

A SYSTEM FOR 3D TRACKING
OF LIMB MOVEMENT

by

REPHAEL KLEIN

A project
presented to the University of Manitoba
in partial fulfillment of the
requirement for the degree of
Master of Engineering
in
Electrical Engineering

Winnipeg, Manitoba

Dec, 1987

© Rephael Klein, 1987

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-48047-5

A SYSTEM FOR 3D TRACKING OF LIMB MOVEMENT

BY

REPHAEL KLEIN

An Engineering Report submitted to the Faculty of Graduate
Studies of the University of Manitoba in partial fulfillment
of the requirements of the degree of

MASTER OF ENGINEERING

(c) 1988

Permission has been granted to the LIBRARY OF THE
UNIVERSITY OF MANITOBA to lend or sell copies of this
Engineering Report, to the NATIONAL LIBRARY OF CANADA TO
microfilm this Report and to lend or sell copies of the film,
and UNIVERSITY MICRO-FILMS to publish an abstract of this
Report.

The author reserves other publication rights, and neither
the Report nor extensive extracts from it may be printed
or otherwise reproduced without the author's permission.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	v
ACKNOWLEDGEMENTS	vi
LIST OF ABBREVIATIONS AND SYMBOLS	vii
 1. INTRODUCTION	 1
1.1 Problem Definition	1
1.2 Review of Related Literature	2
 2. REVIEW OF TRACKING METHODS	 4
2.1 Active-Passive methods	4
2.2 Continuous-Pulsed Waves Tracking Methods	6
2.3 Selecting the Type of Radiation	7
2.3.1 Tracking with Light Beams	7
2.3.2 Tracking with Electromagnetic Waves	8
2.3.3 Tracking with Sound Waves	8
 3. TRACKING VOLUME	 12
3.1 Single Transducer Beam Pattern	12
3.1.1 Axial Response	13
3.1.2 3D Beam Field	16
3.1.3 Beam Pattern of a Transducer	20
3.1.4 Volume of a Beam	22

3.2	Volume of the Array	26
3.2.1	Common Volume of an Array	26
3.2.2	Selecting the Locations of the Transducers .	29
3.2.3	Beam Intersection Due to Three Transducers on on One Axis	30
3.2.4	Redundancy Effect	32
3.2.5	Effect of the Beam Pattern of the Transmitter	33
3.2.6	Tilting the Transmitter	34
4.	SYSTEM ARCHITECTURE AND HARDWARE DESCRIPTION	39
4.1	Selection of Main Computer	41
4.1.1	Graphics Capability	41
4.1.2	Communication Rate	41
4.1.3	Selection of Microprocessor	43
4.2	Control Processor	45
4.2.1	Microprocessor Circuits Description	45
4.2.2	ACIA Circuits Description	48
4.2.3	PTMs Circuit Description	49
4.3	Signal Processors	54
4.3.1	Circuit Description	54
4.3.2	Modification of the SRM	56
4.4	Ultrasound Transducers	57
4.5	Reference Frame	58

5.	SOFTWARE DESCRIPTION	62
5.1	Control Processor Software	62
5.1.1	General Description	63
5.1.2	Structure of the Software	67
5.1.3	Communication Protocols	70
5.2	Main Computer Software	72
5.2.1	Main Menu Description	72
5.2.2	Receivers Set Up Module	74
5.2.3	Distance Test Module	75
5.2.4	Stability Test Module	78
5.2.5	Locating Modules	79
5.3	Locating Algorithm	81
5.3.1	Removing the Square Values	81
5.3.2	Implementation of the Coordinates Transformation	85
5.3.3	Software Implementation	88
6.	TESTING	92
6.1	Testing Hardware Subsystems	92
6.1.1	Frequency of the Clock	92
6.1.2	Measurement of the Timers	93

6.2	System Calibration	95
6.2.1	Distance Measurement	95
6.2.2	Jitter Measurement	98
6.2.3	Tracking Volume Measurement	103
6.2.4	Verification of the Location of a Target .	107
7.	CONCLUSIONS AND RECOMMENDATIONS	111
	REFERENCES	113
	Appendix A: Control Processor Diagrams	116
	Appendix B: Control Processor Program Listings	123
	Appendix C: Main Computer Program Listings	146

ABSTRACT

This work is motivated by the need for a three dimensional (3D) tracking system which could be utilized in the area of behavior modification to rehabilitate people whose limb movement is partially impeded, as caused by stroke or other illnesses. This rehabilitation can be done by the principle of shaping through reinforcement, which is based on the data collected from the tracking system.

The 3D tracking system developed in this project utilizes ultrasound waves. Since ultrasound is often used for underwater tracking, our air ultrasound system has to deal with problems that are unique to ultrasound waves in air which are caused by the relatively high acoustic impedance of the air. The 3D system consists of an array of ultrasound transducers, amplifiers, and a control processor which acts as an intermediate controller between the transducers and a graphics computer. Data representing the distance between a transmitter and the receiving array are acquired simultaneously and transmitted to the graphics computer from the controller. The data are then processed and used by a locating algorithm to compute a trajectory representing the limb movement, with ± 0.5 cm accuracy. The system has been designed constructed and tested.

ACKNOWLEDGEMENTS

I would like to thank Dr. Witold Kinsner, my advisor, for his dedicated and endless help and guidance throughout the research work, as well as for providing the concept and key solutions used in this work.

I would also like to thank to Dr. J. J. Pear for his advice. The tracking system is intended to be used in that research.

I would like to thank to the University of Manitoba, and the Microelectronics Centre, for providing the facilities which were needed to implement this apparatus. In addition, I would like to thank to the Natural Sciences and Engineering Council of Canada and the Manitoba Health Research Council for their partial financial support of this work.

I would like to thank to the examinining commitee: Prof. A. Jacobschuck, Prof. M. Yunik, and Dr. A. DeGroot for their helpful remarks and corrections.

I although express my deep gratitude for my wife Etty, and my family, for their understanding, support, and patience during the research.

LIST OF ABBREVIATIONS AND SYMBOLS

a	Radius of the piston.
ACIA	Asynchronous Communications Interface Adapter.
a_r	Absorption factor of the room walls.
BASIC	Beginners All purpose Symbolic Instruction Code.
b_i	The delay of the transducer.
BLNK	A pulse which is produced by the control processor to inhibit the receiver circuit in the SRM.
c	Speed of the sound waves.
CPU	Central Processing Unit.
d_i	The distance of the target from transducer i.
D_{min}	The minimal distance which is allowed between any receiver transducer of the array, and the central transducer.
dS	Element of area.
dX	The maximum deviation distance, which is allowed for a tilted transmitter for a certain distance Z.
d_1, d_2, d_3	The measured distances by a metric tape.

$d1', d2', d3'$	The calculated distances.
$d3''$	The calculated value of the distance with the measured coefficients.
ECHO	A pulse which denotes the duration of propagation of the ultrasound pulse in the air, and is proportional to the distance of the receiver from the transmitter.
F_{dr}	Distance measurement frequency.
G	The combined attenuation of the acoustic intensity.
G_{θ}	Angular attenuation of the acoustic intensity.
I	Acoustic intensity.
$I(r, \theta)$	Acoustic intensity for distance r , and angle θ .
I_0	Acoustic intensity at $t=0$, and along the acoustic axis.
INIT	A timer which is activated at the beginning of the search or tracking cycle.
IRQ	A interrupt request pulse which is applied to the control processor.
J_1	First order Bessel function.
k	Wave number ($k=\omega/c$).

LE	The linearity error.
LOAD	A timer which is activated for 2 msec, at the end of INIT timer in multi track cycle.
MODE	Denotes a state in the control processor software.
n	The number of the receivers in the array.
p	Momentary pressure.
$p(r,0,t)$	Momentary pressure along the acoustic axis.
$p(r,\theta,t)$	Momentary pressure pattern.
$P(r,0)$	Absolute value of the sound pressure along the acoustic axis.
$P(r,\theta)$	Absolute acoustic pressure.
$P_{ax}(r)$	Far-field sound axial pressure.
PTM	Programmable Timer Module.
Pw	The width of the pulse, which is measured by the Universal Timer in Sec. 6.1.2.
Q	Source strength.
r	Distance between the centre of the source and the point of interest.

r' Distance of the point from a piston element.

Ra_{min} The minimal communication rate of the system.

S Area.

SRM Sonar Ranging Module.

t Time.

T Time constant.

t_{ECHO} The time which is measured by the ECHO timer.

t_i The propagation time which is measured by the ECHO timer.

t_{INIT} The value of the INIT pulse.

U A notation for IC.

v Velocity.

v_o Peak velocity amplitude of the piston.

V Volume of a beam or array of beams.

Val The value of the data for a transducer which is the 16 bit value of two ASCII characters.

Val1 The value of the data for a first character in the pair, and is the 8 bit value of ASCII character.

Val2 The value of the data for a second character in the pair, and is the 8 bit value of ASCII character.

x₂ The distance between two transducers, which is used to calculate x_s (the x coordinate of the target).

x_i,y_i,z_i The coordinate values of transducer i (one of the three which are used for locating).

x_s,y_s,z_s The coordinate values of the target location.

x',y',z' The new values of x,y,z, after the first step of transformation in the location algorithm.

x'',y'',z'' The new values of x,y,z, after the second step of transformation in the location algorithm.

x''',y''',z''' The new values of x,y,z, after the third step of transformation in the location algorithm.

x''',y''',z''' The new values of x,y,z, after the fourth step of transformation in the location algorithm.

Z The trajectory of the location of the target on the axis of the limb movement.

Z_{min} The minimal distance which is allowed between target and central transducer, along the axis of the limb movement.

θ	Angle between r , and the acoustic axis of the transducer.
$\theta, \theta_1, \theta_2$	The transformation angles for each step, in the locating algorithm.
θ_d	The tilt angle of the acoustic axis of a transducer receiver, from the axis of the limb movement.
θ_{d1}	The tilt angle of the acoustic axis of a transducer transmitter, from the axis of the limb movement.
θ_{max}	The tilt angle, where the system stops to measure the distances.
θ_w	The solid angle which denotes the width the beam of a transducer.
ρ_0	Volume density of the air.
σ	Radius of a ring of the piston.
ω	Angular frequency.

CHAPTER I

INTRODUCTION

1.1 Problem Definition

The system which is introduced in my project has been motivated by the possibility to rehabilitate disabled people by altering limb movements, such as stroke victims, brain injured, and other type of handicapped persons, whose coordinated movements are damaged. The rehabilitation method is based on the behavior modification principle of shaping [PeLe87], first by reinforcing very slight limb movements, and gradually increasing the size of the limb movement. The reinforcement is based on the review of the limb movement, which is done by a tracking system. It is, therefore, important to establish a 3D tracking system which locates the target, and draws the 3D trajectory of the limb movement. The analysis of the trajectory is used to stimulate the proper reinforcement for the limb movement.

An experiment in this area has been performed in the past [Pear85]. In his experiment, Dr. Pear used pigeons and tracked their key pecking habits, which were reinforced by a limited time access to food hopper. The tracking system was based on two TV cameras which were connected to a data acquisition system. In order to detect the head movements of the pigeons, he used white birds and painted their head with black shoe polish. Although, this method proved that it is possible to shape the behavior of pigeons, it is complicated method, needs expensive hardware like TV cameras, and complex image processing methods, therefore, a simpler and more accurate system is needed. The difference between the tracking methods, and the reasons for selecting the ultrasound

tracking system are discussed in the following sections.

The raising of a human hand is selected as the first limb movement which is tracked by the system. Since the system is designed for disabled people, the distances over which they move their hand are relatively small, and limb movement is slow. The assumption is that the maximum amplitude of the limb movement is about 85 cm, and most of the time it will be less than 65 cm. The distance measurement rate is 80 Hz, which is higher than this which is needed by the psychologist (30 Hz).

1.2 Review of Related Literature

There are mainly two limb-tracking methods given in the technical literature reports. The first method uses optical means [AmLA84]. In this method several LEDs are attached to an arm of a patient, and the emitted light of the diodes is recorded with cameras on a film or magnetic tape. The recordings are analyzed later. The main problem with this technique is that the 3D tracking requires at least three cameras and complicated methods of image processing, which are difficult to achieve in real time systems. In addition, in order to get effective recording, the LEDs have to be on a dark background, and this restriction complicates the use of the system.

The other method for limb tracking requires that the tested limb be attached to a mechanical turning table [AlIn86]. This table contains goniometers which transmit information about the movement of the hand. The collected data from this system is used for functional neuromuscular stimulation of the upper limb of people whose nerve system is damaged. Although the system is operating in real time, the main disadvantage is that the person

cannot move the limb freely. Consequently, a simple tracking system which does not impede limb movement and does not need a complicated image processing algorithm is selected.

CHAPTER II

REVIEW OF TRACKING METHODS

The tracking method used in this study is based on measuring the distances between the target to an array of receiving transducers. The distance measurement is based on knowing the speed of the waves and the time in which the waves propagate from the transmitter to the receiver. The tracking methods can be distinguished on the basis of: (i) where the source of waves is located, (ii) the type of waves which are used for tracking, and (iii) the type of radiation and medium in which the waves are propagate.

2.1 Active-Passive methods

One of the ways used to categorize tracking methods is by defining whether the target or the observer are the source of the waves [KlKP87]. In the active tracking method the target is radiating waves toward the observer, which receives the waves, and the target is located by a triangulation algorithm. On the other hand, in the passive tracking method the observer is radiating the waves which are reflected by the target to the observer, which utilizes them for further analysis.

When the system is using the active method the signals are usually stronger, and in addition, there is separation between the receivers and the transmitter functions. Therefore, the receiver circuits are simpler than those used in the passive method.

In the active tracking method there is a danger that the system identifies and locates multiple targets which transmit signals with similar characteristics. Thus, the characteristics of

the transmitted signals have to be unique, stable, and well known to the observer. For this reason, this method is not applicable for tracking systems where it is not possible to force the target to transmit the unique signal which is needed to prevent multiple identifications.

Another problem that occurs when the active method is used is that if the observer does not know when the target starts to transmit, there is a need for an additional receiver in the array in order to locate the target [ScRo87]. This constraint increases the number of receivers which are needed to locate a target from three to four.

Since in the passive method the observer transmits the signals, the tracking is not dependent on the signals which are transmitted by the target. On the other hand, in addition to the problem with the complex receiver circuits which is pointed out previously, there are additional problems with the passive method. The location of the observer appears as well, and this is a disadvantage for military systems where the observer does not want to be discovered. In addition, since the passive method is based on reflected signals, it is more effective to echo signals which are reflected by other objects and to the reverberant signals.

