt (Figure 3.8.4) the decay of yaw velocity is faster than for 50 degrees (Figure 3.8.3) and the cross-coupling is more pronounced.

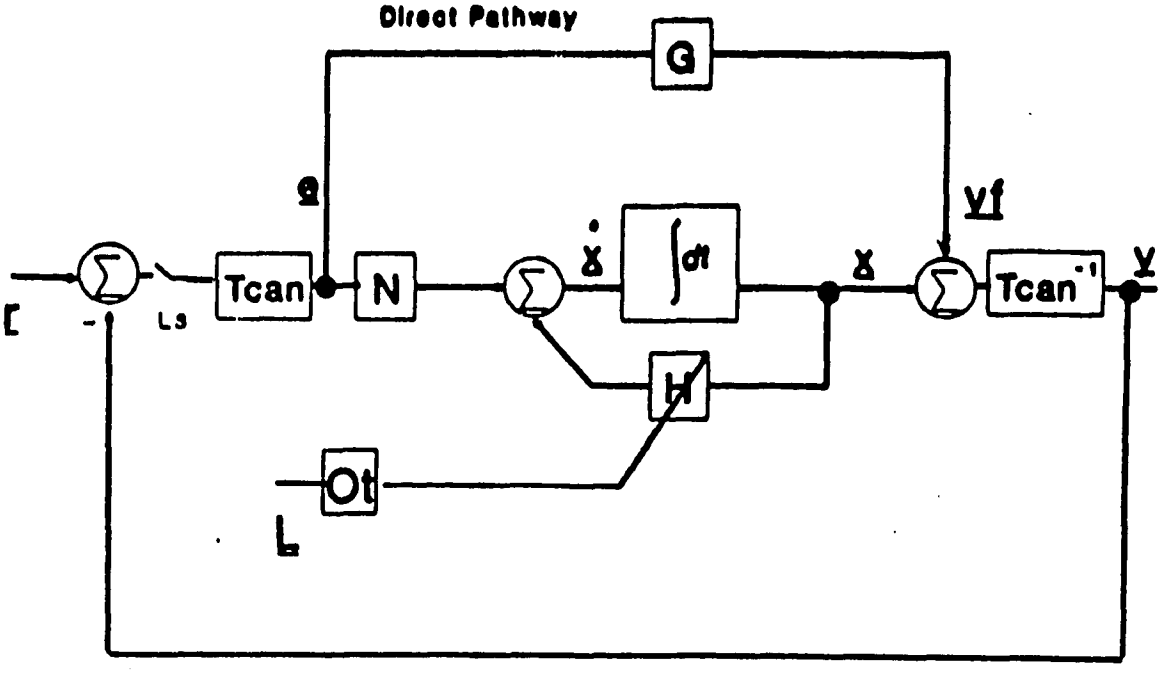


Figure 3.8.1 Three Dimensional Model of the visual input to the velocity storage integrator. The surround velocity is represented by r. The retinal slip signal (e), is transmitted centrally to two elements. One is the direct pathway that is responsible for rapid changes in eye velocity. The second is the visual coupling to the integrator. Tcan transforms from head to canal coordinates and Tcan-1 is the transformation back. H is the system matrix representing the dynamics associated with the integrator. Ot is the transformation that converts linear acceleration (gravity) measured with respect to a head reference frame, into command signals that modify H.

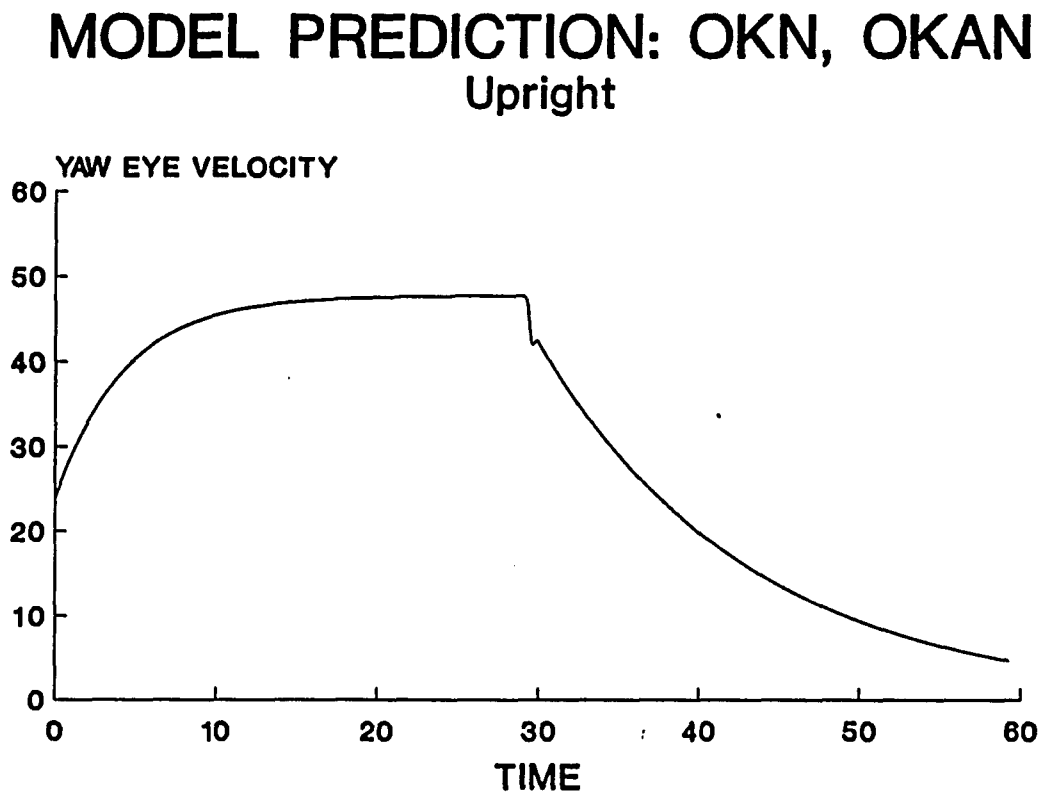


Figure 3.8.2 Model Prediction for OKN and OKAN for the upright condition. Note the absence of pitch velocity since there is no cross-coupling for this case. The time constant associated with OKAN was chosen to be 13 seconds.

MODEL PREDICTION: OKN, OKAN

50 degree tilt

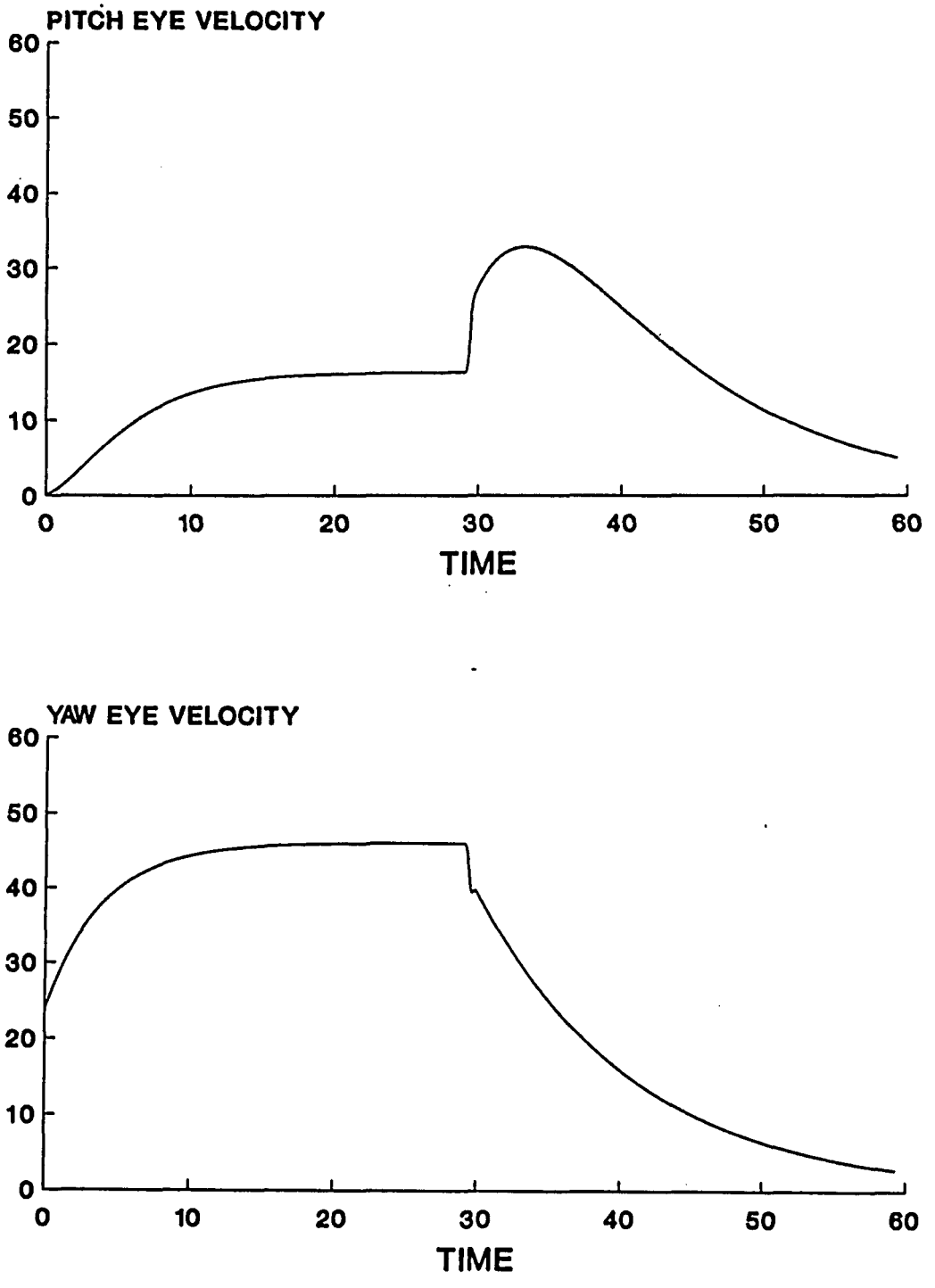


Figure 3.8.3 Model Prediction of OKN and OKAN for a tilt of 50 degrees. Note the cross-coupling from yaw to pitch velocity.

MODEL PREDICTION: OKN, OKAN

90 degree tilt

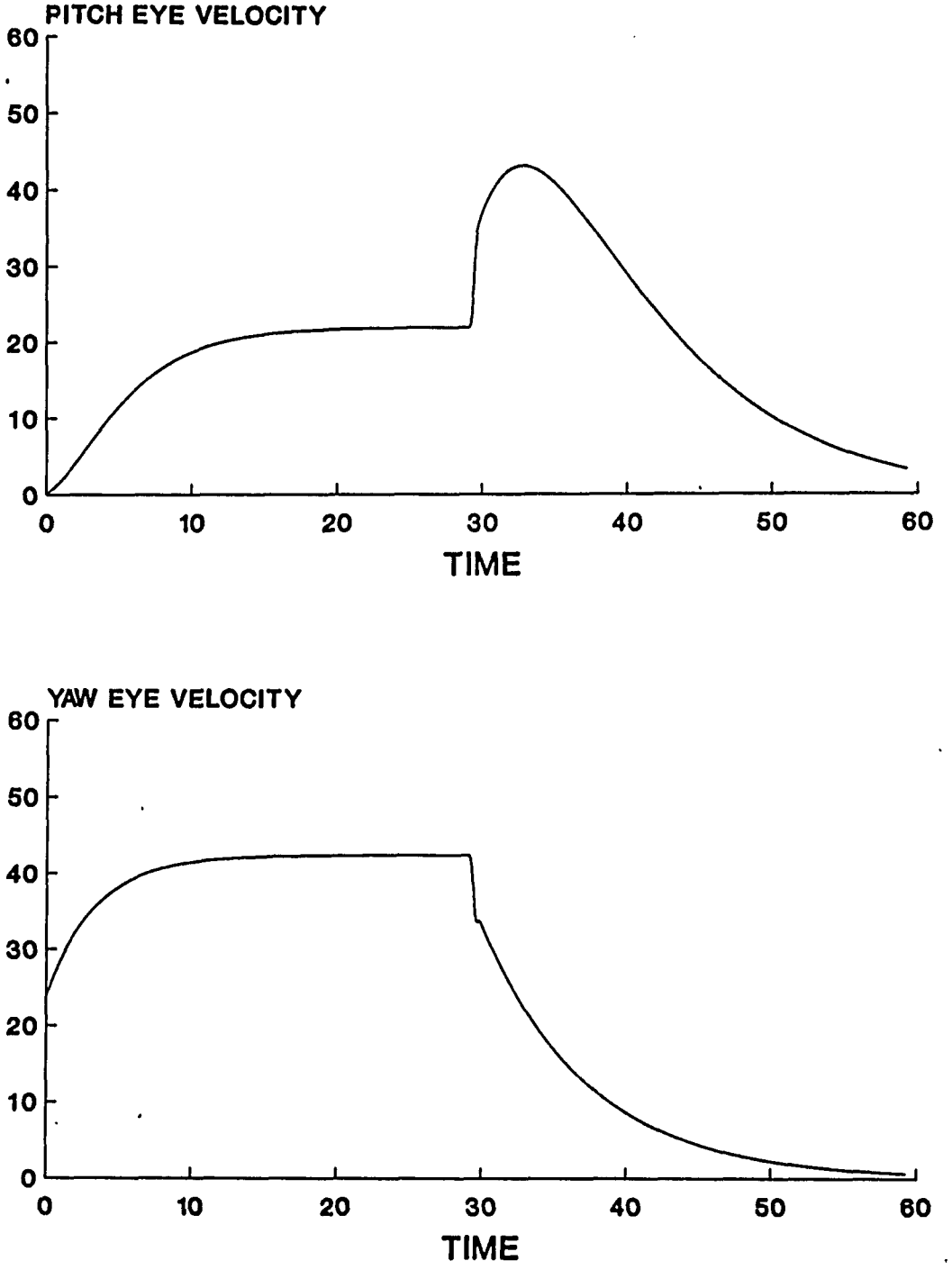


Figure 3.8.4 Model Prediction of OKN and OKAN for a tilt of 90 degrees. Note the pronounced cross-coupling from yaw to pitch velocity.

These simulations show the feasiblity of the model to predict OKN and OKAN for various orientations.

3.9 Discussion and Extension of the Research

In this dissertation a procedure was developed for identifying a three dimensional model of velocity storage in the vestibulo- ocular reflex. In particular we have found a closed form solution for the system matrix in terms of its eigenvalues and eigenvectors. The angle of the yaw axis eigenvector from the spatial vertical was found to be significant in determining the system matrix.

We determined the form of the system matrix by utilizing the correct coordinate frames. In particular, by taking advantage of the fact that in the eigenvector basis, the system matrix is diagonal and changing its eigenvalues is then straightforward. In addition, because two of the eigenvectors rotated with the head, expressing the system matrix in head coordinates greatly simplified the form of the system matrix.

The parameters of the system matrix were obtained from the data by a nonlinear least squares method. The Marquardt algorithm (Marquardt, 1963) was modified to accomodate vector valued functions so that the eigenvalues and eigenvectors could be calculated easily.

Other methods of nonlinear least squares were considered including the Gauss-Newton algorithm and secant methods (Dennis & Schnabel, 1983). Several factors made the Marquardt method superior to the damped Gauss Newton method. When the Gauss- Newton step is too long the Marquardt step is near the gradient descent direction and is often better than the Gauss-Newton step. Also the Marquardt algorithm is well

defined when the Jacobian matrix (of partial derivatives) does not have full column rank whereas the Gauss-Newton method is not.

The secant method (Dennis, Gay & Welsh, 1981) performs better than the Marquardt method on small residual problems that are close to being linear (Dennis & Schnabel, 1983). However, even in these cases, the Marquardt algorithm has fewer lines of code and is less complex.

Updates that have been made to the Marquardt algorithm have mostly improved its robustness (Powell, 1975; Osborne, 1976; More, 1977). For our problem we are not concerned with finding any local minimum so we cannot take advantage of this feature. We have instead concentrated on choosing appropriate initial parameter vectors in order to converge to the best solution.

The extended Marquardt procedure was applied to cross-coupled data from monkeys. The procedure confirmed that the yaw axis eigenvector stayed approximately spatially invariant. It also showed that for small tilts the yaw axis eigenvector shifted in the direction opposite to the tilt of the head corresponding to human perception of the spatial vertical.

The program which was developed to analyze data from monkeys and find the eigenvectors and eigenvalues was shown to be a useful tool in studies of the three dimensional characteristics of the vestibulo-oculo reflex (Dai et al, 1990). Further experimental work is required to compare the results found for arbitrary orientations to data. Further modelling is also essential to account for the up down assymetries and other non-

linearities not considered here. This should show how these phenomena impact on the organizational structure of the vestibulo-ocular reflex. In addition, the fitting procedures can be improved by considering spontaneous nystagmus and secondary nystagmus.

Since the procedure developed in this dissertation is model based, secondary nystagmus was not considered. The model is a first order linear dynamical system and therefore contains modes of behavior which do not predict secondary nystagmus. A surprising feature of the method is the closeness of the fit despite the simplicity of the model. In addition, the effects of visual suppression were not dealt with since only the structure of velocity storage was determined as a function of gravity. Further modelling and behavioral studies should elucidate the effects of these phenomena. To determine a better estimate of primary nystagmus, one could add exponentials to the closed form solutions and then compare the data to the augmented functions. This would correspond to a higher order system.

The effects of microgravity on OKN and OKAN have not been determined conclusively. Assuming our hypothesis that it is gravity which modifies the yaw axis eigenvector, we can conclude that in the absence of such a force, there will be no cross-coupling. This needs to be explored further experimentally.

In summary, we have developed a computational procedure to identify the parameters of the system matrix associated with velocity storage in three dimensions. This should help in further studies of the vestibular system.

APPENDIX A

Matrices and Linear Operators:

Suppose {e1,e2,...,en} is a basis of a vector space V over a field K and, for v V, suppose v = a1v1 + a2e2 + ... + anen. Then the coordinate vector of v relative to {ei}, which we write as a column
vector is:

$$[v] = \begin{matrix} a1 \\ a2 \\ .. \\ an \end{matrix}$$

Matrix representation of a linear operator:

Let T be a linear operator on a vector space V over a field K and suppose {ei} is a basis of V. Now T(e1),...,T(en) are vectors in V and so each is a linear combination of the elements of the basis {ei}.

$$\begin{aligned} T(e1) &= a11e1 + a12e2 + ... + a1nen \\ T(e2) &= a21e1 + a22e2 + ... + a2nen \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ T(en) &= an1e1 + an2e2 + ... + annen \end{aligned}$$

Definition: The transpose of the above matrix of coefficients,

is called the matrix representation of T relative to the basis {ei}:

$$[T] = \begin{matrix} a11 & a21 & \dots & an1 \\ a12 & a22 & \dots & an2 \\ \dots & \dots & \dots & \dots \\ a1n & a2n & \dots & ann \end{matrix}$$

Change of Basis:

Let {ei} be a basis of V, and let {fi} be another basis. Suppose,

$$\begin{aligned} f1 &= a11e1 + a12e2 + \dots + a1nen \\ f2 &= a21e1 + a22e2 + \dots + a2nen \\ &\dots\dots\dots\dots\dots\dots\dots\dots \\ fn &= an1e1 + an2e2 + \dots + annen \end{aligned}$$

The transpose of the above matrix of coefficients is the transition matrix from the old basis {ei} to the new basis {fi}:

$$[P] = \begin{matrix} a11 & a21 & \dots & an1 \\ a12 & a22 & \dots & an2 \\ \dots & \dots & \dots & \dots \\ a1n & a2n & \dots & ann \end{matrix}$$

$Pf1 = a11e1 + \dots + a1nen$

Since the columns of P are images of fi, P is a matrix with respect to the basis{fi}. So P takes a vector wrt {fi} to the

same vector wrt {ei}. Note: since the vectors {fi} are linearly independent, the matrix P is invertible. In fact, its inverse P^{-1} is the transition matrix from the basis {fi} back to the basis {ei}.

Theorem: Let P be the transition matrix from a basis {ei} to a basis {fi} in a vector space V. Then, for any vector $v \in V$, $P[v]_f = [v]_e$ Hence $[v]_f = P^{-1}[v]_e$ Even though P is called the transition matrix from the old basis {ei} to the new basis {fi}, its effect is to transform the coordinates of a vector in the new basis {fi} back to the coordinates in the old basis {ei}.

The next theorem shows how matrix representations of linear operators are affected by a change of basis.

Theorem: Let P be the transition matrix from a basis {ei} to a basis {fi} in a vector space V. Then for any linear operator T on V, $[T]_f = P^{-1}[T]_eP$. Similarity: Suppose A and B are square matrices for which there exists an invertible matrix P such that $B = P^{-1}AP$. Then B is said to be similar to A or is said to be obtained from A by a similarity transformation. A linear operator T is said to be diagonalizable if for some basis {ei} it is represented by a diagonal matrix; the basis {ei} is then said to diagonalize T.

Theorem: Let A be a matrix representation of a linear operator T. Then T is diagonalizable iff there exists an invertible matrix P such that $P^{-1}AP$ is a diagonal matrix. That is, T is diagonalizable iff its matrix representation can be diagonalized by a similarity transformation.

Eigenvalues and Eigenvectors:

Let T: V --> V be a linear operator on a vector space V over a field K. A scalar $\lambda \in K$ is called an eigenvalue of T if there exists a nonzero vector $v \in V$ for which

$$T(v) = kT(v) = k(\lambda v) = \lambda(kv)$$

The set of all such vectors is a subspace of V called the eigenspace of λ.

If A is an n-square matrix over K, then an eigenvalue of A means an eigenvalue of A viewed as an operator on K. $\lambda \in K$ is an eigenvalue of A if, for some nonzero (column) vector $v \in K$,

$$Av = \lambda v$$

In this case v is an eigenvector of A belonging to λ.

Theorem: An n-square matrix A is similar to a diagonal matrix B iff A has n linearly independent eigenvectors. In this case the diagonal elements of B are the corresponding eigenvalues. In this case, if we let P be the matrix whose columns are the n independent eigenvectors of A, then $B = P^{-1}AP$

Definition: The characteristic polynomial of A: det (tI - A)

Theorem: Similar matrices have the same characteristic polynomial.

Proof: Suppose A and B are similar matrices, say $B = P^{-1}AP$ where P is invertible. We show that A and B have the same

characteristic polynomial

$$tI = P^{-1} tIP$$

$$\begin{aligned} |tI - B| &= |tI - P^{-1}AP| = |P^{-1} tIP - P^{-1}AP| \\ &= |P^{-1} (tI-A)P| = |P^{-1}| \, |tI-A| \, |P| \\ &= |tI - A| \end{aligned}$$

APPENDIX B

Euler Angles:

In order to specify the orientation of a subject in space we use Euler angles. These angles describe three successive rotations in order to transform form one coordinate system to another. We first rotate the xyz set of axes, by an angle phi counterclockwise about the z axis. Then, we rotate the new set of axes about the new x axis counterclockwise by an angle of theta , to produce yet another set of axes. Finally, we rotate about the new z, counterclockwise by the angle psi . Therefore the angles phi, theta, and psi completely specify the transformation. We express this transformation as the product of three rotations, each of which is a matrix.