The active method is used by us because the active tracking method is simpler and we can control the transmitted signals by the attachment of a remote controlled transmitter to the hand of the patient (which is the target). In addition, since the transmitter is controlled by the system, the observer knows when the transmitter starts to transmit, and there is no need for a supplementary receiver in the array.

2.2 Continuous-Pulsed Waves Tracking Methods

An other way to classify tracking methods is by the type of the signal which is used for tracking. There are two types of signals which are used for measuring the distance: the first one is the continuous wave method, and the second one is the pulsed wave method.

In the continuous wave method, the signal which is transmitted by the wave source is a continuous signal, and the distance is proportional to difference between the phase of the signal when it arrives at the receiver and the phase when it leaves the source. This method is more immune to noise than the pulsed wave method, because the signals are transmitted continuously, and the phase-locked loops which are used to measure variations in the phase can integrate the signals and filter out high frequency noise. On the other hand, the receiver circuits are more complex than those which are used in the pulsed wave method. In addition, for targets which are located further from the array than the wave length of the signal this method requires an a-priori knowledge of the distance.

In the pulsed wave method the distances are determined by measuring the time in which the transmitted pulse propagates from the transmitter to the receiver. This method is less immune to noise, but the circuits are simpler, and the method is not dependent on the wave length of the signals.

The pulsed wave method was selected because: (i) our system utilizes the active tracking method where the signals are strong, and (ii) the wave length of the tracking signal which is imposed by the manufacturer of the transducer is 6.82 cm, which is smaller

than the limb movement that is defined in Sec. 1.1. In addition, in order to improve the immunity to noise and prevent multiple identifications the system transmits a burst of 16 pulses with the PRF of 49.4 kHz [Texa84].

2.3 Selecting the Type of Radiation

Although, most of the tracking methods are based on the measurement of the duration in which the signal propagates from the transmitter to the receiver, there are big differences in the distances, frequencies, and medium in which these methods can be implemented. In the technical literature we encounter mainly three kinds of radiation which are used for tracking: light waves, electromagnetic waves, and acoustics waves. In this section the differences between these three methods and the reason why the acoustic method was selected for our system are discussed.

2.3.1 Tracking with Light Beams

In spite of major developments in opto-electronics during the recent years, this technique is still costly, and requires complex circuits and algorithms to measure and analyze the related data. The light waves propagate relatively well in air, and are utilized for tracking in 2D [AmLA84]. The problem with this method is, that in order to track in 3D there is a need for at least three cameras, involving complex algorithms to analyze and process the image. For these reasons, this method was not selected.

2.3.2 Tracking with Electromagnetic Waves

Electromagnetic waves propagate well over long distances, and for this reason they are heavily utilized in air and sea navigation, where distances are relatively large. Since the speed of the electromagnetic waves is extremely high, long distances can be measured with a high repetition rate. On the other hand, as described in Sec. 3.2.2 the maximum distance of the target from the array is 1.65 m, and therefore, the propagation time of the EM signal is 0.55 nsec. Since the accuracy needed for distance measurement is 0.1% at least, this accuracy is hard to achieve with regular electronic counters. Therefore, the electromagnetic tracking method is not implemented in our system.

2.3.3 Tracking with Sound Waves

Because of the better impedance matching of the sound waves with water, ultrasound waves are often utilized for tracking objects that are located under water. In air they are used for auto focusing of cameras [Ciar80], and [Ciar84]. As mentioned earlier, the sound waves propagate slower in air than electromagnetic waves, and therefore, for short distances, it is easier to measure the duration of the propagation of sound waves. Since our tracking system is designed for short distances, therefore, acoustic radiation is selected for tracking, despite the problems which are discussed in the current section.

The acoustic impedance of air, is higher than that of other propagation media, like water or solids [ShHs79]. Since the tracking system has to operate in air, it needs relatively large energy in order to transmit sound waves. On the other hand, because the transmitter is attached to the human limb and we do

not want to harm the patient, we are interested in reducing the energy supplied to the transducer, to the minimal level required by the receiver circuits of the array. In order to cope with this problem, a directional transducer was chosen.

The directionality of a transducer, as is described in Sec. 3.1.2, increases according to its size, and the frequency of the sound waves. In order to prevent limitations on the limb movement, the aim is to reduce the size of the transducer, which increases the frequency of the sound waves used. On the other hand, as shown by Fig. 2.1 the absorption of sound energy in air is a proportional to frequency, and for those reasons, the frequency of about 50 kHz was selected by the manufacturer of the transducer [Pola85].

Finally, the system has to cope with reverberation of sound waves. As shown in Fig. 2.2, sound waves are reflected by the walls of a room, and if the reflected signals are strong there is a danger that the system may track the reverberation signal, and not the target. At the instant t , the decrease in the acoustic intensity I , caused by reverberation is calculated from [DoFf83]

$$I(t) = I_0 \exp\left(\frac{-a_r c t}{4V}\right) \quad (2.1)$$

where I_0 is the acoustic intensity at $t=0$, a_r is the absorption coefficient of the room walls, c is the speed of sound, and V is the volume of the room. The crucial time, T , in which the reverberant signal can affect the system, can be evaluated by Eq. (2.1) in the following way: If one assumes that when I is 35 dB less than I_0 , the receiver cannot to process the reverberant signal. In addition by substituting in Eq. (2.1) the values of c with 341 m/sec, V with a typical room volume of 22.5 m³, and a_r

Leaf inserted to correct page numbering

with 10 m^2 [DoFf83], the value of $T \approx 0.1 \text{ sec}$ is evaluated (which is 12.5 times higher than the repetition rate of distance measurement 80 Hz).

Reverberation can result in a multi-target identification problem. This problem can be solved by post processing, and cross correlation of the estimated arrival time of the signals [NgBa86], and [KnCa76]. Our system deals with this problem by two methods: as described in Sec 5.2.3, the first method utilizes the previous measurement of the arrival time of the signals for opening the gate of the receiver circuits only a short time before the ultrasound signal is expected, and the second method uses the fact that the transducers are directional, and the reverberation signals which are arriving usually from the sides are attenuated and below the threshold level of the receiver.

CHAPTER III

TRACKING VOLUME

The tracking volume is the volume in which the ultrasonic system can track the movement of a limb. The limb can be located within this tracking volume by measuring the distances between the limb and at least three receiving transducers that are not located on the one axis. This chapter describes how to calculate the volume of a beam for a single transducer, and the volume at the cross sections of three beams of the receiving transducers which are used for location.

3.1 Single Transducer Beam Pattern

The shape of an ultrasound beam radiated by an ultrasound transducer as used in this tracking system resembles the radiation pattern of a plane circular piston. The shape of such a beam has been calculated by Kinsler et al [KFCS82]. By using these equations and the pertinent data such as radiation pattern and size provided by the manufacturer of the transducer, it is possible to define the effective volume in which the limb can be tracked. Let the piston move uniformly with a sinusoidal velocity of [KFCS82]

$$v = v_0 \exp(j\omega t) \quad (3.1)$$

where v_0 is the velocity amplitude of the piston, and ω is the angular frequency $\omega=2\pi f$, then the source strength Q is a function of the piston velocity, and the area S of the transducer given by

$$dQ = v_o dS \quad (3.2)$$

The momentary pressure p at a point of interest generated by the source can be calculated using the geometry shown in Fig. 3.1. and the following expression [KFCS82]

$$p(r, \theta, t) = j \frac{\rho_o c v_o k}{2\pi} \int_S \frac{\exp(j\omega t - kr')}{r'} dS \quad (3.3)$$

where r is the distance between the centre of the source and the point of interest, θ is the angle between r and the acoustic axis of the transducer, ρ_o is the volume density of the air, c is the thermodynamic speed of sound in the air, $k = \omega/c$ is the wave number, and r' is the distance of the point from a piston element that its area is dS .

3.1.1 Axial Response

In order to derive Eq. (3.3) the pressure along the acoustic axis ($\theta=0$) is derived first. As shown in Fig 3.1, the axial response is calculated by using rings with a radius of σ for the region of $\sigma \leq a$, where a is the radius of the piston. The pressure due to these rings is summed up and produces the sound pressure $p(r, 0, t)$ along the acoustic axis a given by [KFCS82]

$$p(r, 0, t) = j \frac{\rho_o c v_o k}{2\pi} e^{j\omega t} \int_0^a \frac{\exp(-jk \sqrt{r^2 + \sigma^2})}{\sqrt{r^2 + \sigma^2}} 2\pi \sigma d\sigma \quad (3.4)$$

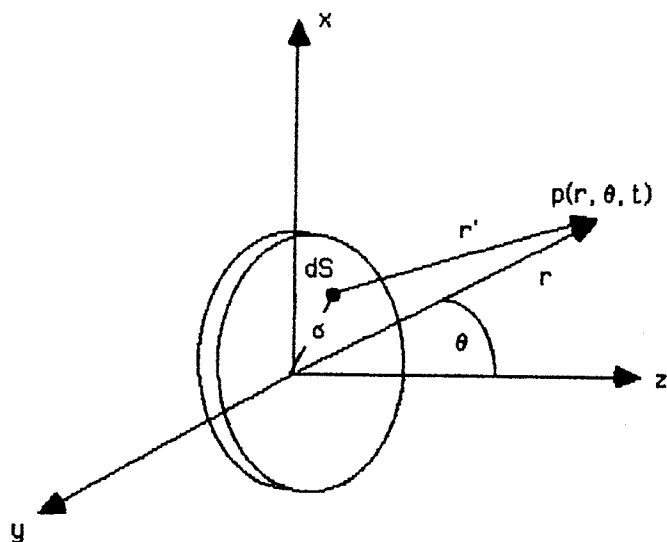


Fig. 3.1. Geometry which is used to derive the radiation of a piston [from KFC82].

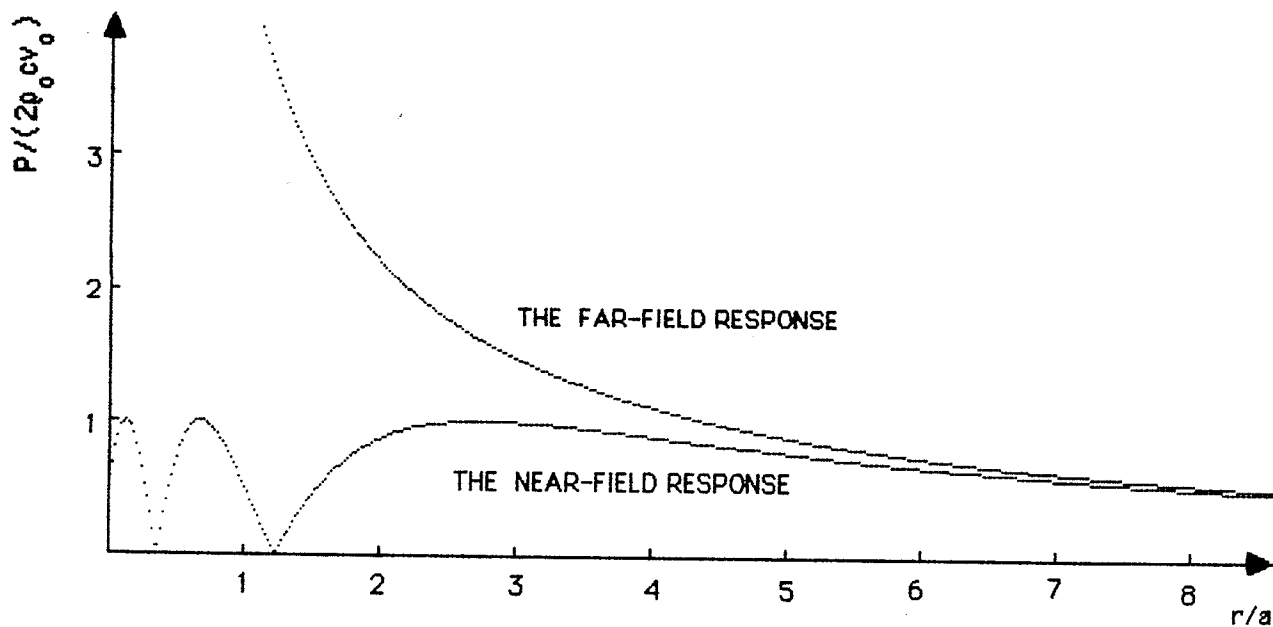


Fig. 3.2. On axis response of a piston with $a/\lambda = 2.815$.

The absolute value of the sound pressure $P(r,0)$ along the acoustic axis is

$$P(r,0) = 2\rho_0 c v_0 \left| \sin \left\{ \frac{1}{2} k r \left[\sqrt{1 + \left(\frac{a}{r} \right)^2} - 1 \right] \right\} \right| \quad (3.5)$$

For $r \gg a$ the square root in Eq. (3.5) can be simplified to

$$\sqrt{1 + \left(\frac{a}{r} \right)^2} \cong 1 + \frac{1}{2} \left(\frac{a}{r} \right)^2 \quad (3.5a)$$

If also $r \gg ka^2$, then Eq. (3.5) can be simplified and the far-field sound axial pressure $P_{ax}(r)$ is

$$P_{ax}(r) = \frac{1}{2} \rho_0 c v_0 \frac{a}{r} k a \quad (3.6)$$

Notice that, for a given transducer size and ultrasound frequency P is inversely proportional to r . The other parameters ρ_0, c, k may be affected by the humidity and temperature of the air. Equations (3.5) and (3.6) are plotted in Fig. 3.2. It can be noticed from Eq. (3.5) that the pressure near the transducer does not increase monotonically but decreases and increases due to interference.

Since the Polaroid transducer is used at a frequency of $f=50$ kHz, and the speed of the sound is about 341 m/s, the wave number is $k=921.2$. Furthermore, since the diameter $2a$ of the transducer is 40 mm then $a/\lambda=2.815$. In Fig. 3.2, the far and near fields are displayed for the Polaroid transducer. At the distance of $7a$ the two equations yield almost the same value. Because the transducers are located further than 1 m from the target the relation r/a is about 50. For this reason the near-field effects can be neglected.

Finally, the acoustic intensity I is calculated according to

$$I = \frac{P^2}{\rho_0 c} \quad (3.7)$$

where P is the sound pressure given by Eqs. (3.5), (3.6), or (3.10a), and is inversely proportional to r^2 . The significance of Eq. (3.7) is that by knowing the relation between the acoustic intensity and r and θ , it is possible to calculate, as described in the following sections, the volume of the beam, and by intersecting the beams to calculate the tracking volume.