A = BCD, where

$$D = \begin{vmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

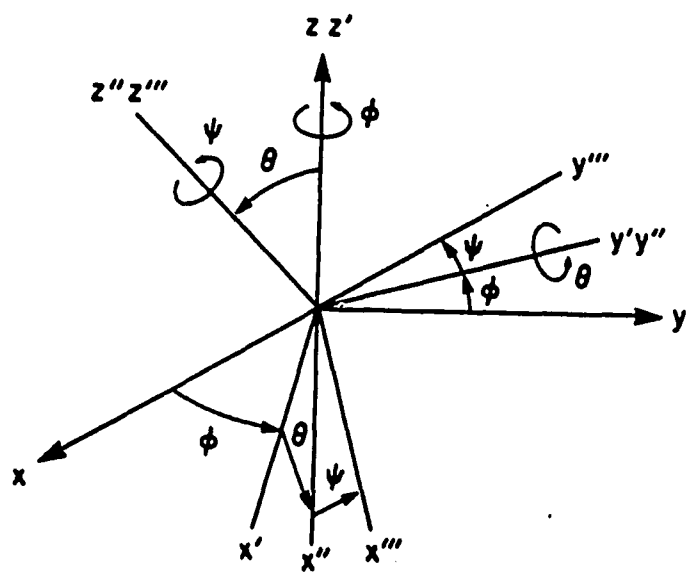
$$C = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{vmatrix}$$

$$B = \begin{vmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

The product A = BCD:

$$\begin{vmatrix} \cos\psi\cos\phi - \cos\theta\sin\phi\sin\psi & \cos\psi\sin\phi + \cos\theta\cos\phi\sin\psi & \sin\psi\sin\theta \\ -\sin\psi\cos\phi - \cos\theta\sin\phi\cos\psi & -\sin\psi\sin\phi + \cos\theta\cos\phi\cos\psi & \cos\psi\sin\theta \\ \sin\theta\sin\phi & -\sin\theta\cos\phi & \cos\theta \end{vmatrix}$$

This particular transformation is a convention, and is somewhat arbitrary. We choose this system because it coincides with our experimental equipment.



APPENDIX C

COMPUTER PROGRAM USED TO IMPLEMENT THE 3-D MODEL

```
C
C  /*                    3D MODEL                                    */
C  /* DEFINE THE ROTATION MATRIX, GIVEN THE 3 ANGLES                */
C  /*  PHI,THETA, AND PSI (EULER ANGLES)                            */
C  /* THEN GIVEN THE INITIAL H MATRIX FOR THE UPRIGHT               */
C  /* CASE, CALCULATE THE NEW H, AFTER ROTATION                     */
C  /*                                                               */
C  /* THE NEW SYSTEM MATRIX ROTATES 2 OF THE                        */
C  /* EIGEN VECTORS, BUT THE BODY VERTICAL                          */
C  /* MAINTAINS SPATIAL VERTICAL - THAT IS IT DOES NOT              */
C  /* ROTATE AT ALL, OR VERY SLIGHTLY.                              */
C
C  /*  SOME ASSUMPTIONS:  WE ASSUME THAT THE SUBJECTI               */
C  /*  IS POSITIONED SO THAT THE X-AXIS COMES OUT OF                */
C  /*  LEFT EAR, THE Y-AXIS OUT OF THE BACK OR HIS HEAD             */
C  /*  AND THE Z-AXIS OUT OF THE TOP OF HIS HEAD.                   */
C  /*  FURTHERMORE, WE ASSUME THAT POSITIVE ROTATIONS               */
C  /*  ARE TO THE LEFT, WHEREAS NEGATIVE ROTATIONS ARE              */
C  /*  TO THE RIGHT.                                                */
C
C
      REAL              ROT(3,3)
      REAL              HORIG(3,3)
C     /* INVERSE ROT */
      REAL              IROT(3,3)
      REAL              E1(3)
      REAL              E2(3)
      REAL              E3(3)
      REAL              H(3,3)
C     /* FOR NEG ROTS */
      REAL              HNEG(3,3)
      REAL              P(3,3)
      REAL              PINV(3,3)
      REAL              R(3,3)
      REAL              RINV(3,3)
      REAL              TCAN(3,3)
      REAL              TCANI(3,3)
      REAL              HCAN(3,3)
      REAL              VISN(3,3)
C     /* IN CANAL COORD */
      REAL              VISNC(3,3)
```

```
      REAL              RETSLP(3)
      REAL              EYEVL(3)
      REAL              TEMP(3,3)
      REAL              TEMPV(3)
      REAL              HPRIME(3,3)
      REAL              NEWH(3,3)
      REAL              ROTLST(3,3)
C     /* INTEGR STATE */
      REAL              INTSTE(3)
C     /* DERIVATIVE OF X  */
      REAL              XDOT(3)
C     /* INITIAL VALUE FOR STATE */
      REAL              STINIT(3)
C     /* STATE IN CANAL COORDINATES */
      REAL              INTCAN(3)
C     /* SURROUND VELOCITY */
      REAL              SURVEL(3)
C      /* DRUM ORIENTATION */
      REAL              ALPHA
      REAL              BETA
      REAL              GAMMA
C     /* STATE OF LIGHT */
      INTEGER           LIGHT
      REAL              TIMINC
      REAL              XOUT(601,3)
      REAL              YOUT(601,3)
      REAL              HGOUT(240,3)
      REAL              PI
      INTEGER           I
      INTEGER           J
      INTEGER           K
      REAL              PHI
      REAL              PSI
      REAL              THETA
C
      INTEGER           INCRS
      INTEGER           ITIMFRS
C
      INTEGER           ITERS
      INTEGER           LENGTH
      REAL              INCAMT
      REAL              TLTDEG
      REAL              TLTINC
      REAL              MINTIM
      REAL              MAXTIM
      REAL              EVMVP
```

```
      REAL              EVMHP
      REAL              VRTLTP
C     /* MAX EIGENVALUE VERTICAL */
      REAL              MXEVVP
      REAL              MNEVVP
      REAL              MXEVHP
      REAL              MNEVHP
C     /* MAX VERTICAL EIGENVECTOR TILT */
      REAL              MXVTLP
      REAL              MNVTLP
C   /* EIGENVALUES AND VECTORS FOR NEGATIVE ROTATIONS */
      REAL              EVMDVN
      REAL              EVMDHN
      REAL              VRTLTN
C     /* MAX EIGENVALUE VERTICAL */
      REAL              MXEVVN
      REAL              MNEVVN
      REAL              MXEVHN
      REAL              MNEVHN
C     /* MAX VERTICAL EIGENVECTOR TILT */
      REAL              MXVTLN
      REAL              MNVTLN
      REAL              XMXTIM
      REAL              XMNTIM
      REAL              ZMXTIM
      REAL              ZMNTIM
      REAL              XMXVAL
      REAL              ZMXVAL
      REAL              XMNVAL
      REAL              ZMNVAL
      COMMON PHI,THETA,PSI,R,RINV,P,PINV,TCAN,TCANI,HCAN
      COMMON/A/VISN,TIMINC
      OPEN (5,FILE= ' ')
      OPEN (6,FILE= ' ')
      OPEN (7,FILE= ' ')
C
C
C
C /* INITIALIZATION - READ IN THE GLOBAL PARAMETERS */
C
C /* READ IN THE ORIGINAL SYSTEM MATRIX WITH THE
EIGENVALUES */
C /* FOR THE UPRIGHT POSITION
*/
  1   FORMAT (G10.3)
  2   FORMAT (G10.3,G10.3)
```

```
      READ (5,3) ((HORIG(I,J),J=1,3),I=1,3)
  3   FORMAT (G10.3,G10.3,G10.3)
  4   FORMAT (I2)
C
C /* READ IN THE VISUAL COUPLING MATRIX */
      READ (5,3) ((VISN(I,J),J=1,3),I=1,3)
C
C
C  /* READ IN MAX AND MIN EIGENVALUES FOR POS ROTATIONS */
      READ (5,2) MXEVVP,MNEVVP
      READ (5,2) MXEVHP,MNEVHP
C /* READ IN MAX AND MIN OF VERTICAL EIGENVECTOR TILT FOR
POS ROT */
      READ (5,2) MXVTLP,MNVTLP
C  /* READ IN MAX AND MIN EIGENVALUES FOR NEG ROTATIONS */
      READ (5,2) MXEVVN,MNEVVN
      READ (5,2) MXEVHN,MNEVHN
C /* READ IN MAX AND MIN OF VERTICAL EIGENVECTOR TILT FOR
NEG ROT */
      READ (5,2) MXVTLN,MNVTLN
      READ (5,4) ITIMFRS
C
C
C /* FORM TCAN - THE TRANSFORMATION FROM HEAD TO CANAL
COORDINATES */
C
      CALL CRROT (TCAN,0.0,-15.0,45.0)
      CALL TRSPSE (TCAN,TCANI,3)
      WRITE(6, 1000)
 1000   FORMAT(/,' ','TCAN: TRANSFORMATION TO CANAL
COORDINATES' )
      CALL PRMAT (TCAN)
      WRITE(6, 1010)
 1010   FORMAT(/,' ','TCANI: TRANSFORM. FROM CANAL
COORDINATES' )
      CALL PRMAT (TCANI)
C
C
C
      WRITE(6, 1020)   MNEVVP
 1020   FORMAT(/,' ','MINIMUM POS VERTICAL EIGENVALUE IS
',F10.3 )
      WRITE(6, 1030)   MXEVVP
 1030   FORMAT(/,' ','MAXIMUM POS VERTICAL EIGENVALUE IS
',F10.3 )
      WRITE(6, 1040)   MNEVHP
 1040   FORMAT(/,' ','MINIMUM POS HORIZONTAL EIGENVALUE IS
```

```
',F10.3 )
      WRITE(6, 1050)   MXEVHP
 1050   FORMAT(/,' ','MAXIMUM POS HORIZONTAL EIGENVALUE IS
',F10.3 )
      WRITE(6, 1060)   MNVTLP
 1060   FORMAT(/,' ','MINIMUM POS VERTICAL TILT IS ',F10.3 )
      WRITE(6, 1070)   MXVTLP
 1070   FORMAT(/,' ','MAXIMUM POS VERTICAL TILT IS ',F10.3 )
      WRITE(6, 1080)   MNEVVN
 1080   FORMAT(/,' ','MINIMUM NEG VERTICAL EIGENVALUE IS
',F10.3 )
      WRITE(6, 1090)   MXEVVN
 1090   FORMAT(/,' ','MAXIMUM NEG VERTICAL EIGENVALUE IS
',F10.3 )
      WRITE(6, 1100)   MNEVHN
 1100   FORMAT(/,' ','MINIMUM NEG HORIZONTAL EIGENVALUE IS
',F10.3 )
      WRITE(6, 1110)   MXEVHN
 1110   FORMAT(/,' ','MAXIMUM NEG HORIZONTAL EIGENVALUE IS
',F10.3 )
      WRITE(6, 1120)   MNVTLN
 1120   FORMAT(/,' ','MINIMUM NEG VERTICAL TILT IS ',F10.3 )
      WRITE(6, 1130)   MXVTLN
 1130   FORMAT(/,' ','MAXIMUM NEG VERTICAL TILT IS ',F10.3 )
C
      WRITE(6, 1140)
 1140   FORMAT(/,'0'        )
      WRITE(6, 1150)
 1150   FORMAT(/,' ','H: ORIGINAL SYSTEM MATRIX' )
      CALL PRMAT (HORIG)
      WRITE(6, 1160)
 1160   FORMAT(/,' ','VISN: VISUAL COUPLING MATRIX' )
      CALL PRMAT (VISN)
C
      READ (5,3) PHI,THETA,PSI
      WRITE(6, 1170)
 1170   FORMAT(/,' ','EULER ANGLES, PHI, THETA, PSI: ')
      WRITE(6, 1180)   PHI, THETA,PSI
 1180   FORMAT(/,' ',F5.0,F5.0,F5.0 )
      READ (5,3) SURVEL
      READ (5,4) LENGTH
      READ (5,1) TIMINC
      ITERS = LENGTH/TIMINC
      WRITE(6, 1190)   TIMINC
 1190   FORMAT(/,' ','TIME INCREMENT IS ',F10.3 )
      WRITE(6, 1200)     LENGTH
 1200   FORMAT(/,' ','LENGTH OF EXPERIMENT IS ',I2,'
```

```
SECONDS' )
      EVMVP = ABS(MNEVVP - MXEVVP) *THETA/90.0
      VRTLTP = ABS(MNVTLP - MXVTLP) *THETA/90.0
      EVMHP = ABS(MNEVHP - MXEVHP) *THETA/90.0
C
      WRITE(6, 1210)   VRTLTP
 1210   FORMAT(/,' ','VERTICAL EVECTOR FOR POS ROT IS
TILTED',F3.0 )
      EVMDVN = ABS(MNEVVN - MXEVVN) *THETA/90.0
      VRTLTN = ABS(MNVTLN - MXVTLN) *THETA/90.0
      EVMDHN = ABS(MNEVHN - MXEVHN) *THETA/90.0
C
      WRITE(6, 1220)   VRTLTN
 1220   FORMAT(/,' ','VERTICAL EVECTOR FOR NEG ROT IS
TILTED',F3.0 )
C
C
      WRITE(6, 1230)