3.1.2 3D Beam Field

The far field can be calculated by summing up the fields which are generated by each strip of the piston with the width of dx which is shown in Fig. 3.3. The source strength of each strip can be obtained by substituting the elementary area dS in Eq. (3.2) with $2a \sin \phi \, dx$ (which is the area of the strip)

$$dQ = 2v_0 a \sin \phi \, dx \quad (3.8)$$

where, as shown in Fig. 3.3, ϕ is the angle between the x axis and the radius which intersects with the edge of the strip. By using the equations for a continuous line source, the pressure contribution of each strip is

$$dp = j\rho_0 c \frac{v_0}{\pi r'} k a \sin \phi e^{j(\omega t - kr')} \, dx \quad (3.9)$$

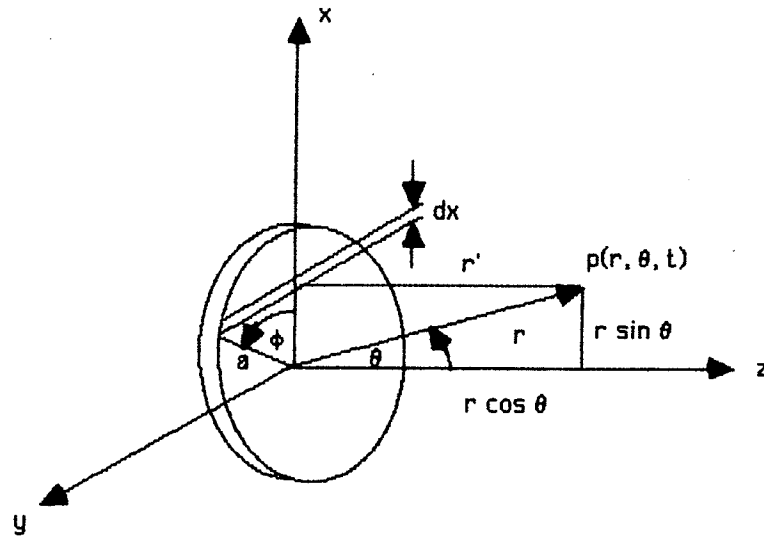


Fig. 3.3. Geometry which is used to derive the angular radiation of a piston from [KFC82].

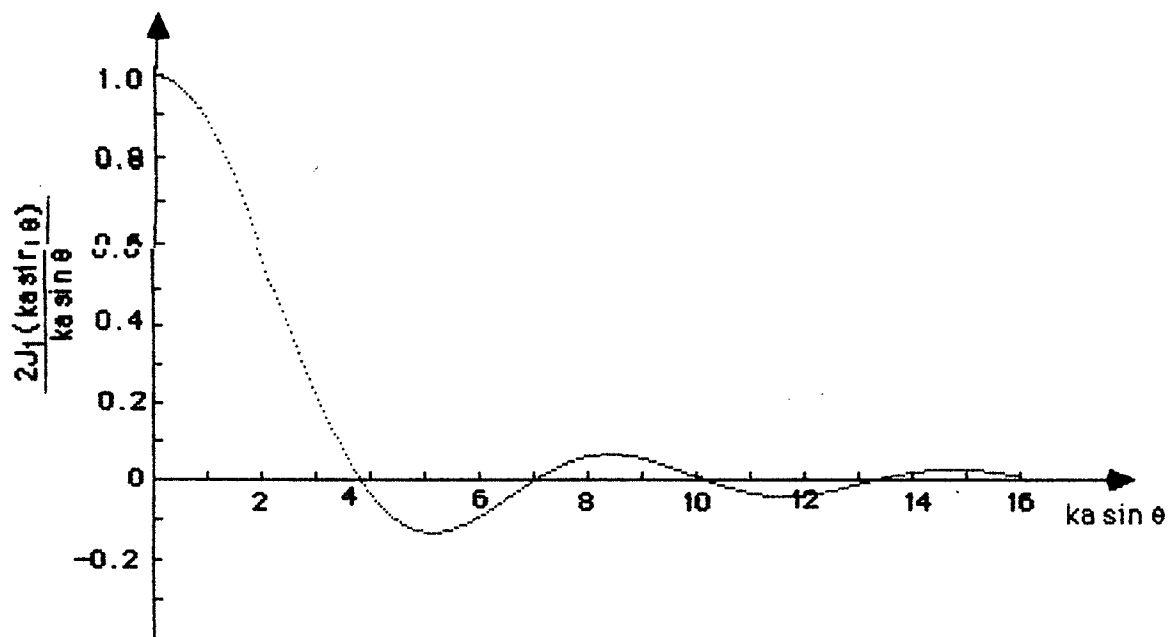


Fig. 3.4. The behavior of the Bessel function $2J_1(ka \sin \theta) / ka \sin \theta$.

As pointed out earlier, we are concerned with the far-field only. For convenience, Cartesian coordinates are translated into polar coordinates. The momentary pressure pattern, $p(r, \theta, t)$, can then be derived in the following [KFCS82]

$$p(r, \theta, t) = j \frac{\rho_0 c}{2} v_0 \frac{a}{r} k a \exp(j(\omega t - kr)) \left[\frac{2J_1(k a \sin \theta)}{k a \sin \theta} \right] \quad (3.10)$$

where J_1 is the first order Bessel function. Equation (3.10) is based on two parts: the first one is similar to Eq (3.6) which is the pressure along the acoustic axis, while the second part in the brackets is dependent on θ , and varies according to the Bessel function. The functional behavior of $2J_1(k a \sin \theta)/k a \sin \theta$ is shown in Fig. 3.4, and is plotted by using a table for Bessel functions which is calculated by Bayer [Beye78]. This Table is used later on to calculate the beam pattern from Eq. (3.10a).

Along the acoustic axis the absolute value of the sound pressure which is calculated by Eq. (3.6), or from Eq. (3.10) has to have the same values. Therefore, by combining Eqs. (3.6), (3.10), the absolute acoustic pressure $P(r, \theta)$ becomes

$$P(r, \theta) = P_{ax}(r) \left[\frac{2J_1(k a \sin \theta)}{k a \sin \theta} \right] \quad (3.10a)$$

If we take $ka=17.7$, then the beam pattern has the shape which is shown in Fig. 3.5a. We can see that the beam of a transducer is more directional when the size of the transducer and the frequency increase. The width of the main lobe is 25° . This is consistent with the beam shown in Fig. 3.5b. However, since this beam pattern was calculated for $ka=10$ the width of the beam was 44.5° , as opposed to our 25° for a more directional beam ($ka=17.7$).

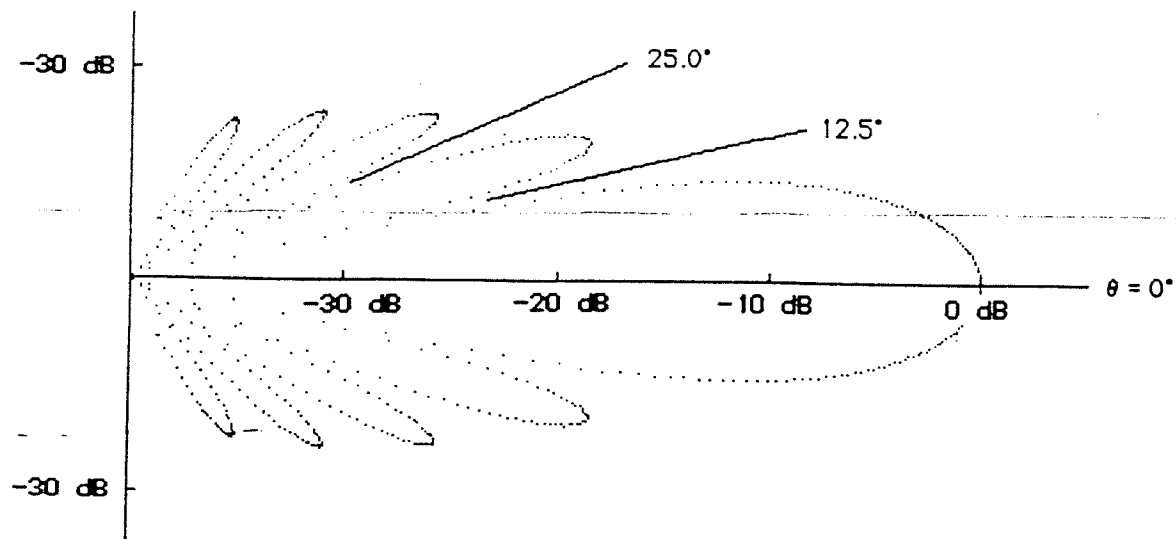


Fig. 3.5a. Beam pattern of a circular plane piston with $ka = 17.70$.

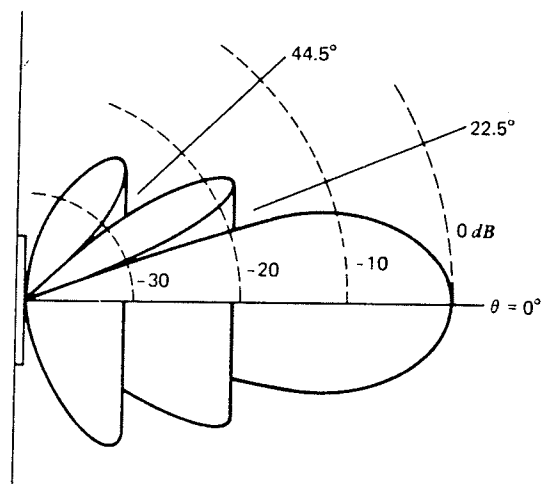


Fig. 3.5b. Beam pattern for a circular plane piston with $ka=10$ [from KFC582].

The measured beam pattern which is given by Polaroid [Pola85] has a shape similar to the calculated one and is shown in Fig. 3.6. The width of the main lobe is bigger and is about 30° . The other difference is that the gain difference between the main lobe and the first side lobe is 15 dB instead of the calculated value of 17 dB.

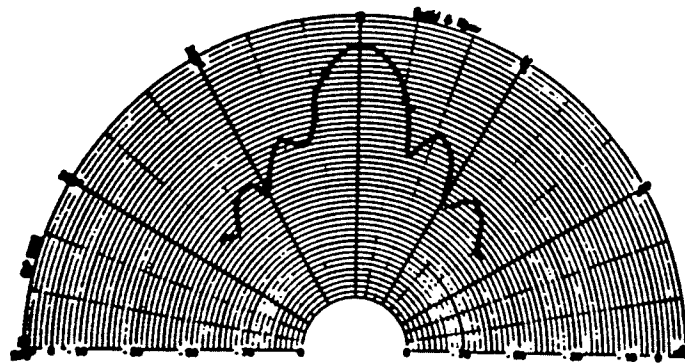
3.1.3 Beam Pattern of a Transducer

The volume of the beam of a transducer can be calculated from its beam pattern. In order to calculate the volume of the beam we have to scale the calculated beam pattern of the transducer to the measured one. This is done by substituting in Eq. (3.10) the calculated value of $ka=17.7$ with the value of $ka=14.8$, which is calculated as follows. As shown Fig. 3.4, the first time that the value of the Bessel function $2J_1(ka \sin \theta)/ka \sin \theta=0$ is when

$$ka \sin \theta = 3.827 \quad (3.11)$$

From Fig. 3.6 we can measure the width of the main lobe, which is 30° , therefore, $\theta=30^\circ/2=15^\circ$. The new value for ka can be obtained by substituting the value of $\sin \theta$ in Eq. (3.11) with the value of $\sin 15^\circ$.

Since the volume of the beam is determined by the intensity of the ultrasound waves, Eqs. (3.7), (3.10a) are used to calculate the volume. This is done in two steps, the first one in this section calculates the beam pattern when the distance r is constant and the angle θ varies, the second step which utilizes the beam pattern for volume calculation is described in the next



Typical Beam Pattern At 50 kHz Note: db normalized to on-axis response.
Note: Curves are representative only. Individual responses may differ.

Fig. 3.6. Measured beam pattern of the Polaroid transducer [from Pola85].

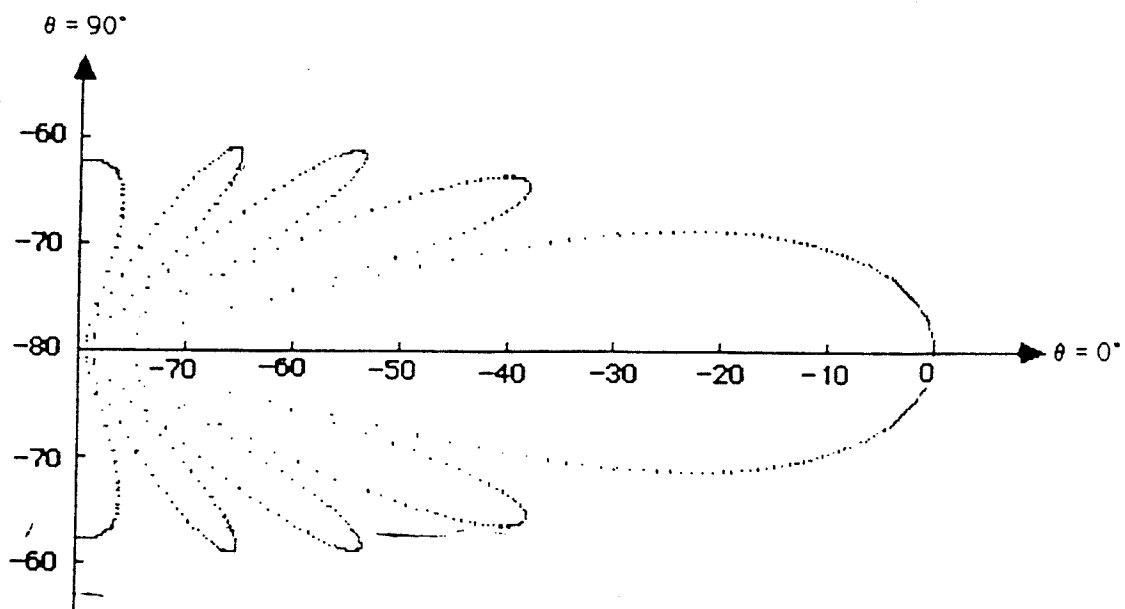


Fig. 3.7. Beam pattern of the ultrasound intensity of a transducer with $ka = 14.80$.

section. By combining Eqs. (3.7), (3.10a), the acoustic intensity $I(r, \theta)$ becomes

$$I(r, \theta) = \frac{(P_{ax}(r))^2}{\rho_0 c} \left[\frac{2J_1(k a \sin \theta)}{k a \sin \theta} \right]^2 \quad (3.12)$$

and the angular attenuation of the acoustic intensity, G_θ , for a constant distance r becomes

$$G_\theta = 20 \log(I(r, \theta) / I(r, 0)) \quad (3.13)$$

The angular attenuation of the acoustic intensity are calculated and shown in Fig. 3.7. The beam is evaluated once and for all in a table in order to speed up further calculations. As it can be seen from Fig. 3.7, for sound intensities above 35 dB there is a single value of θ . Since the dynamic range of the receiver is 35 dB and the amplitude of the side lobes of the beam, which are shown in Fig. 3.7 is smaller than 35 dB, therefore, the beam which will be used for tracking will not contain the side lobes. The dynamic range of the receiver 35 dB will be justified in the next section.