 1230   FORMAT(/,'1'        )
      WRITE(6, 1240)     THETA
 1240   FORMAT(/,' ','TILT OF',F5.0,' DEGREES' )
      CALL MATEQ (HPRIME,HORIG)
C /* MAKE SOME ASSUMPTIONS FOR UPRIGHT CASE */
      CALL MATEQ (H,HORIG)
C /* CHECK IF UPRIGHT - IF SO SKIP ALOT OF PROCESSING */
      IF (.NOT.(ABS(PHI) + ABS(THETA) + ABS(PSI) .NE.0 ))
GOTO       490
C     /* FORM THE SYSTEM MATRIX H */
      CALL CRRTH (H,ROT,VRTLTP,EVMVP,EVMHP,HPRIME)
      CALL MATEQ (HPRIME,HORIG)
      CALL CRRTH (HNEG,ROT,VRTLTN,EVMDVN,EVMDHN,HPRIME)
  490 CONTINUE
C    /* INPUT CHARGED */
      CALL VECTEQ (STINIT,SURVEL)
C /* PUT INITIAL STATE INTO CANAL COORDINATES */
C /* CALL MULTMV (TCAN,SURVEL,INTCAN);  */
C /* GVET VERSION IN CANAL COORDINATES */
C /* CALL MATMUL (TCAN,VISN,VISNC);  */
C /* CALL MULTMV (VISNC,INTCAN,INTSTE); */
      DO        5 I = 1,3
      XOUT (1,I) = 0
      YOUT (1,I) = 0
    5 CONTINUE
C /* CALL MULTMV (TCANI,INTSTE,TEMPV);  */
C     /* TEMPV;  */
C
```

```
C /* GO THROUGH THE TIME FRAMES, CALLING THE MODEL AT */
C /* EVERY ITERATION.                                  */
      DO       10 I = 1,ITIMFRS
      READ (5,4) LIGHT
      WRITE(6, 1250)   LIGHT
 1250   FORMAT(/,' ','LIGHT STATUS IS: ',I2 )
      CALL MODEL (H,HNEG,STINIT,LIGHT,EYEVL,INTSTE)
C   /* STINIT = EYEVL;*/
      DO       15 J = 1,3
      STINIT (J) = EYEVL (J)
   15 CONTINUE
   10 CONTINUE
      RETURN
      END
C
```

```
      SUBROUTINE CRRTH(H,ROT,VRTTLT,EVMODV,EVMODH,HPRIME)
C /*  PROCEDURE TO CREATE THE SYSTEM H MATRIX IN     */
C /*  CANAL COORDINATES                              */
      REAL              H(3,3)
      REAL              ROT(3,3)
      REAL              ROTLST(3,3)
      REAL              HPRIME(3,3)
      REAL              VRTTLT
      REAL              EVMODV
      REAL              EVMODH
      REAL              NEWH(3,3)
      REAL              TEMP(3,3)
      COMMON
     PHI,THETA,PSI,R(3,3),RINV(3,3),P(3,3),PINV(3,3),
     X       TCAN(3,3),TCANI(3,3),HCAN(3,3)
C
      CALL CRROT (ROT,PHI,THETA,PSI)
C  /*  R IS THE ROTATION MRIX WHICH PUT US IN THE MONKEY'S
     */
C  /*  COORDINATE SYSTEM */
      CALL MATEQ (R,ROT)
C     /* INVERSE OF R */
      CALL INV (R,RINV)
C
      CALL CRROT (ROTLST,PHI,VRTTLT,PSI)
      CALL CRRTP (P,ROT,ROTLST)
C     /* P(*,3) = ROTLST (*,3);*/
C    /* ROTATE LAST COL SLIGHTLY */
      WRITE(6, 1260)
 1260   FORMAT(/,'0'        )
      WRITE(6, 1270)
 1270   FORMAT(/,' ','ROTATION MATRIX' )
      CALL PRMAT (ROT)
      WRITE(6, 1280)
 1280   FORMAT(/,'0'        )
      WRITE(6, 1290)
 1290   FORMAT(/,' ','P: EIGENVECTOR MATRIX' )
      CALL PRMAT (P)
      WRITE(6, 1300)
 1300   FORMAT(/,'0'        )
      CALL INV (P,PINV)
      CALL MATMUL (RINV,P,TEMP)
      WRITE(6, 1310)
 1310   FORMAT(/,' ','RINV * P' )
      CALL PRMAT (TEMP)
      HPRIME (1,1) = HPRIME (1,1) + EVMODV
```

```
      HPRIME (3,3) = HPRIME (3,3) - EVMODH
      WRITE(6, 1320)
 1320   FORMAT(/,' ','HPRIME: EIGENVALUE MATRIX AFTER
ROTATION' )
      CALL PRMAT (HPRIME)
      WRITE(6, 1330)
 1330   FORMAT(/,'0'        )
C
      CALL MATMUL (HPRIME,PINV,TEMP)
      CALL MATMUL (P,TEMP,NEWH)
      WRITE(6, 1340)
 1340   FORMAT(/,' ','NEWH: SYSTEM H MATRIX AFTER ROTATION'
)
      CALL PRMAT (NEWH)
C  /* NOW PUT H MATRIX INTO MONKEY COORDINATE SYSTEM */
C  /* NEWH = RINV*NEWH*R */
C
      CALL MATMUL (NEWH,R,TEMP)
      CALL MATMUL (RINV,TEMP,NEWH)
C
      CALL MATEQ (H,NEWH)
      DO 5 I = 1,3
        DO 6 J = 1,3
          IF (ABS(H(I,J)).LT.0.001)H(I,J)=0.
    6   CONTINUE
    5 CONTINUE
      WRITE(6, 1350)
 1350   FORMAT(/,' ','H: SYSTEM H MATRIX IN MONKEY FRAME' )
      CALL PRMAT (H)
C
C  /*  PUT THE SYSTEM MATRIX INTO CANAL COORDINATES */
      CALL MATMUL (TCAN,NEWH,TEMP)
      CALL MATMUL (TEMP,TCANI,HCAN)
      WRITE(6, 1360)
 1360   FORMAT(/,' ','HCAN: SYSTEM MATRIX IN CANAL FRAME' )
      CALL PRMAT (HCAN)
      RETURN
      END
C
```

```
      SUBROUTINE CRROT(ROT,PHI,THETA,PSI)
C
C /* CREATE ROTATION MATRIX GIVEN THE 3 EULER ANGLES */
C
      REAL              ROT(3,3)
      REAL              PHI
      REAL              THETA
      REAL              PSI
C
      ROT (1,1) = COSD (PSI)*COSD(PHI) -
     COSD(THETA)*SIND(PHI)*SIND(PSI)
      ROT (2,1) = COSD(PSI)*SIND(PHI) +
     COSD(THETA)*COSD(PHI)*SIND(PSI)
      ROT (3,1) = SIND (PSI) * SIND (THETA)
C
      ROT (1,2) = - SIND(PSI)*COSD(PHI)
     -COSD(THETA)*SIND(PHI)*COSD(PSI)
      ROT (2,2) = - SIND(PSI)*SIND(PHI)
     +COSD(THETA)*COSD(PHI)*COSD(PSI)
      ROT (3,2) = COSD (PSI) * SIND (THETA)
C
      ROT (1,3) = SIND(THETA)*SIND(PHI)
      ROT (2,3) = - SIND (THETA)*COSD(PHI)
      ROT (3,3) = COSD (THETA)
      RETURN
      END
C
      SUBROUTINE CRRTP(P,ROT,ROTLST)
C
C /* CREATE THE P MATRIX - ROTATED EIGENVECTOR MATRIX */
C /* ROTATING THE VERTICAL DIFFERENTLY THAN OTHER 2
*/
C
      REAL              P(3,3)
      REAL              ROT(3,3)
      REAL              ROTLST(3,3)
C
C  /* P(*,1) = ROT (*,1); */
C  /* P(*,2) = ROT (*,2); */
C  /* P(*,3) = ROTLST (*,3);*/
      DO        20 I = 1,3
      P (I,1) = ROT (I,1)
      P (I,2) = ROT (I,2)
      P (I,3) = ROTLST (I,3)
   20 CONTINUE
      RETURN
      END
```

```
C
      FUNCTION DET33(A)
C
C  /*  CALCULATE THE DETERMINANT OF THE GIVEN 3 X 3 MATRIX
*/
C
      REAL              A(3,3)
C
      DET33   =  (A(1,1)*A(2,2)*A(3,3) +
     X           A(1,2)*A(2,3)*A(3,1) +
     X           A(1,3)*A(2,1)*A(3,2) -
     X           A(1,3)*A(2,2)*A(3,1) -
     X           A(1,2)*A(2,1)*A(3,3) -
     X           A(1,1)*A(2,3)*A(3,2) )
      RETURN
      END
C
      SUBROUTINE HGCONV (XIN,OUT,N,M,K)
      DIMENSION XIN(N,3),OUT(M,3)
      J =1
      DO 10 I =1,N-K,K
        OUT(J,1) = XIN (I,1)
        OUT(J,2) = XIN (I,2)
        OUT(J,3) = XIN (I,3)
        J = J + 1
  10  CONTINUE
      PRINT *,'I,J= ',I,J
      RETURN
      END
```

```
SUBROUTINE INV(A,B)
C
C      /* CALCULATE THE INVERSE OF A AND PUT RESULT IN B */
C
       REAL              A(3,3)
       REAL              B(3,3)
       REAL              C(3,3)
       REAL              DET
C
       DET = DET33 (A)
       IF (.NOT.(DET .EQ. 0 ))  GOTO      500
       WRITE(6, 1370)
 1370   FORMAT(/,' ','DET IS ZERO - INVERSE CANNOT BE
CALCULATED' )
       STOP
C
C
  500 CONTINUE
C      /* FIND THE CLASSIC ADJOINT AND SAVE TEMPORARILY IN
B */
C
       B(1,1) =  A(2,2)*A(3,3) - A(2,3)*A(3,2)
       B(1,2) = -A(2,1)*A(3,3) + A(2,3)*A(3,1)
       B(1,3) =  A(2,1)*A(3,2) - A(2,2)*A(3,1)
C
       B(2,1) = -A(1,2)*A(3,3) + A(1,3)*A(3,2)
       B(2,2) =  A(1,1)*A(3,3) - A(1,3)*A(3,1)
       B(2,3) = -A(1,1)*A(3,2) + A(1,2)*A(3,1)
C
       B(3,1) =  A(1,2)*A(2,3) - A(1,3)*A(2,2)
       B(3,2) = -A(1,1)*A(2,3) + A(1,3)*A(2,1)
       B(3,3) =  A(1,1)*A(2,2) - A(1,2)*A(2,1)
C
C      /* PUT INVERSE IN B BY DIVIDING BY DET */
       CALL TRSPSE (B,C,3)
C      /* B = C;*/
       DO        25 I = 1,3
       DO        30 J = 1,3
       B(I,J) = C(I,J)
       B(I,J) = B(I,J)/DET
   30 CONTINUE
   25 CONTINUE
C
C      /* B = B / DET;*/
       RETURN
       END
```

```
SUBROUTINE INTEGR(X,XDOT,TIMINC)
      REAL              X(3)
      REAL              XDOT(3)
      REAL              TIMINC
      INTEGER           I
      INTEGER           J
C
C
      DO        35 I = 1,3
      X(I) = X(I) + XDOT(I) * TIMINC
C
   35 CONTINUE
      RETURN
      END
```

```
SUBROUTINE MATEQ (A,B)
      REAL              A(3,3)
      REAL              B(3,3)
      INTEGER           I
      INTEGER           J
      DO       40 I = 1,3
      DO       45 J = 1,3
      A (I,J) = B(I,J)
   45 CONTINUE
   40 CONTINUE
      RETURN
      END


SUBROUTINE MATMUL(A,B,C)
      REAL              A(3,3)
      REAL              B(3,3)
      REAL              C(3,3)
      REAL              SUM
      INTEGER           I
      INTEGER           J
      INTEGER           K
C
      N = 3
      DO        50 I = 1,N
      DO        55 J = 1,N
      SUM = 0
      DO        60 K = 1,N
      SUM = SUM + A (I,K) * B (K,J)
   60 CONTINUE
      C (I,J) = SUM
   55 CONTINUE
   50 CONTINUE
      RETURN
      END
C
```

```
SUBROUTINE MAXMIN(A,MAXVAL,MAXTIM,MINVAL,MINTIM)
      REAL              A(3)
      REAL              MAXVAL
      REAL              MINVAL
      REAL              MAXTIM
      REAL              MINTIM
      INTEGER           I
      INTEGER           N
C
      MAXVAL = A (1)
      MINVAL = A (1)
      MAXTIM = 0
      MINTIM = 0
      N = 3
C
      DO        65 I = 1,N
      IF (.NOT.(A(I) .GT. MAXVAL ))  GOTO       510
      MAXTIM = I
      MAXVAL = A (I)
  510 CONTINUE
      IF (.NOT.(A(I) .LT. MINVAL ))  GOTO       520
      MINTIM = I
      MINVAL = A (I)
  520 CONTINUE
   65 CONTINUE
      RETURN
      END
C
```

```
SUBROUTINE MODEL(HPOS,HNEG,SURVEL,LIGHT,EYEVL,INTSTE)
      REAL              SURVEL(3)
      REAL              HPOS(3,3)
      REAL              HNEG(3,3)
      REAL              EYEVL(3)
      REAL              INTSTE(3)
      REAL              H(3,3)
      REAL              RV(3)
      REAL              VF(3)
      REAL              VN(3)
      REAL              HV(3)
      REAL              RETSLP(3)
      REAL              XDOT(3)
      REAL              TEMPV(3)
      REAL              XOUT(601,3)
      REAL              YOUT(601,3)
      REAL              HGOUT(120,3)
      REAL              G1
      INTEGER           I
      INTEGER           LIGHT
      COMMON/A/VISN(3,3),TIMINC
C
      G1 = 0.55
      DO 4 I = 1,3
        RV (I) = 0
   4  CONTINUE
      WRITE(6, 1000)   SURVEL
 1000   FORMAT(/,' ','SURVEL= ', F10.3 )
C  /* XDOT = N(E) + HX */
      IF (.NOT.(LIGHT .EQ. 1 ))  GOTO      490
      CALL MULTMV (VISN,SURVEL,XDOT)
      DO         5 I = 1,3
      INTSTE(I) = 0
C     /* INITIAL CASE */
      VF(I) = (G1 * SURVEL(I)) / (1 + G1)
      EYEVL(I) = VF(I)
    5 CONTINUE
  490 CONTINUE
      IF (.NOT.(LIGHT .EQ. 0 ))  GOTO      500
      DO        10 I = 1,3
      XDOT(I) = SURVEL(I)
      EYEVL(I) = INTSTE(I) + RV(I)
   10 CONTINUE
  500 CONTINUE
      DO        15 J = 1,3
      RV(J) = 0
```

```
   15 CONTINUE
      ITERS = 601
C  /* XOUT (1,*) = INTSTE; */
      DO        20 I = 1,3
      XOUT (1,I) = INTSTE (I)
      YOUT (1,I) = EYEVL(I)
   20 CONTINUE
C
      DO        25 I = 2,ITERS
      DO        30 J = 1,3
      VN(J) = INTSTE(J) + RV(J)
      EYEVL(J) = VN(J)
   30 CONTINUE
      CALL MATEQ (H,HPOS)
      CALL INTEGR (INTSTE,XDOT,TIMINC)
C     /* VERTICAL COMPONENT */
      IF (.NOT.(INTSTE(1) .LT. 0  ))  GOTO      510
C     /* CHANGE DYNAMICS */
      CALL MATEQ (H,HNEG)
  510 CONTINUE
      CALL MULTMV (H,INTSTE,TEMPV)
      CALL VECTEQ (XDOT,TEMPV)
      IF (.NOT.(LIGHT .EQ. 1 ))  GOTO      520
      DO        35 J = 1,3
      RETSLP(J) = SURVEL(J) - VF(J) - INTSTE(J)
      VF(J) = G1 * RETSLP(J)
      EYEVL(J) = VN(J) + VF(J)
C
   35 CONTINUE
      CALL MULTMV (VISN,RETSLP,TEMPV)
      DO        40 J = 1,3
      XDOT(J) = XDOT(J) + TEMPV(J)
   40 CONTINUE
  520 CONTINUE
      DO        45 J = 1,3
      YOUT(I,J) = EYEVL(J)
      XOUT(I,J) = VN(J)
   45 CONTINUE
   25 CONTINUE
      CALL PRDIAG (XOUT,YOUT)
C     CALL PRMXMN
      CALL HGCONV (YOUT,HGOUT,601,120,5)
      CALL PRTVEL (HGOUT,120,1,TIMINC)
      RETURN
      END
C
```

```
      SUBROUTINE MULTMV(H,X,Y)
C
C /* MULTIPLY MATRIX H BY VECTOR X AND PUT RESULTING VECTOR
IN Y */
C
            REAL              H(3,3)
            REAL              X(3)
            REAL              Y(3)
            INTEGER           I
            INTEGER           J
C
            DO 111 I = 1,3
             Y(I) = 0
        111 CONTINUE
            DO       110 I = 1,3
            DO       115 J = 1,3
            Y (I) = Y (I) + H(I,J) * X (J)
        115 CONTINUE
        110 CONTINUE
            RETURN
            END
C
```

```
SUBROUTINE PRDIAG(XOUT,YOUT)
       REAL  XOUT (601,3)
       REAL  YOUT (601,3)
      WRITE(6, 1390)
 1390   FORMAT(/,' ','STATEX',  'STATEY','STATEZ' )
      WRITE(6, 1400)     XOUT(1,1),  XOUT(1,2),  XOUT(1,3)
 1400   FORMAT(/,' ','0 SECS:   ',F10.3,F10.3,F10.3 )
      WRITE(6, 1410)     XOUT(2,1),  XOUT(2,2),  XOUT(2,3)
 1410   FORMAT(/,' ','.05 SECS: ',F10.3,F10.3,F10.3 )
      WRITE(6, 1420)     XOUT(61,1),  XOUT(61,2),
XOUT(61,3)
 1420   FORMAT(/,' ','3 SECS:   ',F10.3,F10.3,F10.3 )
      WRITE(6, 1430)     XOUT(101,1),  XOUT(101,2),
XOUT(101,3)
 1430   FORMAT(/,' ','5 SECS:   ',F10.3,F10.3,F10.3 )
      WRITE(6, 1440)     XOUT(251,1),  XOUT(251,2),
XOUT(251,3)
 1440   FORMAT(/,' ','12.5 SECS:',F10.3,F10.3,F10.3 )
      WRITE(6, 1450)     XOUT(501,1),  XOUT(501,2),
XOUT(501,3)
 1450   FORMAT(/,' ','25 SECS:  ',F10.3,F10.3,F10.3 )
      WRITE(6, 1460)     XOUT(601,1),  XOUT(601,2),
XOUT(601,3)
 1460   FORMAT(/,' ','30 SECS:  ',F10.3,F10.3,F10.3 )
      WRITE(6, 1470)
 1470   FORMAT(/,' ','VELOCX',  'VELOCY',  'VELOCZ','ANALX'
)
      WRITE(6, 1480)     YOUT(1,1),  YOUT(1,2),  YOUT(1,3)
 1480   FORMAT(/,' ','0 SECS:   ',F10.3,F10.3,F10.3 )
      WRITE(6, 1490)     YOUT(2,1),  YOUT(2,2),  YOUT(2,3)
 1490   FORMAT(/,' ','.05 SECS: ',F10.3,F10.3,F10.3 )
      WRITE(6, 1500)     YOUT(61,1),  YOUT(61,2),
YOUT(61,3)
 1500   FORMAT(/,' ','3 SECS:   ',F10.3,F10.3,F10.3 )
      WRITE(6, 1510)     YOUT(101,1),  YOUT(101,2),
YOUT(101,3)
 1510   FORMAT(/,' ','5 SECS:   ',F10.3,F10.3,F10.3 )
      WRITE(6, 1520)     YOUT(251,1),  YOUT(251,2),
YOUT(251,3)
 1520   FORMAT(/,' ','12.5 SECS:',F10.3,F10.3,F10.3 )
      WRITE(6, 1530)     YOUT(501,1),  YOUT(501,2),
YOUT(501,3)
 1530   FORMAT(/,' ','25 SECS:  ',F10.3,F10.3,F10.3 )
      WRITE(6, 1540)     YOUT(601,1),  YOUT(601,2),
YOUT(601,3)
 1540   FORMAT(/,' ','30 SECS:  ',F10.3,F10.3,F10.3 )
      RETURN
      END
```

```
SUBROUTINE PRMAT(A)
      REAL              A(3,3)
      INTEGER           I
      INTEGER           J
      N = 3
      WRITE(6, 1550)
 1550   FORMAT(/,'0'         )
      DO       120 I = 1,N
      WRITE(6, 1560)    A(I,1),A(I,2),A(I,3)
 1560   FORMAT(3F15.10 )
  120 CONTINUE
      RETURN
      END
C
```

```
      SUBROUTINE PRMXMN
C /* PRINT OUT THE MAX AND MIN VALUES FOR THE VELOCITIES
CALCULATED */
C
C     CALL MAXMIN(YOUT(*,1),XMXVAL,XMXTIM,XMNVAL,XMNTIM)
C     CALL MAXMIN(YOUT(*,3),ZMXVAL,ZMXTIM,ZMNVAL,ZMNTIM)
      XMXTIM = XMXTIM * TIMINC
      XMNTIM = XMNTIM * TIMINC
      ZMXTIM = ZMXTIM * TIMINC
      ZMNTIM = ZMNTIM * TIMINC
C
      WRITE(6, 1580)
 1580   FORMAT(/,' ','MAXIMUM AND MINIMUM VALUES ' )
      WRITE(6, 1590)     THETA,XMXVAL
 1590   FORMAT(/,' ','TILT= ',F4.0, ' VELOCITY X REACHED
MAX ',F7.2 )
      WRITE(6, 1600)     XMXTIM,  XMNVAL,XMNTIM
 1600   FORMAT(/,' ',' AT TME ',F5.2,' AND MIN ',F7.2,' AT
TME ',F5.2 )
      WRITE(6, 1610)     THETA,ZMXVAL
 1610   FORMAT(/,' ','TILT= ',F4.0, ' VELOCITY Z REACHED
MAX ',F7.2 )
      WRITE(6, 1620)     ZMXTIM,  ZMNVAL,ZMNTIM
 1620   FORMAT(/,' ',' AT TME ',F5.2,' AND MIN !,F7.2,' AT
TME ',F5.2 )
      RETURN
      END
C
C
```

```
SUBROUTINE PRTVEL(OUT,N,K,TIMINC)
      REAL              DEGREE
      REAL              OUT(N,3)
      REAL              TIME
      REAL              TIMINC
      INTEGER           I
      INTEGER           J
      INTEGER           K
      INTEGER           N
C
      DO       130 I = 1,N
      TIME = (I-1)*TIMINC * K
      WRITE(7, 1630)   TIME, OUT (I,1),OUT(I,2),OUT(I,3)
 1630   FORMAT(/,' ',F5.2,3F12.2 )
  130 CONTINUE
      RETURN
      END
C
```

```
      SUBROUTINE TRSPSE(A,B,N)
      C
      C  /* B = TRSPSE (A)   */
            INTEGER           I
            INTEGER           J
            INTEGER           N
      C
            REAL              A(3,3)
            REAL              B(3,3)
      C
            DO       135 I = 1,N
            DO       140 J = 1,N
            B (I,J) = A (J,I)
        140 CONTINUE
        135 CONTINUE
            RETURN
            END
```

```
SUBROUTINE VECTEQ(V,W)
      REAL              V(3)
      REAL              W(3)
      INTEGER           I
      DO       145 I = 1,3
      V (I) = W (I)
  145 CONTINUE
      RETURN
      END


FUNCTION SIND (X)
      REAL PI
      REAL X
      PI = 3.14191592654
      SIND = SIN ((X*PI)/180.0)
      RETURN
      END
C


FUNCTION COSD (X)
      REAL PI
      REAL X
      PI = 3.141592654
      COSD = SIN (((X+90.0)*PI)/180.0)
      RETURN
      END
```