3.1.4. Volume of a Beam

The volume of a beam is defined as the volume in which the receiver will receive a minimal level of ultrasound signal. If the strength of the ultrasound signal will be below the minimal level, the signal processor will not be able to detect the burst of ultrasound pulses which is defined in Sec. 2.2, and to send an echo pulse to the control processor. This means that if the transmitter will be outside the volume of the beam of the transducer, the receiver circuits of the transducer will not get enough signal which is needed to find the distance of the

transmitter from the receiver.

The acoustic intensity is calculated from equation (3.7), and is inversely proportional to r^2 . In addition, the air absorbs the ultrasound waves about 1.8 dB/m at the room temperature, humidity of 37 percent, and the frequency of 50 kHz [Mass65]. From data sheets of signal processor SN28827 [Texa84], the range of the transducer is about 20 m because of the sensitivity of the receiver and the power of the transmitter. Therefore, the combined attenuation of the acoustic intensity, G , related to the distance of 20 m becomes

$$G = 20 \log \left(\frac{r^2}{20^2} \right) + 1.8(r - 20) \quad (3.13a)$$

where r is the distance of the limb from the transducer.

Since, the acoustic intensity at distances smaller than 20 m, is greater than at the distance of 20 m, therefore, for distances smaller than 20 m the tracked limb can deviate from the acoustic axis without to exit from the volume of the beam. The maximum deviation of θ for a given distance r without of going out of the beam, is obtained when $G=G_\theta$, where G is the combined attenuation, and G_θ is the angular attenuation.

For short distances the acoustic intensity which is evaluated by Eq. (3.13a) is greater than 35 dB from the ultrasound intensity at the distance of 20 m. Since, the dynamic range of the receiver in the SRM which is calculated from Fig. 3.8, and is $12/0.23$, that equals to 34.35 dB, therefore, for these short distances, although the acoustic intensity is bigger than 35 dB, the deviation from the acoustic axis can not be greater from this

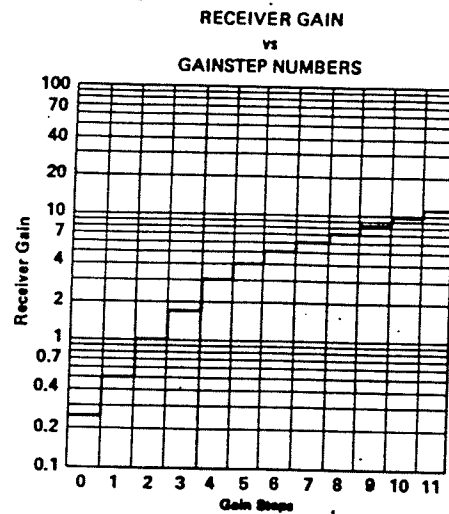


Fig. 3.8. The gain of the receiver circuit in the SRM [from Texa84].

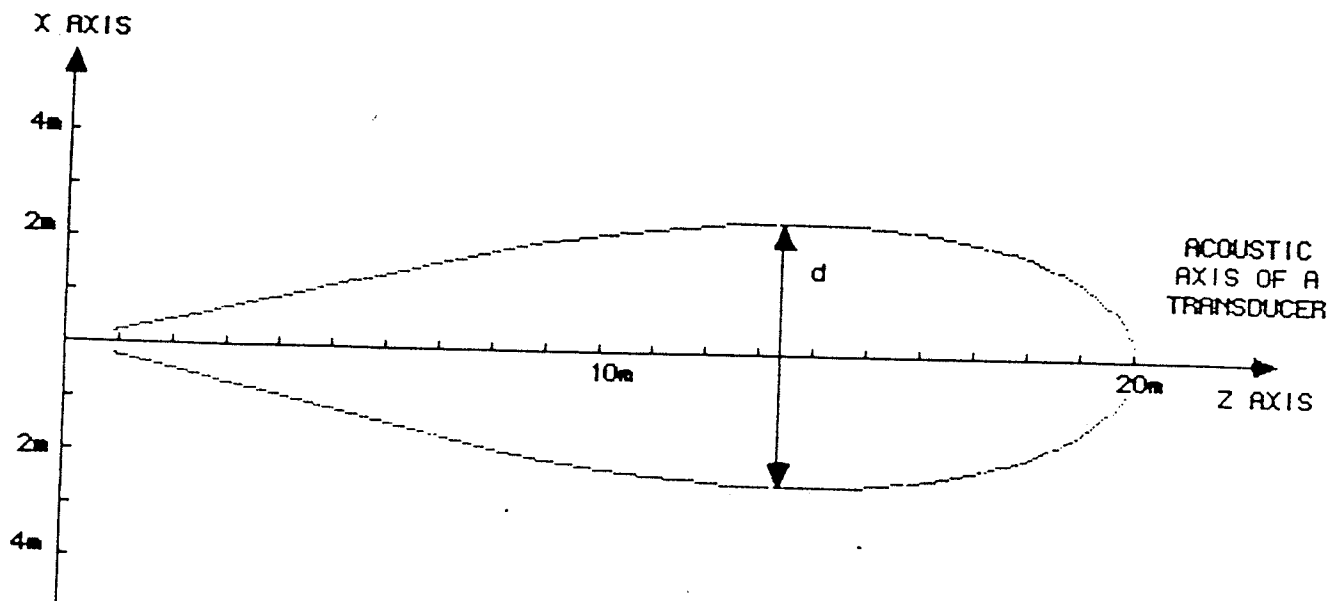


Fig. 3.9. The effective area at X,Z plane which is covered by a transducer, that is located at the origin.

which is imposed by $G_{\theta}=35$ dB.

The combined attenuation, G , for the distances from 0.9 m to 20 m is calculated by Eq. (3.13) in 5 cm steps. The values for θ the deviation angle are obtained by substituting the angular attenuation G_{θ} in Eq. (3.12) with the combined attenuation G . In order to speed up the calculations, the table which was calculated in the previous section is used to obtain θ . As earlier mentioned, if the combined attenuation is greater than 35 dB, then the value of $G_{\theta}=35$ dB is placed in Eq. (3.12). The evaluated values of θ for the related r are plotted in Fig. 3.9, which defines the boundaries of the volume of the beam.

The volume of the beam can be calculated by rotating the shape which is shown in Fig. 3.9 around the acoustic axis. The volume of the beam is relatively big when it is compared to the volume where the tracked limb moves. The beam volume has the length of $z=20$ m where z is the distance along the acoustic axis, and maximum value of $d\cong 4.5$ m where d is the diameter of the rotating shape. The diameter of the volume near the transducer is relatively small ($d\cong 46.5$ cm at $z=90$ cm). This diameter increases proportional to z , like a cone with a solid angle of 30° , which is the width of the main lobe of the beam pattern. The reason for it is because, at this region the acoustic intensity is greater than 35 dB from the acoustic intensity at $z=20$ m, therefore, the receiver circuits are saturated, and the volume is limited only by the beam pattern of the transducer. Only at the distance of $z\cong 8$ m the beam stops to increase proportional to z , because of the acoustic attenuation the ultrasound signals are now in the dynamic range of the receiver. As the values of z are greater the ultrasound signals become smaller, and the beam becomes narrower.

As it will be shown in the following sections most of this big volume can not be utilized for tracking purposes.

3.2 Volume of the Array

As earlier mentioned the volume of an array is calculated from the cross sections of the beams of the transducer array. At first the volume is calculated for an array where there are no limitations for the target movement, and the target can be located far away from the array, thus the common volume of the beams is relatively large. Since, the array of the transducers is attached to a reference frame, and the limb has a characteristic movement, therefore, the volume of the array of the transducers is evaluated for a target which its movement is limited by the reference frame volume, and by the characteristic of the limb movement. Finally, the decrease in the volume of the array as a result of the beam pattern, and the tilt of the transmitter, is calculated.

3.2.1 Common Volume of an Array

In order to calculate the volume of the array where there are no limitations on the limb movement, an experimental array is arranged. This array contains three transducers which are placed at the distance of 2 m between them, and are located on the same axis. This axis is perpendicular to the acoustic axis of the central transducer. The two additional transducers are tilted so that their acoustic axes intersect with the main transducer acoustic axis at the distance of 10 m, which is half of the length of the beam of a single transducer.

The common volume of the beams can be calculated by first drawing the projections of the three beams pattern on a plane which contain the acoustic axes of the three transducers, then calculating the common area which is covered by the beams of the transducers, and finally, calculating the volume by rotating the common area around the acoustic axis of the central transducer. The projection of the beam of each transducers is calculated in the same way as it is calculated in Sec. 3.1.4 for a single transducer. In addition, the beam of the transducer is shifted to the location of the transducer and tilted in the same angle as the transducer is tilted.

As shown in Fig. 3.10, the shaded area denotes the common area of the beams, and it resembles two triangles with the base length of $b \cong 4$ m and the heights of $h_1 = 5$ m, and $h_2 = 9$ m. When these two triangles are rotated they produce two cones, which their summed up volume V is

$$V = \pi r_1^2 (h_1 + h_2) / 3 \quad (3.14)$$

where $r_1 \cong 2$ m is the radius of the common base of the two cones. By substituting in Eq. (3.14) the values of r_1, h_1, h_2 with the values which are measured from Fig. 3.10 the volume of the two cones is calculated to be more than $V = 58 \text{ m}^3$.

The range of distances $4.5 \text{ m} > z > 18.5 \text{ m}$ (where z is the distance of the target from transducer Rx.#1) is not applicable for tracking, because it is located outside the reference frame. Therefore, most of the volume which was calculated by Eq. (3.14) can not be utilized.

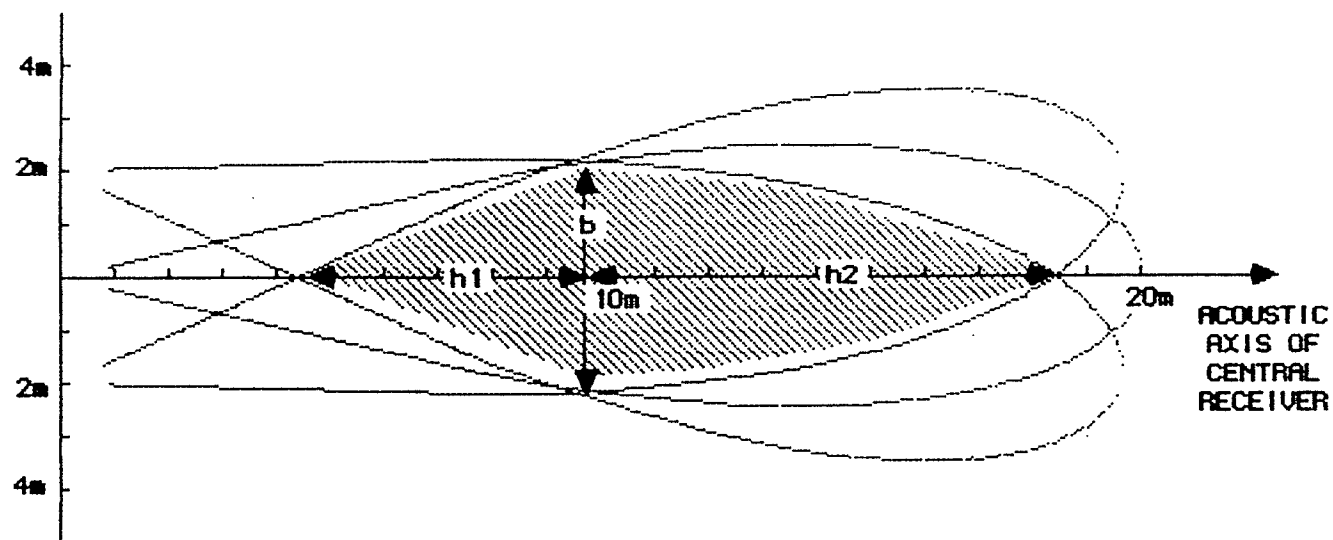


Fig. 3.10. The common area which is covered by the three receivers.

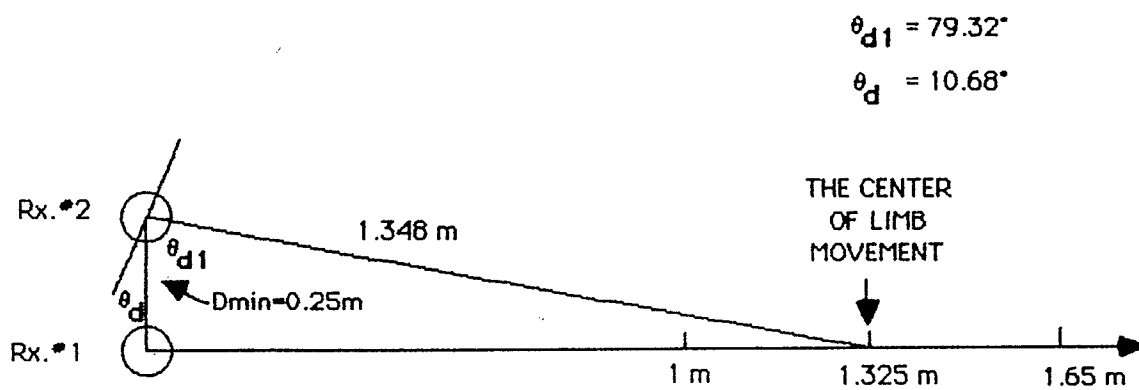


Fig. 3.11. Calculating the tilting angle of Rx#2 from the axis of the movement.

3.2.2 Selecting the Locations of the Transducers

In order to track of the limb movement which is defined in Sec. 1.1 a suitable array was selected. Since, the movement of the limb is in general one axis movement, therefore, the patient will be placed in such a position that the axis of his limb movement will be as close as possible to the acoustic axis of the central transducer of the array. In order to prevent problems with the receiver saturation, the array is located at a minimal distance of larger than the 80 cm from the limb. Actually, most of the limb movement occurs at distances further than 1 m from the array.