APPENDIX D

COMPUTER PROGRAM IMPLEMENTING EXTENDED MARQUARDT ALGORITHM

```
C NONLINEAR LEAST SQUARES ESTIMATION OF PARAMETERS
C USING   MODIFIED MARQUARDT-LEVENBERG METHOD
C MODIFIED FROM PROGRAM IN NUMERICAL RECIPES
C
DIMENSION A(6),X(1200),Y(1200),SIG(1200),LISTA(6),
     *   COVAR(6,6),ALPHA(6,6)
      DIMENSION
YP(600),YR(600),YY(600),YPZ(600),YYZ(600),Z(1200)
      DIMENSION DYDA(6),LLISTA(6),LATR(6)
      COMMON SIG
      PRINT *,'    PLEASE ENTER INPUT PARAMETER FILE NAME:
____.PRM'
      OPEN (5, FILE = ' ')
      PRINT *,'    PLEASE ENTER INPUT DATA FILE NAME:
____.DAT'
      OPEN (7, FILE = ' ')
      PRINT *,'    PLEASE ENTER CONVERGENCE OUTPUT FILE
NAME: ____.OUT'
      OPEN (6, FILE = ' ')
      PRINT *,'    PLEASE ENTER PLOTTING OUTPUT FILE NAME:
____.OUT'
      OPEN (8, FILE = ' ')
      PI = 3.1415926536
C   READ IN THE NUMBER OF PARAMETERS
      READ (5,5) N
    5 FORMAT (I3)
C      PRINT *,'N = ',N
      NCA = N
      MA = N
C  read in angle of rotation in degrees, angle of
eigenvector and time constants
      READ (5,10) RHOD
C      PRINT *,'RHO= ',RHOD
      READ (5,10) ETAD
C      PRINT *,'ETA= ',ETAD
      READ (5,10) TC1
C      PRINT *,'TC1= ',TC1
      READ (5,10) TC3
C      PRINT *,'TC3= ',TC3
   10 FORMAT (F10.3)
C      PRINT *,'RHO,ETA,TC1,TC3= '
```

```
C       PRINT *,RHOD,ETAD,TC1,TC3
        THETAD = 90.0 - RHOD + ETAD
        PRINT *,'ANGLE OF INITIAL EIGENVECTOR IS   ',ETAD
        PRINT *,'INITIAL PITCH TIME CONSTANT IS    ',TC1
        PRINT *,'INITIAL YAW TIME CONSTANT IS      ',TC3
C  change to radians
        RHO = (RHOD*PI)/180.0
        ETA = (ETAD*PI)/180.0
        THETA = (THETAD*PI)/180.0
C  compute the components of the vertical eigenvector
(s1,s2,s3)
C    s2 is 0 for all tilt experiments
        S1 = COS(THETA)
        S2 = 0
        S3 = SIN(THETA)
        S = S1/S3
C  READ IN THE NUMBER OF PARAMETERS TO FIT
C       READ (5,5) MFIT
C       PRINT *,'*** MFIT ',MFIT
C  read in the index into parameter list and attributes (1:
fit;0:no fit)
C
        DO 6 I=1,MA
       READ (5,7) LLISTA(I),LATR(I)
C          PRINT *,' *** LLISTA ',LLISTA(I),LATR(I)
      6 CONTINUE
      7 FORMAT (I4,I2)
        READ (5,10) DELTAT
C        PRINT *,'*** DELTAT ',DELTAT
C    read in the calibrations for x,y,z components
        READ (5,30) XCAL,YCAL,ZCAL
     30 FORMAT (3F10.3)
C        PRINT *,' *** CALS ',XCAL,YCAL,ZCAL
C
C  read in the x,y,z components of the velocity vector
C    nop: number of points
C    negative values indicate end of file
C
        I = 1
        NOP = 0
  9000 CONTINUE
        READ (7,30) XVAL,YVAL,ZVAL
C      WRITE (6,30) XVAL,YVAL,ZVAL
        IF(XVAL.LT.0.0) GO TO 9999
        YP(I) = XVAL*XCAL
        YR(I) = YVAL*YCAL
```

```
      YY(I) = ZVAL*ZCAL
      NOP = NOP + 1
      I = I + 1
      GO TO 9000
 9999 CONTINUE
C  assume that there is no roll data - ndata = nop*2  :
total # of points
      NDATA = NOP*2
C  set up the x values - time
      DO 9010 I=1,NDATA
        X(I) = I-1
C       PRINT,' ***X(I) ',I,X(I)
 9010 CONTINUE
C  combine yaw and pitch data into one vector - y (yp then
yy)
      DO 26 I=1,NOP
        Y(I) = YP(I)
        IF(Y(I).LT.0.0) Y(I)=0.0
C     PRINT,' *** Y(I)= ',I,Y(I)
   26 CONTINUE
      K = NOP+1
      L = 2*NOP
      DO 27 I=K,L
        Y(I) = YY(I-NOP)
        IF(Y(I).LT.0.0) Y(I)=0.0
C     PRINT,' *** Y(I)= ',I,Y(I)
   27 CONTINUE
      DO 29 I=1,NDATA
C     WRITE(6,9020) I,Y(I)
 9020 FORMAT (' I,Y(I) ',I5,F10.3)
 29   CONTINUE
C  if there is roll...
C     K = 2*NOP+1
C     L = 3*NOP
C     DO 28 I=K,L
C       Y(I) = YR(I)
C       IF(Y(I).LT.0.0) Y(I)=0.0
C       PRINT,' *** Y(I)= ',I,Y(I)
C  28 CONTINUE
C  create LISTA (permutation list) according to which
parameters to fit
C
      MFIT = 0
      DO 8 I=1,MA
     IF(LATR(I).EQ.1) THEN
        MFIT = MFIT + 1
```

```
            LISTA(MFIT) = LLISTA(I)
          ENDIF
        8 CONTINUE
          J = MFIT + 1
          DO 9 I=1,MA
          IF (LATR(I).EQ.0) THEN
            LISTA(J) = LLISTA(I)
            J = J + 1
          ENDIF
        9 CONTINUE
          DO 18 I=1,MA
C           PRINT *,'LISTA = ',LISTA(I)
       18 CONTINUE
C  SET THE COEFFICIENTS C11 AND C12  AND THE PARAMETERS A(*)
C  X1 IS THE INITIAL VALUE FOR PITCH AND X3 THE INITIAL
VALUE FOR YAW
C
          X1 = YP(1)
          X3 = YY(1)
          C11 = X1 - S*X3
          C13 = S*X3
          C33 = X3
          C31 = 0
C set the parameters
          A(1) = C11
          A(3) = C13
          A(2) = -1.0/TC1
          A(4) = -1.0/TC3
          A(5) = C31
          A(6) = C33
C PRINT THE PARAMETERS COMPUTED
          PRINT *,'INITIAL PARAMETERS: A(I)'
          PRINT *,A(1),A(2),A(3),A(4),A(5),A(6)
C         COMPUTE THE STANDARD DEVIATION
C          CALL SETSIG (Y,NDATA)
C        WRITE(6,13)SIG(1)
       13 FORMAT(' *** SIG= ',F10.3)
C
C  initially set sigma to 1
C
          DO 3 J=1,NDATA
             SIG(J) = 1
    3     CONTINUE
C
C initial call to MRQMIN
C
```

```
      ALAMDA = -0.01
      CALL MRQMIN(X,Y,NDATA,A,MA,LISTA,MFIT,COVAR,ALPHA,
     *  NCA,CHISQ,ALAMDA,DELTAT)
      ITERS = 100
      SAVCHI = CHISQ
      DO 35 I=1,ITERS
         CALL MRQMIN(X,Y,NDATA,A,MA,LISTA,MFIT,COVAR,ALPHA,
     *     NCA,CHISQ,ALAMDA,DELTAT)
      IF(CHISQ.GT.SAVCHI) GOTO 1010
        IF(ABS(CHISQ-SAVCHI).GE. 0.001)GO TO 1000
       WRITE (6,20) A(1),A(2),A(3),A(4),A(5),A(6)
   20 FORMAT (' CONVERGED ',6F10.3)
C
C  compute final value of fit function Z
C
C      CALL SETSIG (Z,Y,NDATA)
C      CHISQ = CHISQ / SIG(1)
      PRINT *,'*** CHISQ= ',CHISQ
      DEGFREE = NDATA - MFIT
      GOODNESS = CHISQ/DEGFREE
      PRINT *,'GOODNESS OF FIT: ',GOODNESS
C      PRINT *,'*** CHISQ= ',CHISQ
C
C compute the angle of the vertical eigenvector computed
(arc cot of s)
C
      S = A(3)/X3
      ANG = 1.0/S
      THETA = ATAN (ANG)
C  convert back to degrees
      THETAD = (THETA*180.0)/PI
C  eta = theta - 90 + rho  (in monkey frame)
      ETAD = THETAD - 90.0 + RHOD
C
      PRINT *,'ANGLE OF COMPUTED EIGENVECTOR IS ',ETAD
C print the pitch and yaw time contants computed (inverse
of eigenvalues)
C
      PRINT *,'PITCH TIME CONSTANT: ',-1.0/A(2)
      PRINT *,'YAW TIME CONSTANT:   ',-1.0/A(4)
C
C  PRINT OUT DATA FOR PLOTTING
C
      DO 50 J=1,NDATA
       CALL FUNCS (X(J),A,YMOD,DYDA,MA,NDATA,DELTAT)
       Z(J)=YMOD
```

```
          X(J)=X(J)*DELTAT
   50 CONTINUE
      DO 60 J=1,NOP
       YPZ(J)=Z(J)
   60 CONTINUE
      DO 65 J=NOP+1,NDATA
       K=J-NOP
       YYZ(K)=Z(J)
   65 CONTINUE
      DO 70 J=1,NOP
       WRITE(8,100) X(J),YP(J),YPZ(J),YY(J),YYZ(J)
  100       FORMAT (5F12.3)
   70 CONTINUE
C  final call to MRQMIN
C          ALAMDA = 0.
C           CALL
MRQMIN(X,Y,NDATA,A,MA,LISTA,MFIT,COVAR,ALPHA,
C     *       NCA,CHISQ,ALAMDA,DELTAT)
C      DO 80 J=1,MA
C         PRINT *,'VARIANCES ',J,COVAR(J,J)
C80    CONTINUE
       RETURN
 1000 CONTINUE
 1010 CONTINUE
       SAVCHI = CHISQ
   35 CONTINUE
C
       RETURN
       END
```

```
      SUBROUTINE COVSRT (COVAR,NCVM,MA,LISTA,MFIT)
      C sort covariance matrix according to LISTA - permutation
      list
            DIMENSION COVAR(NCVM,NCVM),LISTA(MFIT)
      C* TEST
      C       WRITE(6,9000)
      C9000   FORMAT(' ENTERED COVSRT')
      C* END TEST
            MAA=MA-1
            DO 12 J=1,MAA
              JJ=J+1
              DO 11 I=JJ,MA
                COVAR (I,J)=0
         11   CONTINUE
         12 CONTINUE
            MFITT=MFIT-1
            DO 14 I =1,MFITT
              II=I+1
              DO 13 J=II,MFIT
                IF(LISTA(J).GT.LISTA(I))
           *      COVAR(LISTA(J),LISTA(I)) = COVAR (I,J)
                IF(LISTA(J).LE.LISTA(I))
           *      COVAR(LISTA(I),LISTA(J)) = COVAR (I,J)
         13   CONTINUE
         14 CONTINUE
            SWAP = COVAR(1,1)
            DO 15 J=1,MA
              COVAR(1,J)=COVAR(J,J)
              COVAR(J,J)=0
         15 CONTINUE
            COVAR(LISTA(1),LISTA(1)) = SWAP
            DO 16 J=2,MFIT
              COVAR(LISTA(J),LISTA(J))=COVAR(1,J)
         16 CONTINUE
            DO 18 J=2,MA
              JJ=J-1
              DO 17 I=1,JJ
                COVAR (I,J) = COVAR (J,I)
         17   CONTINUE
         18 CONTINUE
      C* TEST
      C       WRITE(6,9010)
      C9010   FORMAT(' EXIT COVSRT')
      C* END TEST
            RETURN
            END
```

```
      SUBROUTINE SETSIG (X,Y,N)
            DIMENSION X(N),Y(N)
              COMMON SIG(1200)
      C      WRITE(6,40) N
        40   FORMAT(' N = ',I3)
            VAR = 0.
            DO 20 I =1,N
           VAR = VAR + (X(I) - Y(I))*(X(I) - Y(I))
         20 CONTINUE
            VAR = VAR / N
            SIGMA = SQRT (VAR)
            DO 30 I = 1,N
              SIG (I) = SIGMA
         30 CONTINUE
            RETURN
            END
      C
```

```
      SUBROUTINE MRQMIN(X,Y,NDATA,A,MA,LISTA,MFIT,
           * COVAR,ALPHA,NCA,CHISQ,ALAMDA,DELTAT)
            DIMENSION X(NDATA),Y(NDATA),A(MA),ALPHA(NCA,NCA),
           *
      BETA(6),DA(6),ONEDA(6,6),LISTA(MA),COVAR(NCA,NCA),ATRY(6)

              COMMON SIG(1200)
            IF (ALAMDA.GE.0) GO TO 1030
              KK = MFIT + 1
              DO 12 J = 1,MA
                IHIT = 0
                DO 11 K = 1,MFIT
                  IF (LISTA(K).EQ.J)IHIT=IHIT+1
         11     CONTINUE
                IF (IHIT.NE.0) GO TO 1020
                  LISTA(KK) = J
                  KK = KK + 1
       1020   CONTINUE
              IF (IHIT.GT.1)
           *         PRINT *,'IMPROPER PERMUTATION IN LISTA'
         12 CONTINUE
              IF (KK.NE.(MA+1)) PRINT *, 'IMPROPER PERMUTATION IN
      LISTA'
              ALAMDA = 0.01
              CALL
      MRQCOF(X,Y,NDATA,A,MA,LISTA,MFIT,ALPHA,BETA,NCA,
           *     CHISQ,DELTAT)
              OCHISQ = CHISQ
              DO 13 J=1,MA
                ATRY (J) = A (J)
         13 CONTINUE
       1030   CONTINUE
              DO 15 J = 1,MFIT
                DO 14 K=1,MFIT
                  COVAR (J,K) = ALPHA (J,K)
         14 CONTINUE
              COVAR (J,J)=ALPHA(J,J)*(1. + ALAMDA)
              ONEDA(J,1) = BETA(J)
         15 CONTINUE
      C
              CALL GAUSSJ (COVAR,MFIT,NCA,ONEDA,1,1)
              DO 19 I=1,MFIT
                DA(I) = ONEDA (I,1)
      C       WRITE (6,8020) DA(I)
       8020 FORMAT ('*** DA ',F10.3)
         19   CONTINUE
              IF (ALAMDA.NE.0) GO TO 1040
```

```
            CALL COVSRT (COVAR,NCA,MA,LISTA,MFIT)
            RETURN
 1040 CONTINUE
          DO 16 J=1,MFIT
            ATRY (LISTA(J)) = ATRY(LISTA(J)) + DA (J)
   16 CONTINUE
C       PRINT,'*** ATRY ',ATRY(1),ATRY(2),ATRY(3),ATRY(4)
C
          CALL MRQCOF (X,Y,NDATA,ATRY,MA,LISTA,MFIT,
     *       COVAR,DA,NCA,CHISQ,DELTAT)
C        WRITE(6,8040) CHISQ
 8040 FORMAT('*** CHITRY ',F10.3)
          IF (CHISQ.GE.OCHISQ) GO TO 1050
C            PRINT,'*** CHILE',CHISQ,OCHISQ
            ALAMDA = 0.1 * ALAMDA
            OCHISQ = CHISQ
            DO 18 J=1,MFIT
              DO 17 K=1,MFIT
                ALPHA (J,K) = COVAR (J,K)
   17 CONTINUE
            BETA (J) = DA (J)
            A(LISTA(J)) = ATRY (LISTA (J))
   18 CONTINUE
          GO TO 1060
 1050 CONTINUE
          ALAMDA = 10. * ALAMDA
          CHISQ = OCHISQ
 1060 CONTINUE
C        PRINT,'*** A ',A(1),A(2),A(3),A(4)
        RETURN
        END
C
```

```
      SUBROUTINE FUNCS (X,A,Y,DYDA,NA,N,DELTAT)
            DIMENSION A(NA),DYDA(NA)
            M = N/2
            T = DELTAT*X
            IF(X.GE.M)T=DELTAT*(X-M)
            IF (X.GE.M) GOTO 9000
            Y = A(1) * EXP(A(2)*T) + A(3) * EXP (A(4)*T)
            DYDA(1) = EXP (A(2)*T)
            DYDA(2) = A(1)*T*EXP(A(2)*T)
            DYDA(3) = EXP (A(4)*T)
            DYDA(4) = A(3)*T*EXP(A(4)*T)
            DYDA(5) = 0
            DYDA(6) = 0
       9000 CONTINUE
            IF (X.LT.M) GO TO 9010
            Y = A(5) * EXP(A(2)*T) + A(6) * EXP (A(4)*T)
            DYDA(1) = 0
            DYDA(2) = A(5)*T*EXP(A(2)*T)
            DYDA(3) = 0
            DYDA(4) = A(6)*T*EXP(A(4)*T)
            DYDA(5) = EXP (A(2)*T)
      *     DYDA(6) = EXP (A(4)*T)
       9010 CONTINUE
            RETURN
            END
```

```
      SUBROUTINE MRQCOF(X,Y,NDATA,A,MA,LISTA,MFIT,
     *    ALPHA,BETA,NALP,CHISQ,DELTAT)
        DIMENSION X(NDATA),Y(NDATA),ALPHA(NALP,NALP),
     *  BETA(MA),DYDA(6),LISTA(MFIT),A(MA)
          COMMON SIG(1200)
        DO 12 J=1,MFIT
          DO 11 K=1,J
            ALPHA (J,K)=0.
   11   CONTINUE
          BETA (J) = 0.
   12   CONTINUE
        CHISQ = 0.
        DO 15 I=1,NDATA
C     WRITE(6,20)I
 20     FORMAT(' *** I= ',I5)
          CALL FUNCS (X(I),A,YMOD,DYDA,MA,NDATA,DELTAT)
          SIG2I=1./(SIG(I)*SIG(I))
          DY =Y(I)-YMOD
          DO 14 J=1,MFIT
            WT=DYDA(LISTA(J))*SIG2I
            DO 13 K=1,J
              ALPHA(J,K)=ALPHA(J,K)+WT*DYDA(LISTA(K))
   13     CONTINUE
            BETA(J) = BETA(J)+DY*WT
   14   CONTINUE
          CHISQ = CHISQ+DY*DY*SIG2I
   15   CONTINUE
         DO 17 J=2,MFIT
           JJ=J-1
           DO 16 K=1,JJ
             ALPHA (K,J) = ALPHA (J,K)
   16   CONTINUE
   17     CONTINUE
          RETURN
          END

       SUBROUTINE GAUSSJ (A,N,NP,B,M,MP)
       DIMENSION
A(NP,NP),B(NP,MP),IPIV(50),INDXR(50),INDXC(50)
C* TEST
C      WRITE(6,9000)
C9000 FORMAT(' ENTERED GAUSS')
C* END TEST
       DO 11 J=1,N
         IPIV(J)=0
   11 CONTINUE
```

```
      DO 22 I=1,N
        BIG=0
        DO 13 J=1,N
          IF(IPIV(J).EQ.1) GO TO 1030
            DO 12 K=1,N
              IF(IPIV(K).NE.0) GO TO 1020
                IF(ABS(A(J,K)).LT.BIG) GO TO 1010
                  BIG = ABS(A(J,K))
                  IROW = J
                  ICOL = K
1010          CONTINUE
              IF (IPIV(K).GT.1)
     *          WRITE (6,100)
 100     FORMAT ( 'SINGULAR MATRIX')
1020       CONTINUE
  12        CONTINUE
1030   CONTINUE
  13    CONTINUE
        IPIV(ICOL) = IPIV(ICOL) + 1
        IF (IROW.EQ.ICOL) GO TO 1040
          DO 14 L=1,N
            DUM = A(IROW,L)
            A(IROW,L) = A(ICOL,L)
            A(ICOL,L) = DUM
  14      CONTINUE
          DO 15 L = 1,M
            DUM = B(IROW,L)
            B(IROW,L) = B(ICOL,L)
            B(ICOL,L) = DUM
  15      CONTINUE
1040  CONTINUE
        INDXR(I) = IROW
        INDXC(I) = ICOL
        IF (A(ICOL,ICOL).EQ.0)
     *  WRITE (6,200)
 200     FORMAT ( 'SINGULAR MATRIX')
        PIVINV = 1./A(ICOL,ICOL)
        A(ICOL,ICOL) = 1.
        DO 16 L=1,N
          A(ICOL,L) = A(ICOL,L)*PIVINV
  16    CONTINUE
        DO 17 L=1,M
          B(ICOL,L) = B(ICOL,L)*PIVINV
  17    CONTINUE
        DO 21 LL=1,N
          IF (LL.EQ.ICOL) GO TO 1050
```

```
            DUM = A(LL,ICOL)
            A(LL,ICOL) = 0.
            DO 18 L=1,N
              A(LL,L) = A(LL,L) - A(ICOL,L)*DUM
   18       CONTINUE
            DO 19 L=1,M
              B(LL,L) = B(LL,L) - B(ICOL,L)*DUM
   19       CONTINUE
 1050   CONTINUE
   21    CONTINUE
   22 CONTINUE
      DO 24 LL=1,N
        L = (N-L) + 1
        IF (INDXR(L).EQ.INDXC(L)) GO TO 1060
          DO 23 K=1,N
            DUM = A(K,INDXR(L))
            A(K,INDXR(L)) = A(K,INDXC(L))
            A(K,INDXC(L)) = DUM
   23     CONTINUE
 1060 CONTINUE
   24 CONTINUE
C* TEST
C       WRITE(6,9010)
9010   FORMAT(' EXIT GAUSS')
C* END TEST
      RETURN
      END
```

Bibliography

Aubert H., "Eine scheinbare bedeutende Drehung von Objekten bei Neigung des Kopfes nach rechts oder links," Virchows Arch., vol. 20, pp. 381-93, 1861. See Schöne, 1984.

Buizza A. and R. Schmid, "Visual-vestibular interaction in the control of eye movement: Mathematical modelling and computer simulation," Biological Cybernetics, vol. 43, pp. 209-223.

Cohen B., D. Helwig and T. Raphan, "Baclofen and velocity storage: a model of the effects of the drug on the vestibulo- ocular reflex in the rhesus monkey," Journal of Physiology, vol. 393, pp. 703-725, 1987.

Cohen B., Henn V., Raphan T., Dennett D., "Velocity storage, nystagmus, and visual-vestibular interactions in humans. Ann NY Acd Sci, vol. 374, pp. 421-433, 1981.

Cohen B., V. Matsuo, and T. Raphan, "Quantitative analysis of the velocity characteristics of optokinetic nystagmus and optokinetic after nystagmus," Journal of Physiology, vol 270, pp. 321-344, 1977.

Collewijn H., "An analog model of the rabbit's optokinetic system," Brain Research, vol. 36, pp 71-88, 1972. Director S. and R. Rohrer, Introduction to Systems Theory, New York: McGraw-Hill Book Company, 1972.

Dai, J. M., Raphan, T., Sturm, D., Cohen, B., " Characterizaton of the three dimensional structure of velocity storage in the monkey," Society for Neuroscience, 1990.

Dennis J. E., Schnabel, R.B., Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall, 1983.

Dennis, J. E., Gay, D. M., Welsch, R. E., "Algorithm 573 NL2SOL-An adaptive nonlinear least-squares algorithm [E4], TOMS, vol. 7, pp. 369-383, 1981.

Director S., Rohrer R., Introduction to System Theory. McGraw Hill, New York, 1972.

Fanelli R., Schnabolk C., Raphan T., "Neural network modelling of velocity generation during off-vertical axis rotation (OVAR), Society for Neuroscience, vol. 14, pp.173, 1988.

Fernandez C., Goldberg J.M., "Physiology of peripheral neurons innervating otolith organs of the squirrel monkey," Journal of Neurophysiology, vol. 39, pp. 970-984, 1976.

Goldberg J.M., Fernandez C., "Physiological mechanisms of the nystagmus produced by rotations about an earth-horizontal axis, Annals of the Newy York Academy of Science, vol. 374, pp. 40-43, 1981.

Goldberg J.M., Fernandez C., "Physiology of peripheral neurons innervating semicircular canals of the squirrel monkey, J. Neurophysiol, vol. 34, pp. 635-660, 1971.

Goldstein H., Classical Mechanics, Addison Wesley, Mass., 1965.

Hain T.C., "A model of the nystagmus induced by off vertical axis rotation," Biological Cybernetics, vol. 54, pp. 337-350, 1986.

Harris, L.R., "Vestibular and optokinetic eye movements evoked in the cat by rotation about a tilted axis," Experimental Brain

Research, vol. 66, pp. 522-532, 1987.

Henn V. and B. Cohen, "Visual-vestibular in motion perception and generation of nystagmus," Neurosciences Research Program Bulletin, Cambridge, Massachusetts: M.I.T Press, 1980

Howard I.P, Human Visual Orientation, John Wiley & Sons, Chichester, 1982.

Hudspeth A.J., Corey, D.P., Sensitivity, polarity and coduction change in the response of vertebrate hair cells to controlled mechanical stimuli. Annals of the National Academy of Science, vol. 74:6, pp. 2407-2411, 1977.

Kaplan W., Lewis, D., Calculus and Linear Algebra, Wiley & Sons, p.974, 1971.

Jell R.M., Sett S., Ireland D.J., "Human optokinetic afternystagmus, is the fast component of OKAN decay due to smooth pursuit?," Acta Otolaryngol., Stockh., vol. 104, pp. 298-306, 1987.

Kaufmann E.L., Reese E.P., Volkmann J., and Rogers S., Summary Report No. 5DC 131-1-5, Psychophysical Res. Unit, Mount Holyoke College, 1949.

Lang S., Linear Algebra, Reading, Massachusetts: Addison Wesley, 1966.

Lafortune, S. , Ireland, D.J., Jell, R.M. & Duval, L., "Human optokinetic afternystagmus. Charging characteristics and stimulus exposure time dependence in the two-component model," Acta Otolaryngol., vol 101, pp. 353-360, 1986.

Lowenstein,O. E., Sand, A., "The mechanism of the semi-circular

canals: A study of responses of single fiber preparations to angular accelerations and to rotation at constant speed", Proc. R. Soc. Lond. Ser. B, vol 129, pp. 256- 275, 1940.

Mann C.W., Berthelot-Berry N.H., Dauterive H.J., "The perception of the vertical: I. Visual and non-labyrinthine cues," Journal of Experimental Psychology, vol. 39, pp. 538-47, 1949.

Marquardt, D.W., "An algorithm for least-squares estimation of non-linear parameters," SIAM Journal of Applications of Mathematics, vol. 11, pp. 431-441, 1963.

Matsuo V. and B. Cohen, "Vertical optokinetic nystagmus and vestibular nystagmus in the monkey: Up-down asymmetry and effects of gravity," Experimental Brain Research, vol. 53, pp. 197-216, 1984.

Melvill-Jones G., Milsum J.H., "Neural response of the vestibular system to translational acceleration. In Supplement to conference on systems analysis approach to neurophysiological problems, Brainerd, Minnesota, pp. 8-20, 1969.

Mittelstaedt H., "The subjective vertical as a function of visual and extraretinal cues," Acta Psychologica, vol. 63, pp. 63-85, 1986.

More, J. J., "The Levenberg-Marquardt algorithm: implementation and theory," Numerical Analysis, G.A. Watson, ed., Lecture Notes in Mathematics, vol. 630, pp. 105-116, 1977.

Mowrer O.H., "The influence of vision during bodily rotation upon the duration of post-rotational vestibular nystagmus," Acta Otolarynology, vol. 25, pp. 351-364, 1937.

Meuller G.E., "Uber das Aubertsche Phanomenon," Z. Psychol.

Physiol. Sinnesorg., vol 49, pp. 109-246, 1916. See Schöne, 1984.

Oppenheim A. and R. Schafer, Digital signal processing, New Jersey: Prentice Hall, 1975.

Osborne, M.R., "Nonlinear least squares - the Levenberg algorithm revisited," Journal of Australian Mathematics Society, vol. 19, pp. 343-357, 1976.

Powell, M. J. D., "Convergence properties of a class of minimization algorithms," Nonlinear Programming, O. Mangasarian, R. Meyer, and S. Robinson, eds., Academic Press, New York, pp. 1-27, 1975.

Pellionisz A., "Tensorial aspects of the multidimensional approach to the vestibulo-oculomotor reflex and gaze," in Reviews of Oculomotor Research. I. Adaptive Mechanisms in Gaze Control, Amsterdam: Elsevier, pp. 281-296,1985.

Press W., Flannery B., Teukolsky S., Vettering W., Numerical Recipes The Art of Scientific Computing, Cambridge University Press, Cambridge, 1987.

Rapan T. and B. Cohen, "Integration ond its relation to ocular compensatory movements," The Mount Sinai Journal of Medicine, vol. 47, No. 4, July-August, pp. 410-417, 1980.

Raphan T. and B. Cohen, "Gravitational effects on visual- vestibular-oculomotor coordinate transformations generating compensatory eye movements," Proceedings of the Society for Neuroscience, vol. 9, pp. 316, 1983.

Raphan T. and B. Cohen, "Multidimensional organization of the ves-

tibulo-ocular reflex (VOR)," in Adaptive Processes in Visual and Oculomotor Systems, Pergamon Press, pp. 285-292, 1985.

Raphan T. and B. Cohen, "Effects of gravity on the principal axes of velocity storage in three dimensions,Society for Neuroscience, Abstract, vol. 13, pp. 1225, 1987.

Raphan T. and B. Cohen, "Organizational principles of velocity storage in three dimensions; the effect of gravity on cross- coupling of optokinetic nystagmus (OKAN)," Annals of the New York Academy of Sciences, 1988.

Raphan T., V. Matsuo, and B. Cohen, "A velocity storage mechanism responsible for optokinetic nystagmus (OKN), optokinetic after-nystagmus (OKAN), and vestibular nystagmus," in Control of gaze by brain stem neurons, Amsterdam: Elsevier/North Holland Biomedical Press, pp. 37-47, 1977.

Raphan T., V. Matsuo, and B. Cohen, "Velocity Storage in the vestibulo-ocular reflex arc," Experimental Brain Research, vol. 35, pp. 229-248, 1979.

Raphan T. and C. Schnabolk, "Modelling slow phase velocity generation during off-vertical axis rotation (OVAR)," Annals of the New York Academy of Sciences,1988.

Resine H., Raphan T., Cohen B., Katz E., "Signal processing in the vestibular nuclei during off-vertical rotation (OVAR), Society for Neuroscience, vol. 14, pp. 172, 1988.

Robinson D.A., "Vestibular and optokinetic symbiosis: an example of explaining by modeling," in Control of gaze by brain stem neurons, Amsterdam: Elsevier, pp. 49-58, 1977.

Robinson D.A., "The use of matrices in analyzing the three- dimensional behavior of the vestibulo-ocular reflex," Biological Cybernetics, vol. 46, pp. 53-66, 1982.

Schiff D., B. Cohen and T. Raphan, "Roll OKN and OKAN Effects of head position Society for Neuroscience, Abstract, vol. 12, pp. 774, 1986.

Schöne H., Spatial Orientation, Princeton University Press, Princeton, N.J.,1984.

Simpson J.I., W. Graf, "Eye muscle geometry and compensatory eye movements in lateral eyed and frontal eyed animals," in, Vestibular and Oculomotor physiology, International Meeting of the Barany Society, pp. 20-30, 1981.

Skavenski, A.A. and D.A. Robinson, "Role of abducens motoneurons in the vestibulo-ocular reflex," Journal of Neurophysiology, vol. 36, pp. 724-738, 1973.

Steinhausen W., "Uber die beobachtung der cupula in den bogengangesampullen des labyrinths des lebenden," Hects. Pflugers Arch., vol. 232, pp. 500-512, 1933.

Sturm D. and T. Raphan, "Modelling the three dimensional structure of velocity storage in the vestibulo-ocular reflex (VOR)," Proceedings of the Northeast Bioengineering IEEE,1988.

Ter Braak J.W., "Untersuchungen ueber optokinetischen Nystagmus," Arch. Neerl Physiology, vol. 21, pp. 309-376, 1936.

Waespe W., B. Cohen, T. Raphan, "Role of the flocculus in optokinetic nystagmus and visual-vestibular interaction: Effects of flocculectomy," Experimental Brain Research, vol. 50, pp. 9-

33, 1983.

Wilson V. and G. Melvill-Jones, Mammalian vestibular physiology, New York: Plenum Press, 1979.

Witkin H.A., Asch S.E., "Studies in space orientation: IV. Further experiments on perception of the upright with displaced visual fields," Journal of Experimental Psychology, vol38, pp. 762-782, 1948.

Zadeh L.A. and C. A. Desoer, Linear system theory: The state approach, New York: McGraw Hill, 1963.