The location of the second transducer is designated by the fact, that at the distance of $Z_{min}=1$ m from the array the beam pattern of the transmitter has to cover at least the central and the second transducer. Since, the width of the transmitter beam is $\theta_v=30^\circ$, therefore the distance of the second receiver from the central receiver, D_{min} , can be calculated by the trigonometric equation of

$$D_{min} \leq Z_{min} \tan(\theta_v/2) \quad (3.15)$$

D_{min} is evaluated by substituting in Eq. (3.15) Z_{min} and θ_v with the previous mentioned values, and equals to 26.78 cm. For reasons of convenience, the distance of $D_{min}=25$ cm is selected.

As mentioned earlier in Sec. 1.1, the assumption is that the hand cannot displace more than 65 cm, therefore, the largest distance of the array from the limb will be 1.65 m and the centre of the limb movement will be at $z=1.325$ m. In order to get better reception from the array it is useful to tilt the receivers in the direction of the axis of the limb movement, and that their

acoustic axis will intersect with the center of the limb movement. As shown in Fig. 11, the distance of Rx.#2 from the centre of the limb movement can be evaluated by Pythagorean theorem and equals to 1.348 m. By using elementary trigonometric relations, the value for the angle $\theta_{d1}=80.32^\circ$ is calculated. Since, the sum of θ_d the tilt angle of the transducer Rx.#2 and θ_{d1} produces a right angle, then $\theta_d=10.68^\circ$.

The second receiver is placed on the X axis at the location of 25 cm. For reasons of symmetry three additional receivers are added. One is placed on the X axis at -25 cm, and the two other are placed on the Y axis at ± 25 cm. All the three transducers are tilted toward the centre of the limb movement in the same way as Rx.#2 is tilted.

3.2.3 Beam Intersection Due to Three Transducers on One Axis

At this section the volume of the three transducers which are located on X axis is calculated. The calculation is done using the same method as is used in Sec. 3.2.1, except that the locations, and tilt of the two other transducers which are on the X axis, are the same as described in the previous section. In addition the volume is calculated for the distances of $0.8 \text{ m} \leq z \leq 1.65 \text{ m}$, where the estimated path of the limb movement is.

The projections on X,Z plane and the common area of the three beams which is marked as the light shaded area are shown in Fig. 3.12. As shown in Fig. 3.12, the common area resembles three isosceles trapezoids which the first one is created by Rx.#2 lower and Rx.#3 upper boundaries, the second (very small) by the boundaries of Rx.#1, and the third by Rx.#3 lower and Rx.#2 upper

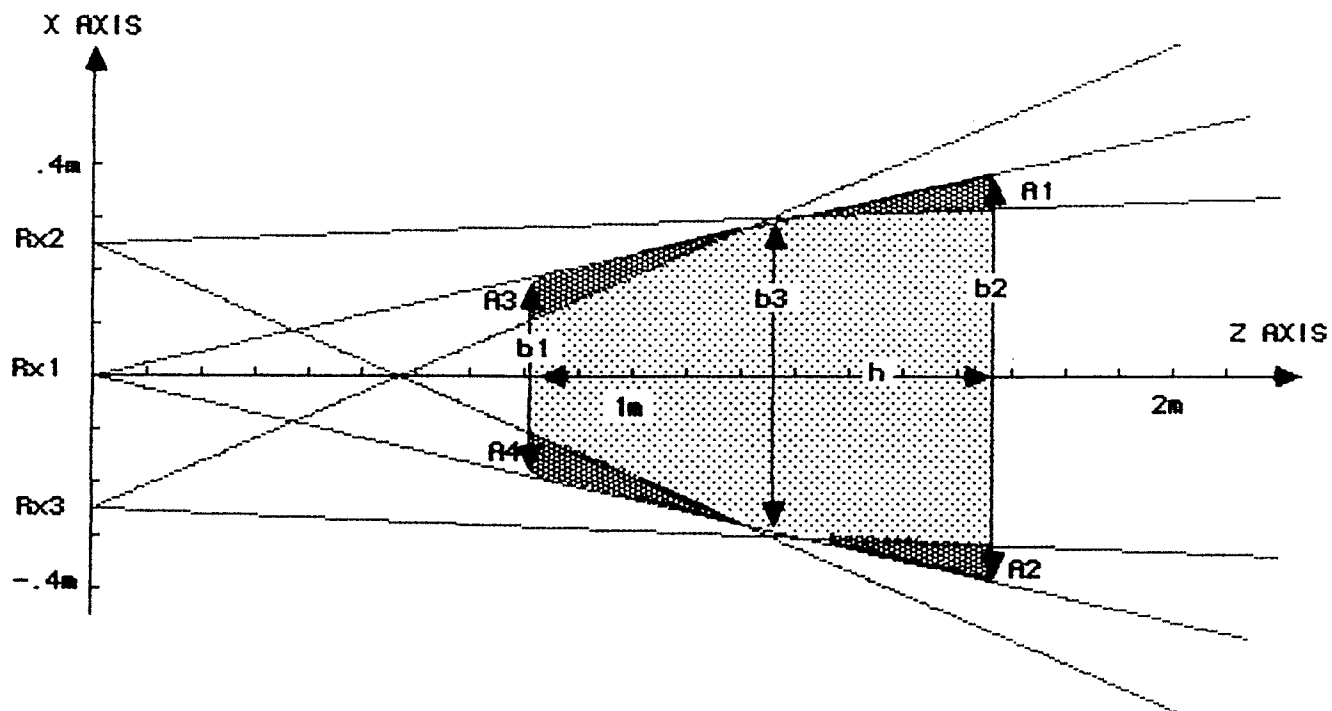


Fig. 3.12. The common area which is covered by three, and two receivers.

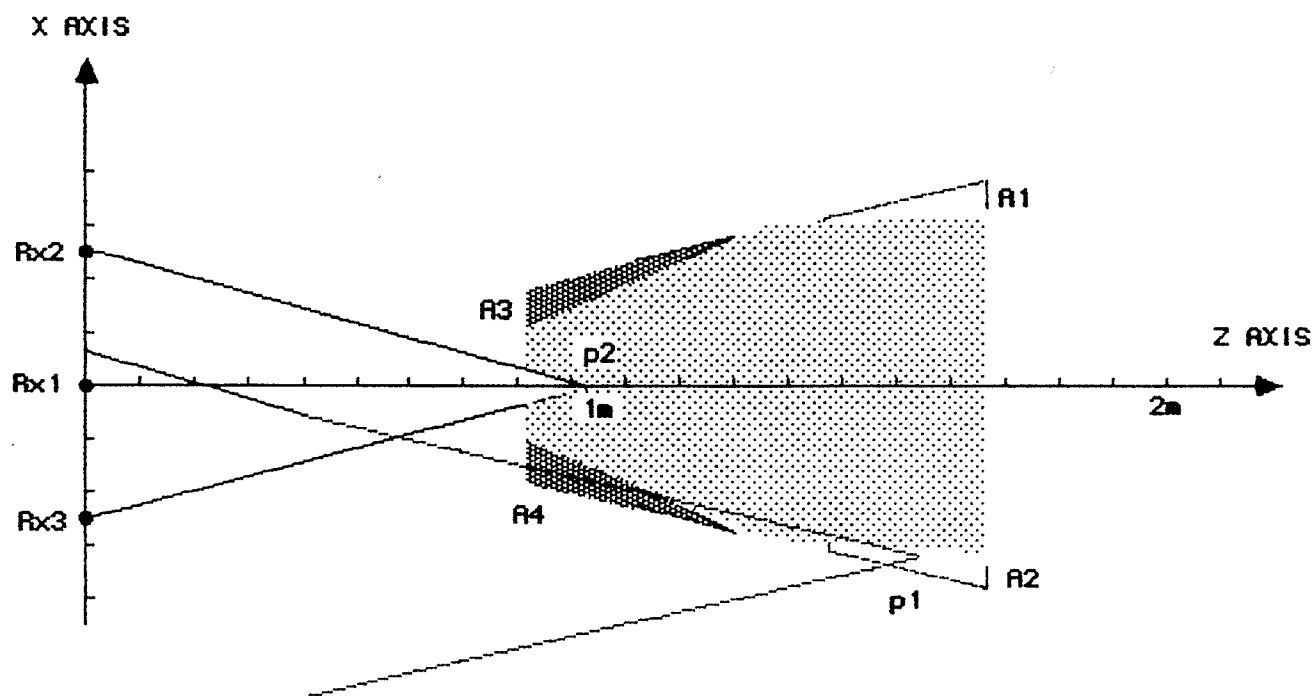


Fig. 3.13. The effect of the transmitter beam pattern on the volume.

boundaries. The common volume is calculated by rotating the three trapezoids around the Z axis and produce three frustums. The summed up volume of these three frustums is evaluated and equals to 0.25 m^3 . This volume is relatively small compared to the one which was calculated in the previous construction.

3.2.4 Redundancy Effect

As described in Sec. 5.3, a target can be located only by three receivers that are not located on the same axis. Since, Rx1, Rx2, and Rx3 are located on the same axis an additional transducer is needed. For reasons of symmetry two additional transducers are selected and placed on Y axis, and create with Rx1 a similar common beam pattern to this which is created in the previous section by the transducers on the X axis. Because the coverage of the third transducer is achieved by a transducer which is located on Y axis, the third transducer on X axis is redundant and the tracking volume can be calculated by finding the common area which is covered only by two transducers.

The common area which is covered by two out of the three receivers is shown in Fig. 3.12, and the additional areas which are added to the calculated area in the previous section are marked by the heavily shaded dots. The summed up area resembles a isosceles trapezoid with height of $h=85 \text{ cm}$, and bases with the length of $b_1=42.89 \text{ cm}$ and $b_2=88.42 \text{ cm}$. The volume is calculated by rotating the trapezoid around Z axis and equals by rounding to 0.30 m^3 .

The common volume of the three transducers which are located on the X axis produces a frustum which is symmetrical to the Z axis. In addition, the two additional transducers Rx.#4 and

Rx. #5 that are attached to the Y axis have the same distance from Rx. #1 and are tilted toward the centre of the limb movement as Rx. #2 and Rx. #3, therefore, the common volume which is created the beams of Rx. #1 ,Rx. #4, and Rx. #5 is the same which created by the beams Rx. #1 to Rx. #3. For this reason, each point which is covered by two transducers which are attached the X axis is covered by a third transducer which is attached to the Y axis, or vice versa each point which is covered by two transducers which are attached the Y axis is covered by a third transducer which is attached to the X axis.

3.2.5 Effect of the Beam Pattern of the Transmitter

In the previous section we calculate the tracking volume without the effect of the beam pattern of the transmitter. Since the transducer is reversible the same beam pattern which was calculated in Sec. 3.1.4 is used for calculating the effect of the beam pattern of the transmitter.

As shown in Fig. 3.12, the marked area A2 is covered by Rx. #2 and Rx. #1 receivers. In addition, it can be observed in Fig. 3.13 point P1 that if the transmitter is placed in the area A2 then its beam pattern does not cover Rx. #2. For this reason the marked area A2 is not effective for tracking. Similarly, it can be proven that the area which marked as A1 is not effective for tracking too. In addition as shown in Fig. 3.13, when a transmitter is placed in the triangle which its defined by the points Rx. #2, p2, and Rx. #3 then the transmitter covers only the receiver Rx. #1, therefore, this area is not effective for tracking either. The part of this triangle from P1 till the line $z=80$ cm which is covered by the receivers and not by the transmitter is called the small triangle. When the small triangle is rotated

around Z axis a cone is generated and this cone is called the small cone.

The tracking volume is calculated by rotating the shaded areas in Fig. 3.13 around Z axis. As shown in Fig. 3.12, this volume can be calculated by summing up the frustums which is created by: rotating the trapezoid which is limited by b_2 , b_3 , and the outer rays of Rx.#3 and Rx.#2, with the volume of the frustum which is created by: rotating the trapezoid which is limited by b_1 , b_3 , and the rays of Rx.#1, and deleting the volume of the small cone. This volume, which is the tracking volume for an array with an untilted transmitter is evaluated, and equals by rounding to 0.27 m^3 .

3.2.6 Tilting the Transmitter

In the previous section the tracking volume was calculated for the case where the acoustic axis of the transmitter was parallel to the acoustic axis of the central transducer. This volume varies and becomes smaller when the transmitter tilts. There are two cases of tilt which can be encountered. The first one is when the transmitter deviates from the main acoustic axis and tilts toward the array. In this case if the tilt angle is smaller than 15° , then the reference volume is included in the tracking volume. The second case is where the transducer tilts in the outside direction of the array, therefore, some parts of the reference volume are not included in the tracking volume. Since, this case is crucial for the tracking system all the calculations in this section are related to the second case.

The largest fading in the volume is when the transmitter tilts in 15° . In this case the inner side of the beam of the

transmitter is parallel with the acoustic axis of the central transducer, thus, any deviation of the target from Z axis causes that Rx.#1 is outside of the beam pattern of the transmitter. In this case the tracking volume will be limited to the Z axis and practically will be zero, and any tilt above 15° will cause the loss of tracking. As shown in Fig. 3.14, when the tilt angle of the transmitter $\theta_{d1} \leq 15^\circ$ then the transmitter can deviate from the Z axis in the out direction until the inner side of the transmitter beam intersects with Rx.#1. In addition the maximum deviation distance, dX, which is allowed for a certain distance Z and tilt angle is calculated from Fig. 3.14 by

$$dX = Z \tan \varepsilon = Z \tan (15^\circ - \theta_{d1}) \quad (3.16)$$

where the value of ε is calculated from two geometric equations that are: $\alpha + 15^\circ = 90^\circ$, and $\varepsilon + \alpha + \theta_{d1} = 90^\circ$.

As calculated in Eq. (3.16), for a given deviation angle θ_{d1} the value of the maximum deviation from Z axis, dX, is proportional to the distance Z, therefore, the volume of the tilted transmitter is a cone that its base is parallel to $z=0$ plane, its edge is at the location of Rx.#1 and forms a solid angle of 2ε . Actually, we are interested only in the part of the cone which is limited by $0.8 \text{ m} \leq z \leq 1.65 \text{ m}$, and is called the Tx. frustum. The intersection of the Tx. frustum with the reference volume is the tracking volume for a tilted transmitter.

As shown in Fig. 3.15, the tracking volume is calculated for three regions of θ_{d1} . At the first region, where $\theta_{d1} \leq 2.219^\circ$ the tracking volume can be calculated by summing up the volumes of two frustums, and subtracting the volume of the small cone which is calculated in the previous section. The boundary between the

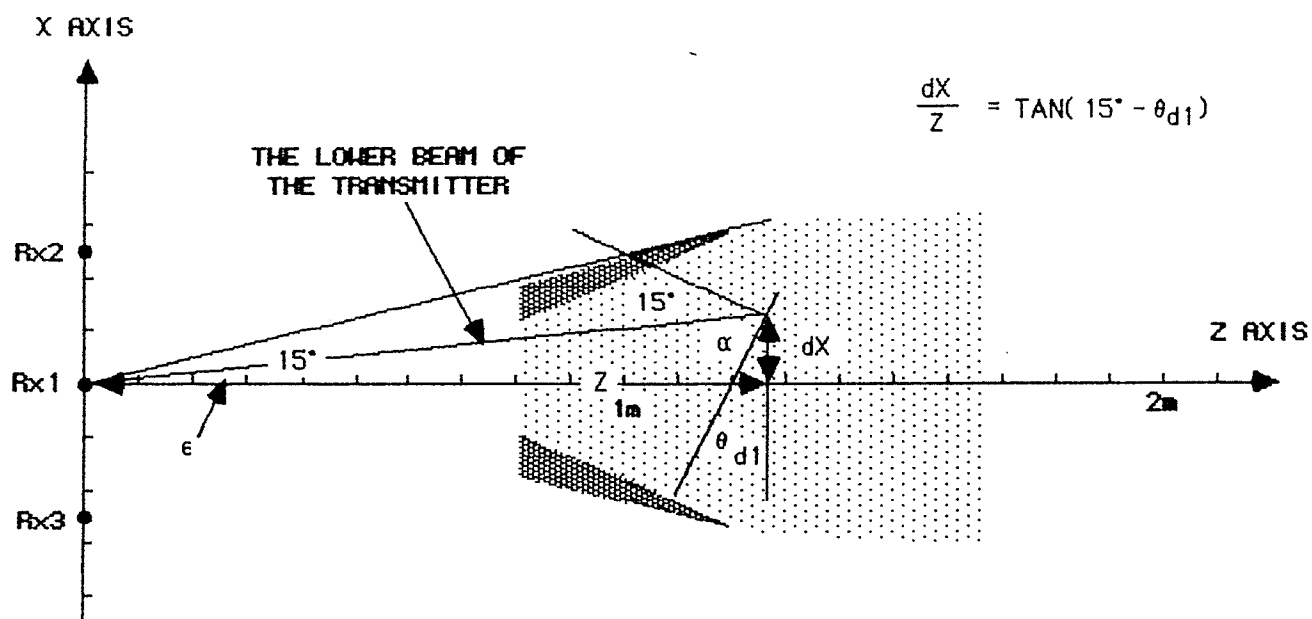


Fig. 3.14. Calculation of the maximal deviation dX from Z axis, as function of θ_{d1} (the tilt angle of the transmitter).

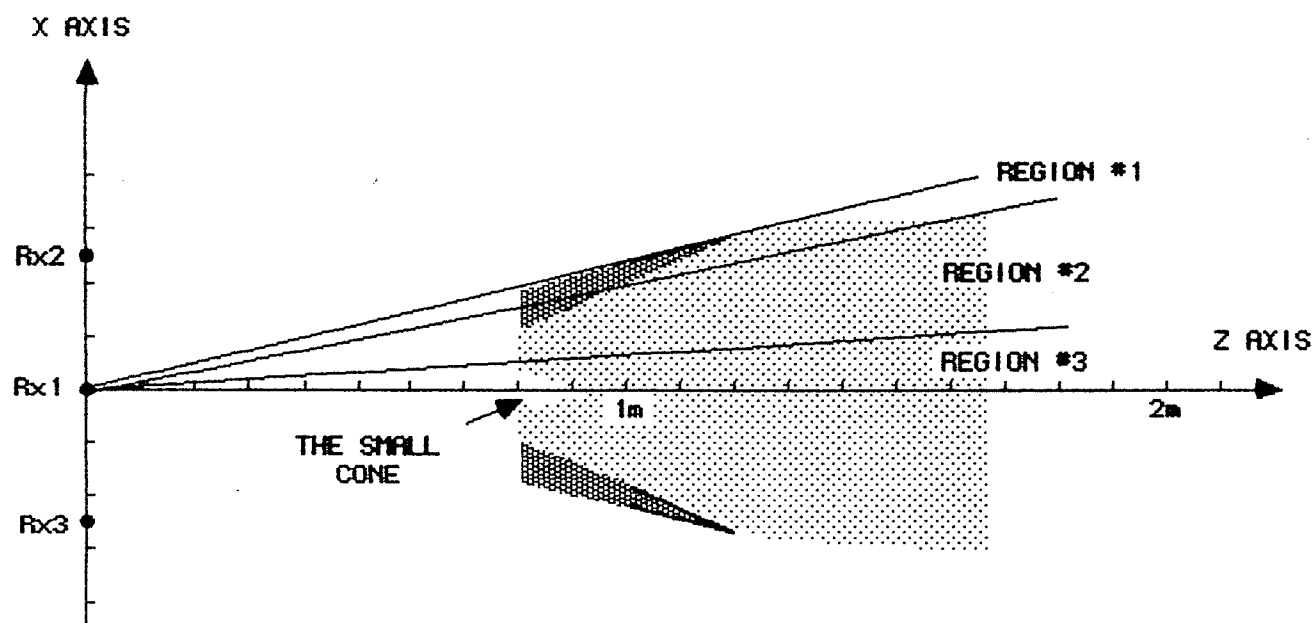


Fig. 3.15. The three regions for calculating the decrease in the tracking volume.

two frustums is the value of z where the calculated value of dX intersects with the boundary of the shaded area in Fig. 3.15. The volume of the first of the two frustums equals to the relative part of the Tx. frustum when it is limited between 80 cm to z , and the volume of the second frustum is the relative part of the reference volume when it is limited between z to 1.65 m. The second region is defined where the tilt of the transmitter is $12.449^\circ > t_{d1} > 2.219^\circ$, and in this region the tracking volume is calculated by subtracting from the volume of the Tx. frustum the volume of the small cone. The third region is defined when the tilt of the transmitter $\theta_{d1} \geq 12.449^\circ$, and in this region only the relative part volume of the small cone is deleted from the Tx. frustum.

This method of calculating the tracking volume for a tilted transducer is implemented by a computer program, and graph of the tracking volume of the tilted transmitter normalized with the reference volume, is shown in Fig. 3.16.

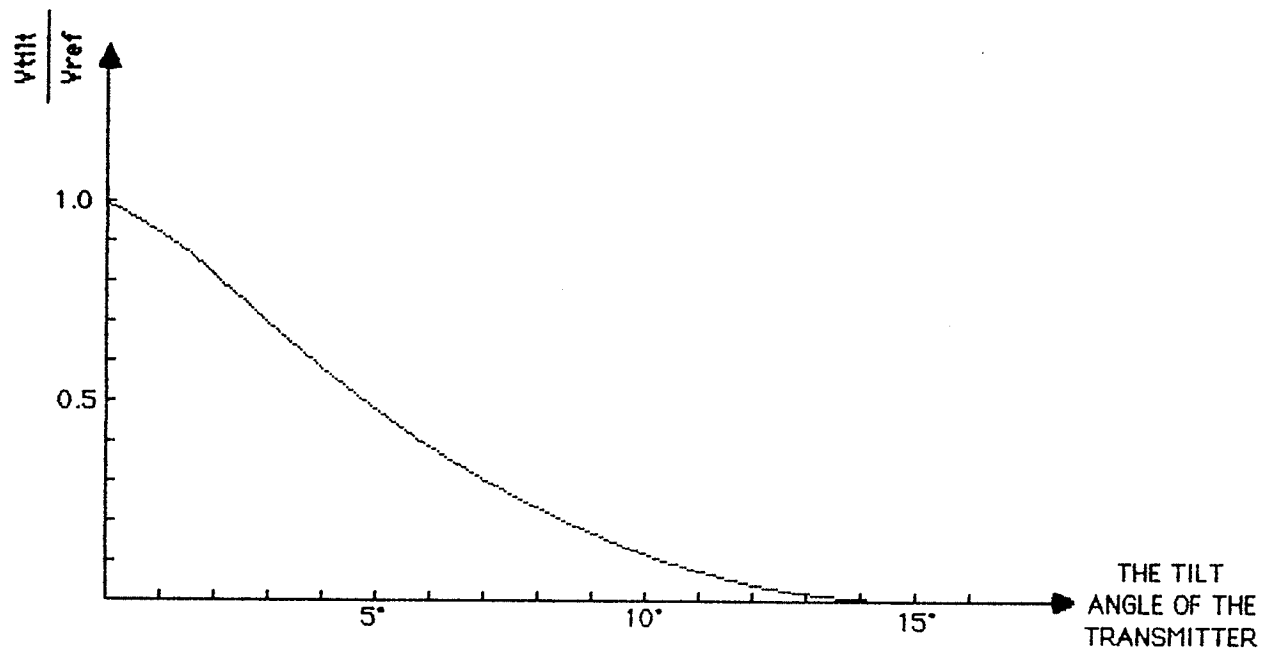


Fig. 3.16. The normalized decrease in the volume as function of the tilt of the transmitter.

CHAPTER IV

SYSTEM ARCHITECTURE AND HARDWARE DESCRIPTION

The tracking system is based on a transmitter and an array of ultrasound receivers, which are controlled by a computerized system. As shown in Fig. 4.1, the system consists of:

- (i) A single ultrasound transducer which is attached to the tracked limb (the target), and transmits a burst of 16 ultrasound pulses.
- (ii) An array of five ultrasound transducers, which receive the burst of ultrasonic pulses, and are attached to a rigid reference frame. The justification for the placement and the tilt of the transducers are described in Sec 3.2.2. The signals which are received by there transducers are used to measure the distances of the target from the transducers.
- (iii) Each of the ultrasound transducers is connected to a signal processor which is controlled by the control processor. When the signal processor of the transmitter is triggered by the control processor, it generates the burst of pulses for the ultrasound transmitter. Each signal processor in a receiver contains an analog receiver circuit. When the receiving transducers receive a burst of pulses, the signal processor generates an echo signal which terminates the time measurement (in the control processor) that started when the transmitter was triggered.
- (iv) The control processor is controlled by the main computer. Its main tasks are: to execute commands which are received from the main computer, and to provide the hardware and the software which is needed to control the signal processors. The hardware of

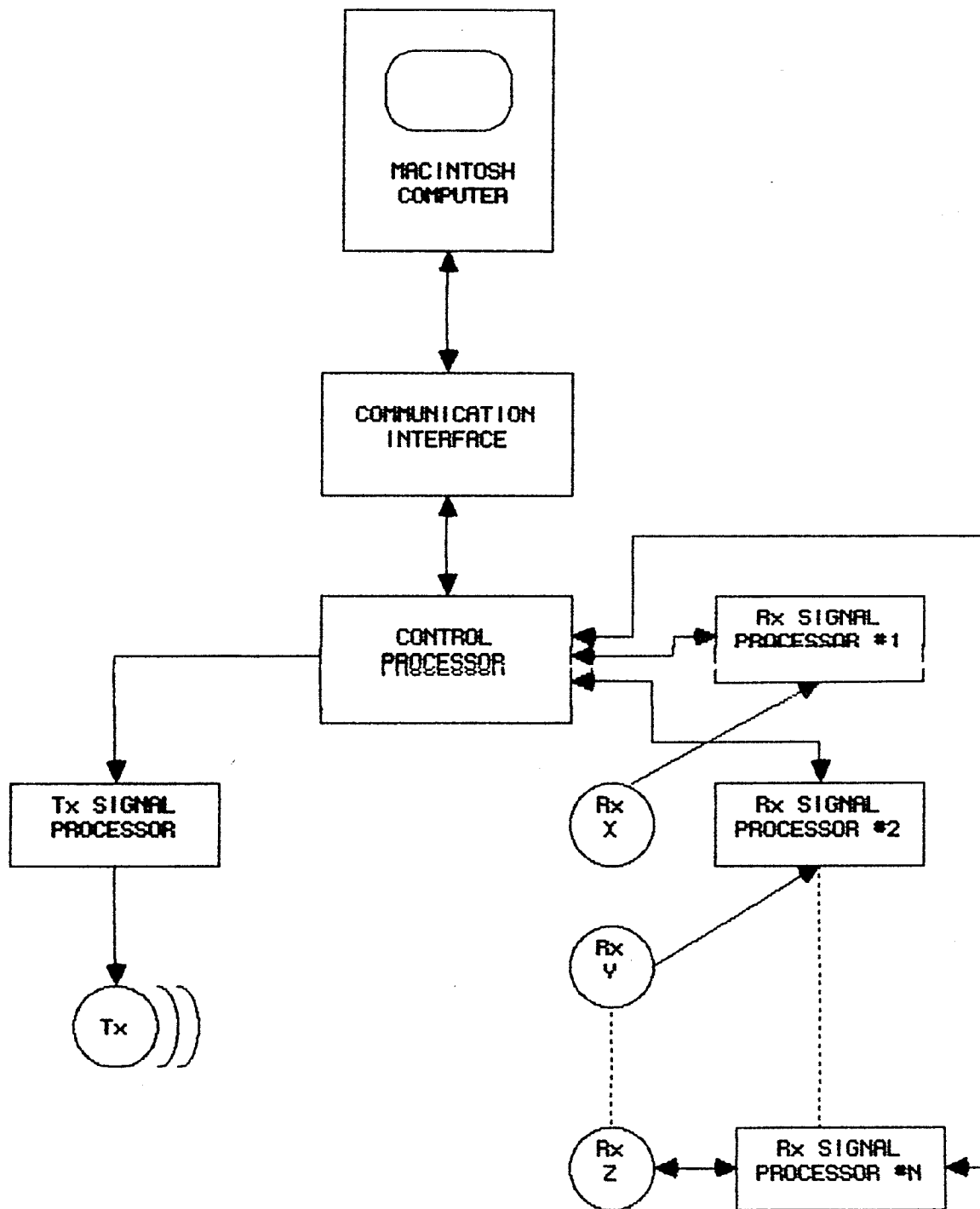


Fig. 4.1. A functional diagram of a 3D tracking system.

the control processor is described in Sec 4.2, and its software is described in Sec 5.1.

(v) The main computer provides main control of the tracking system, data acquisition, trajectory processing, and interfacing to the user. The hardware of the main computer is described in the next section, and the software in Sec 5.2.

4.1 Selection of Main Computer










































The main computer, Macintosh, has been selected for the following capabilities:

4.1.1 Graphics Capability

The Macintosh computer is a graphics oriented computer, with a built in high resolution screen of 512x350 pixels. As shown in Fig. 4.2, the graphic capability is strongly supported by graphics primitives which are even better than those of the Amiga computer, which is a graphics oriented computer. These primitives are stored as subroutines in the ROM of the Macintosh, and are called QuickDraw [Webb86]. The high graphics capability can be utilized for displaying the 3D trajectory of the limb movement.

4.1.2 Communication Rate

As previously pointed out, the distance measurement frequency of the tracking system is 80 Hz. In addition, as described in Sec. 4.2 the distance is measured by Programmable Timer Modules with the range of 16 bits, which are transmitted to the main computer as two ASCII characters. Since, each ASCII character is transmitted by 10 bits at least, the minimal

							text	
draw	 			 	 		 	
fill	 				 			
invert	 			 	 		 	



 = Amiga
  = Macintosh

Fig. 4.2. A comparison of the graphics primitives supported by Amiga and Macintosh computers, after [Webb86].

communication rate of the system, Ra_{min} , can be calculated by

$$Ra_{min} = 10(2n+2)F_{dr} \quad (4.1)$$

where n is the number of the receivers in the array, and F_{dr} is distance measurement frequency. The value of $Ra_{min}=11200$ baud is calculated by substituting in Eq. (4.1), the value n with 6, and the value of F_{dr} with 80 Hz.

Since the rate of 11200 baud is not a standard rate, the rate of 19200 baud is selected for communication between the main computer and the control processor. The structure of the communication strings is defined in Sec. 5.1.3, and Eq. (4.1) contains two additional characters which are added to synchronize the data. Since Macintosh computer contains a sophisticated communication controller with a buffer, therefore when the MS BASIC interpreter is used it is possible to communicate at the rate of 19200 baud and to have a buffer with the size of 32 Kbyte. These numbers are higher then the 9600 baud and 256 byte buffer length restrictions, which can be obtained when the standard Microsoft basic interpreter is used with the IBM PC.

4.1.3 Selection of Microprocessor

The microprocessor of the main computer has to manage with the calculations that are used by the locating algorithm (and are described in Sec. 5.3). In addition, because in certain cases the distance measurement results are jittering (as it is proved in Sec. 6.2.2), there is a necessity to store the data which arrives from the control processor for post processing programs which will cleanup the erroneous data by heuristic assumptions. When the distance measurement frequency equals 80 Hz the data arrives at

the rate of 672000 bytes/min and the system requires an access to a large memory.

The MC68000 microprocessor which is used by Macintosh computer, is compared in TABLE 4.I to 8088 microprocessor which is used by IBM PC computer. Although today, both of the machines have newer models, when the computer was selected at the beginning of 1986, these two machines were widely used and available for selection.

TABLE 4.I
A COMPARISON OF THE MC68000 AND 8088 PERFORMANCE

	MC68000 [John86]	8088 [Inte79]
External Data Bus	16 bits	8 bits
Internal Registers	32 bits	16 bits
Address Lines	24 lines	20 lines
Clock Frequency	8 MHz	4.77 MHz
Standard Support*	4.125 Mb	640 Kb

The standard support* in TABLE 4.I refers to the size of the memory which is supported by the operating system of the Macintosh for the MC68000, and by MS DOS for the 8088.

It is obvious from TABLE 4.I, that the MC68000 is more powerful for computation and has access to larger memory. In addition the Macintosh computer has the advantages of graphical orientation and better communication port. For these reasons the Macintosh was selected as the main computer in our system.

4.2 Control Processor

As previously mentioned, the tasks of the control processor are to receive a tracking command from the main computer, execute it by controlling the signal processors, and to complete the command execution by sending the measured data to the main computer for further processing.

In order to execute these functions, the control processor consists of (i) a microprocessor (MC6802), (ii) a communication peripheral processor (MC6850), (iii) Programmable Timer Modules (PTMs) type MC6840. The block diagram of the control processor is shown in Fig. 4.3. The operation of these circuits is described in the following sections, and the indexing of the circuits refers to the electrical circuits schematics which are given in Appendix A.

4.2.1 Microprocessor Circuits Description

The MC6802 is a 8 bit microprocessor, its instruction set and buses are similar to these of the MC6800 microprocessor. The operation of the MC6800 is described by Greenfield and Wray [GrWr81]. In addition, the 6802 contains a RAM of 128 bytes and a clock circuit.

The clock circuit of the MC6802 needs an external 4 MHz crystal, which is divided (by an internal circuit) to four. This division provides a machine cycle with the duration of 1 μ sec. In addition, the divided clock is used as an input signal to the counters in the PTMs. The RAM provides the temporary storage (which is necessary for implementation of the control processor software) and is described in Sec. 5.2.

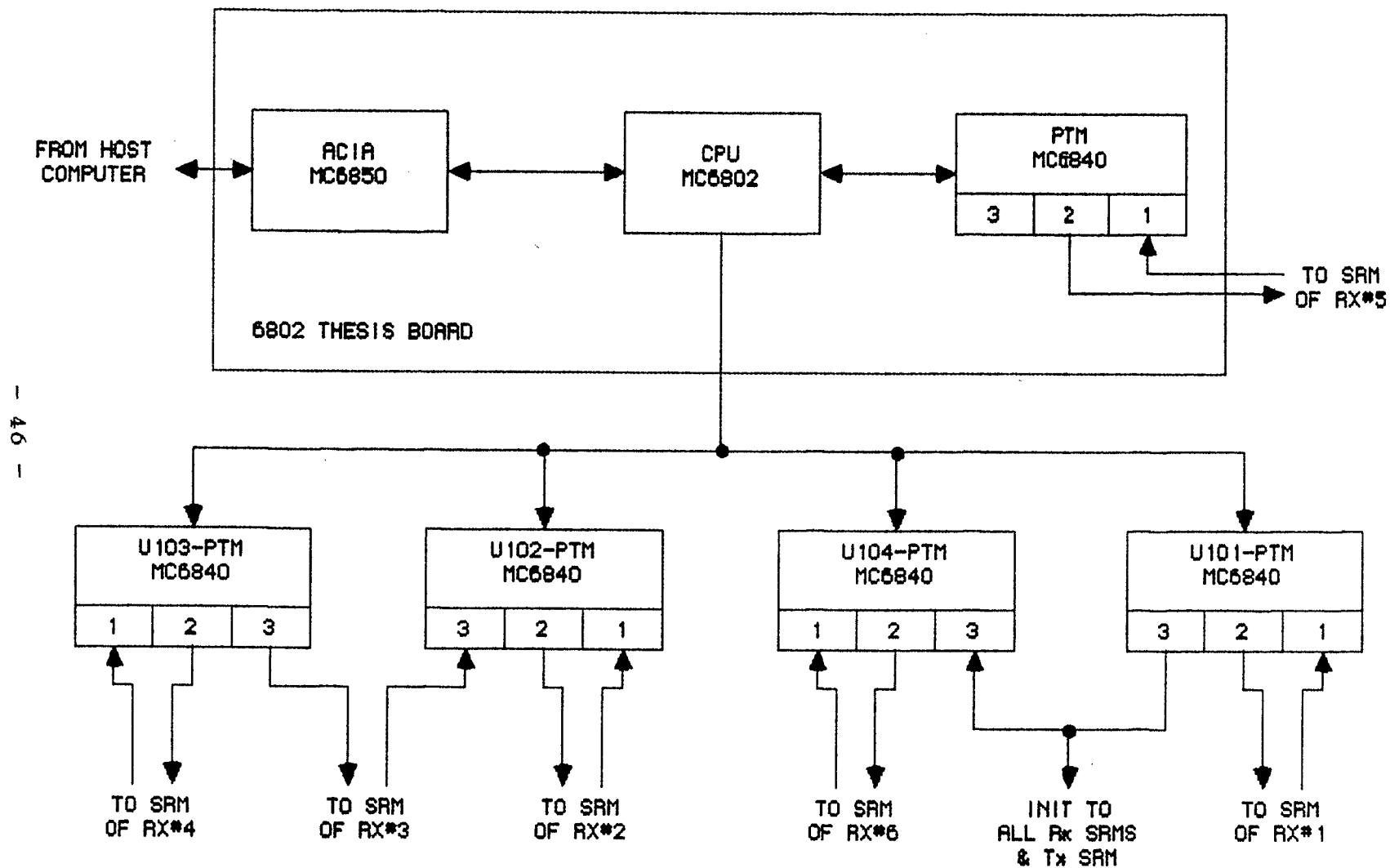


Fig. 4.3. Control processor block diagram.

The microprocessor executes instructions that are stored in a 2 Kbyte EPROM (2716). The MC6802 microprocessor with the addition of EPROM, PIA (that is not used by us), ACIA, PTM, and address decoding circuits are mounted on the same printed circuit board (6802 Thesis Board 2) which is manufactured by the University of Manitoba [UofM86].

The microprocessor receives and sends its instructions or data through the eight data lines. The device to which the microprocessor applies is determined by address decoder circuits U3 on 6802 Thesis Board 2, and as shown in Fig. A.2.1. U100 on Timers Board. The assignment of the addresses (in hexadecimal notation) for the control processor peripherals is described in TABLE 4.II.

TABLE 4.II
THE MEMORY MAP OF THE CONTROL PROCESSOR

Address	Circuit	Function
0000-007F	U9 6802	RAM
2800-2FFF	U6 6850	ACIA
3000-37FF	U5 6840	PTM
3800-3FFF	U11 6821	PIA
8000-83FF	U101 6840	PTM
8400-87FF	U102 6840	PTM
8800-8BFF	U103 6840	PTM
8C00-8FFF	U104 6840	PTM
F800-FFFF	U10 2716	EPROM

The control processor program can be interrupted by event driven signals like message arrival from the main computer, or the termination of the tracking cycle. This interruption is done through the IRQ line which connected to pin 4 of the

microprocessor.

4.2.2 ACIA Circuits Description

The main computer communicates with the control processor through U6 which is MC6850 (ACIA). In addition, U7 and U8 are used to scale down the communication signals (which are transmitted at the RS232 signal levels) to 0 and 5 V signal levels that are used by the ACIA.

The baud rate of 19200 is established by (i) supplying a clock with the frequency of 307.2 kHz to pin 3 (RxC) and pin 4 (TxC) of the ACIA, (ii) programming the ACIA to divide the incoming clock by 16. The incoming clock frequency (307.2 kHz) is achieved by dividing by 8 (with U115) the frequency of the crystal controlled oscillator U4.

The ACIA operation is determined by the information which is written in its control register. The address of the control register is any one of the even addresses in the range of addresses which is determined for the ACIA in TABLE 4.II. For reasons of convenience only the address of 2E00 hexadecimal is used for addressing the control register of the ACIA. The control register determines (by its bits 0 and 1) how many clock pulses are used to synchronize the incoming data. The bigger the number is, then the system is more noise immune. For reasons of not exceeding the maximal clock frequency which is allowed by the ACIA the mode of divide by 16 is selected when the baud rate of 19200 is used. In order to speed up the data communication the data is transferred without parity bit and with only one stop bit. In addition, since the PTM data consists of 16 bits, all the eight bits of the serial word are used for transmission. The length of

the word, the type of parity bit, and the number of the stop bits, are set by bits 2 to 4 of the control register of the ACIA.

Upon the arrival of data from the main computer the ACIA can interrupt the control processor software program, and this is done by setting bit 7 of the control register to one. Similarly, the ACIA can also interrupt the control processor software program when it finishes to transmit the data, and this is done by programming bits 5 and 6 of the control register.

The data is transferred between the ACIA and the CPU through the data register. The address of the data register is any one of the odd addresses in the range of addresses which is determined for the ACIA in TABLE 4.II. For reasons of convenience, only the address of 2E01 hexadecimal is used for addressing the data register of the ACIA.

4.2.3 PTMs Circuit Description

The control processor contains five Integrated Circuits (ICs) of PTMs which provide the functions of (i) measurement of the time delay in which the ultrasound burst propagates from the target to a receiver, (ii) supply of the BLNK pulses which are needed to prevent reverberation, (iii) as described in Sec 5.1, synchronization and setting the execution time of the search and track cycles.

In general, each of the PTMs consists of three independent 16-bit timers, which their functions are shown in Fig 4.2 and described in TABLE 4.III.

TABLE 4. III
PTMS ASSIGNMENT

Reference	Task
U5-Timer #1	Measures ECHO pulse of Rx. #5
U5-Timer #2	Provides BLNK pulse for Rx. #5
U5-Timer #3	Not used
U101-Timer #1	Measures ECHO pulse of Rx. #1
U101-Timer #2	Provides BLNK pulse for Rx. #1
U101-Timer #3	Provides INIT pulse
U102-Timer #1	Measures ECHO pulse of Rx. #2
U102-Timer #2	Provides BLNK pulse for Rx. #2
U102-Timer #3	Measures ECHO pulse of Rx. #3
U103-Timer #1	Measures ECHO pulse of Rx. #4
U103-Timer #2	Provides BLNK pulse for Rx. #4
U103-Timer #3	Provides BLNK pulse for Rx. #3
U104-Timer #1	Measures ECHO pulse of Rx. #6
U104-Timer #2	Provides BLNK pulse for Rx. #6
U104-Timer #3	Provides LOAD pulse

Each of the five PTMs consists of eight 8-bit registers that control the PTM, define the width of the pulse which is generated, or accumulate the data of the time measurement. The addresses of the PTMs are determined by address decoders (U100 and U3) and are described by TABLE 4. II. Since the address decoder circuits provide access to a larger range than eight addresses (which are needed by a PTM), each register has multiple addresses. For reasons of convenience the software of the control processor uses only the first eight addresses of the range, except U5 where the addresses of 3700H to 3707H are used. The function of the PTM registers is described by TABLE 4. IV.

TABLE 4. IV
PTM REGISTERS FUNCTION

Register No.	Function
0	Control of Timers #1, #3
1	Control of Timer #2
2	MSB data of Timer #1
3	LSB data of Timer #1
4	MSB data of Timer #2
5	LSB data of Timer #2
6	MSB data of Timer #3
7	LSB data of Timer #3

The first two registers of the PTM are used as control registers for the three timers. Register #1 controls Timer #2, and Register #0 that directs its data to two registers which control Timer #1 and Timer #3. The related timer (Timer #1 or Timer #3) which is controlled by Register #0, is determined by the status of bit 0 of Register #1.

The counters in the timers can select the source of clock pulses from (i) an external source that is connected to the clock input of the timer, (ii) 1 MHz internal clock which is provided by the CPU through E input of the PTM. The selection of the wanted clock is done by programming bit 1 of the control register of the related timer. All the timers of the tracking system are utilizing the internal clock which is provided by the CPU. In addition, Timer #3 of a PTM can select with bit 0 of the control register a divide by-8 prescaler. This option is implemented by the control processor in the search mode, where (as shown by Fig. 5.1) the length of INIT pulse is 100 msec, and the counter of U101-Timer #3 has to count, 10^5 pulses with the duration of 1 μ sec. Since the number of 10^5 is greater than the size of the counter (16 bits

which equals to 65536) the prescaler of U101-Timer #3 is used.

Bit 4 of the control register determines whether the timer is a pulse generator, or measures the duration of a pulse. The BLNK, INIT, and LOAD timers are used as pulse generators, and the ECHO timers are used for pulse duration measurement.

When a timer is used for time measurement, it counts clock pulses (with its data registers) only when zero voltage is provided to its Gate input. The counting stops at the instant that 5 V are provided to its Gate input. Since the data registers of the ECHO timers are counting in descendant order, they are loaded by the software at the beginning of each tracking (or search cycle) with the maximum value which is FFFFH. The counting starts as shown in Figs. 5.1 and 5.2 at the instant that the inverted INIT pulse is supplied (through the OR gates U110, and U113) to the Gate inputs of the ECHO timers. The counting of a timer is terminated at the moment that the ECHO pulse is received from the related SRM. These ECHO pulses are supplied to the timers through the second input of the OR gates.

When the PTM is used as a pulse generator it provides a pulse through its Output pin. The output of the pulse can be disabled by programing bit 7 of the corresponding control register. In addition, the timer can be programmed (by bit 5 of the corresponded control register) as a continuous, or as a single shot pulse generator. The output of all the timers of the control processor (which are used as pulse generators) is enabled when they are invoked, and disabled when they are not active. In addition they are used as a single shot pulse generator.

There are two methods to invoke the timers to generate a pulse (i) by writing the width of the pulse to its data registers, (ii) by a negative going pulse which is applied to its Gate input. In method (ii) the width of the pulse which is generated is determined by pre-written data which is in its data registers. The selected method of invoking the timers is determined by bit 4 of the correspondent control register.

When the tracking or search cycle starts the INIT timer is invoked by method (i). On the other hand, as shown in Fig. 5.2 the start of the BLNK pulses is synchronized with the rising edge of the INIT pulse, therefore the BLNK timers are invoked by method (ii) which is implemented by the inverted INIT pulse which is provided by U111a to the Gate inputs of the corresponding timers. In the tracking mode the LOAD timer is invoked to generate a pulse with the width of 2 msec by the falling edge of the INIT pulse. The additional pulses of INIT (with the width of 10 msec) that are necessary to complete the tracking command are invoked by the falling edge of LOAD pulse which is connected to the Gate input of the INIT timer. By the cascade connection of the two timers (INIT and LOAD) and restarting the INIT timer by the LOAD timer a tracking cycle with a stable period time of 12 msec is achieved. Since the timers are invoked by the pulses which are connected to their Gate inputs and not by the act of writing to their data registers, the duration of the cycle is not dependent on the execution time of the software.

The CPU of the control processor is notified about the termination of the INIT pulse by the IRQ pulse (which is the interrupt) of the IC which contains the INIT timer (U101). This is done by setting bit 6 of the correspondent control register to level "1". In addition, the INIT and BLNK pulses which are needed

for the operation of the SRMs, are buffered by U111 and U112.

4.3 Signal Processors

The signal processor is responsible for (i) generation of the ultrasound burst which is transmitted by the target, (ii) reception of the burst from the corresponding receiver and processing the burst. These functions are performed by a modified Sonar Ranging Module (SRM) type SN28827, which is manufactured by Texas Instrument [Texa84].

4.3.1 Circuit Description

As shown by Fig. 4.4, the signal processor contains two ICs and a buffer which their functions are:

(i) U2 (TL851) is the SRM controller which provides:

Upon the receiving of the INIT pulse from the control processor it generates a burst of 16 pulses with a PRF of 50 kHz.

A variable gain control to the receiver amplifier IC (U1). The gain is increased automatically according to the time which passes from the rising edge of the INIT pulse. The reason for the variation in the gain is to compensate the loss of the ultrasound intensity when the signal propagate in the air.

An ECHO pulse to the control processor which is provided when the receiver amplifier transfers the received burst signal to the control IC. The ECHO pulse can be inhibited by the control processor with a BLNK pulse. This BLNK pulse is used to prevent the system from identifying a false target by reverberant signals

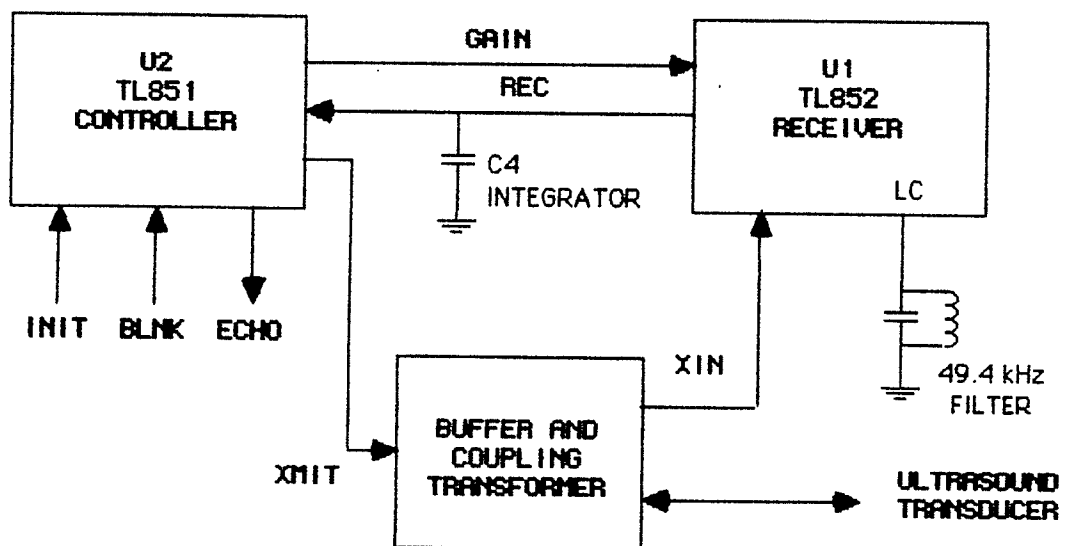


Fig. 4.4. A block diagram of a signal processor.

(that are described by Sec 2.3.3).

(ii) U1 (TL852) is the receiver amplifier of SRM. As earlier pointed out, the gain of the amplifier is controlled by U2 (The controller). An external band pass filter (type LC) which is tuned to 49.4 kHz (the frequency ultrasound pulses) is connected at the junction LC of U1. At the output of the receiver there is a capacitor (C4) which integrates the incoming pulses. This integrator is discharged frequently by a current source, and only when enough pulses are received the amplitude of the integrator is high enough to pass the triggering level of the comparator circuits (that the controller circuit U2 contains) and a ECHO pulse is generated by the comparator. By this method the system is not triggered by accidental pulses which are received by the transducer.

(iii) The buffer circuits are buffering the burst of the pulses, and amplifying it (by a transformer) to a level of 300 V. The amplified signal is provided to an ultrasound transducer which transmits the burst of pulses. In addition, the secondary tap of the transformer provides a coupling from the transducer to the receiver amplifier (U1) for the signals which are received by the transducer.

4.3.2 Modification of the SRM

In order to implement the SRM as a receiver only and to enlarge the width of the beam of the transducers, the receiving SRM is modified in the following way:

(i) The transmitter circuit is disabled by disconnecting the base of the transmit transistor in the correspondent module.

(ii) In order to increase the narrow beam pattern of the transducer the automatic gain control function of the SRM is canceled, and the gain is set to its maximum level. By this action the sensitivity of the receiver is 280 mV. In addition, the dynamic range of the receiver is increased to 35 dB, and the width of the beam (as described in Sec. 3.1.3) which is about 30° is achieved.

4.4 Ultrasound Transducers

As pointed out in Chap. III, the characteristic of the ultrasound transducer sets the boundaries of its beam and respectively the tracking volume. Therefore it is crucial to select the proper transducer for the tracking system. The ultrasound transducers are divided in to groups, according to the method that they convert the electrical energy to ultrasound waves. The main three methods which are used by the transducers in the air are (i) electromagnetic, (ii) piezoelectric, (iii) electrostatic. The method which is the most suitable for implementation in the air is the electrostatic transducer because of its good impedance matching with air [Mass65].

The operation of electrostatic transducer is based on two plates, one of them fixed and the second is a moving part (which is a thin foil). The plates are acting like a capacitor and are charged by a DC voltage in the range of 150 V to 200 V. When the plates are charged an electrostatic force is exerted on the thin foil plate. Any change in the voltage between the plates causes the foil to transform the electrical energy to sound waves. When the sound waves hit the transducer the foil moves, and the movement is transformed to electrical energy. Since the operation

of this transducer is affecting the air directly (its response is the changes in a gap of air which is the distance between the capacitor plates), the electrostatic transducer is more efficient in the air than other methods [Mass65]. In addition, the frequency response of the transducer is narrow banded around the resonant frequency of the foil, and this is advantage for the tracking system where only a single frequency is used.

Since the good matching with the air is determined by the properties of the foil, the Polaroid transducer consists of a plastic (Kapton) with the properties of noncreeping, elastic, light, and extremely thin foil [BESS80]. In addition in order to achieve good conductivity, the front plane of the foil is plated with gold. The full specification of the transducer are given by Polaroid [Pola85].

4.5 Reference Frame

As described by Sec 5.3, the target is located by measuring the distances of the target from points that their placement is known. Therefore, the placement of the transducers which are used to measure the distances has to be fixed. For this reason the reference frame (that the transducers are mounted on its beams) is fixed and rigid. The rigidness is achieved by using as beams of the frame metal pipes with the width of 1 in, and the fixation of the frame is achieved by binding together the beams of the frame with rigid clamps [KeeK80]. As shown by Fig. 4.5, the size of the of the frame is 2 m X 2 m X 1.1 m. This size contains the tracking volume and the array of transducers is attached by swivels to the two upper internal beams.

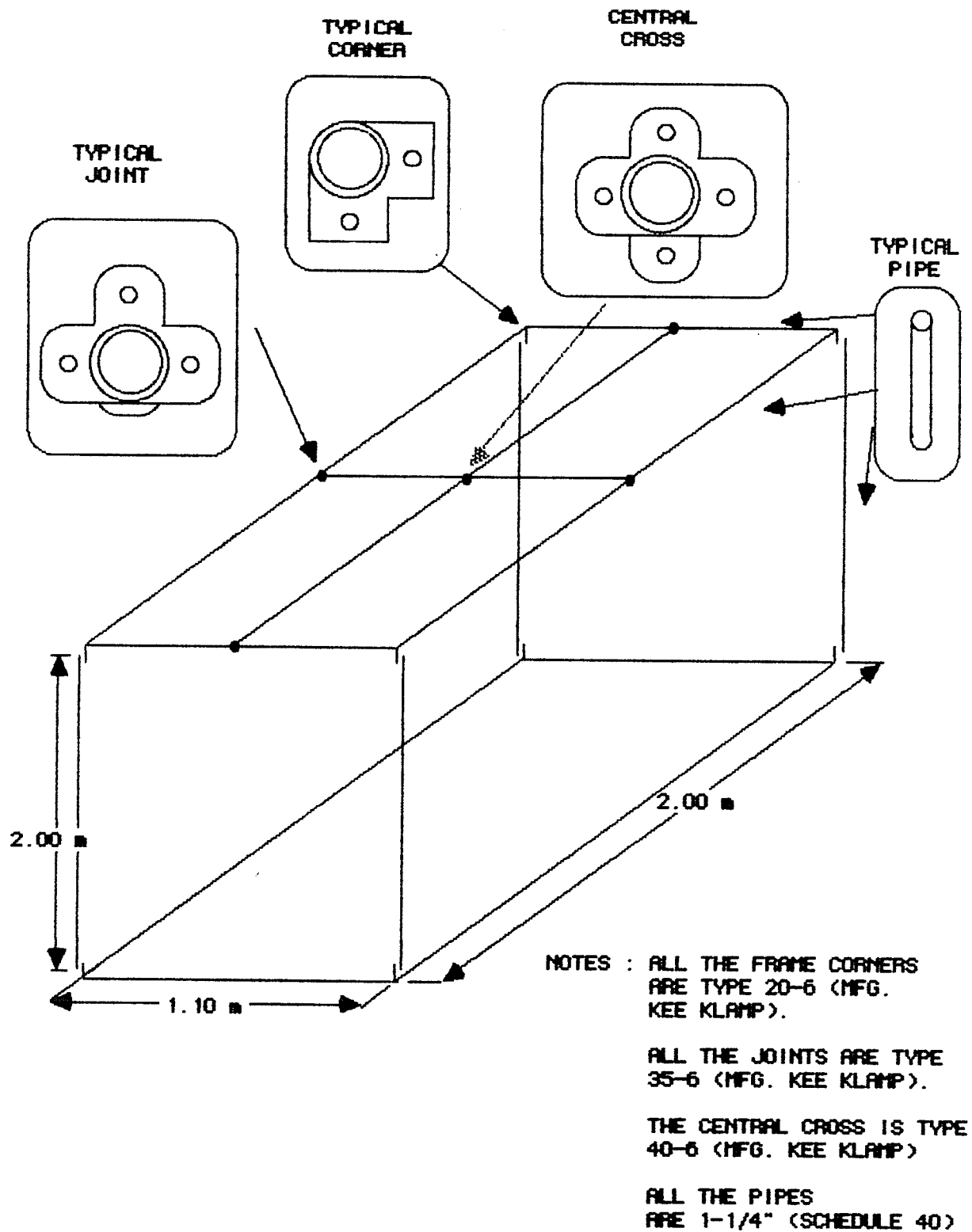


Fig. 4.5. The construction of the reference frame.

In order to obtain the tracking volume the transducers has to be positioned and tilted as described in Sec. 3.2.2. As shown by Fig. 4.6 a two parts swivel is used. The first one (part A) is a female swivel socket member which allows to adjust the position of the swivel by moving it on the correspondent beam. In addition, by rotating this part around the pipe the attached transducer can be tilted in one plane. The full 3D tilt adjustment can be obtained by additional rotation of the second part of the swivel (part B which is a male single swivel socket) around the screw which combines the two parts of the swivel. The transducer is attached to the swivel by a short and easy to work over vinyl pipe with a diameter of 1.25 in. This pipe is connected to part B of the swivel.

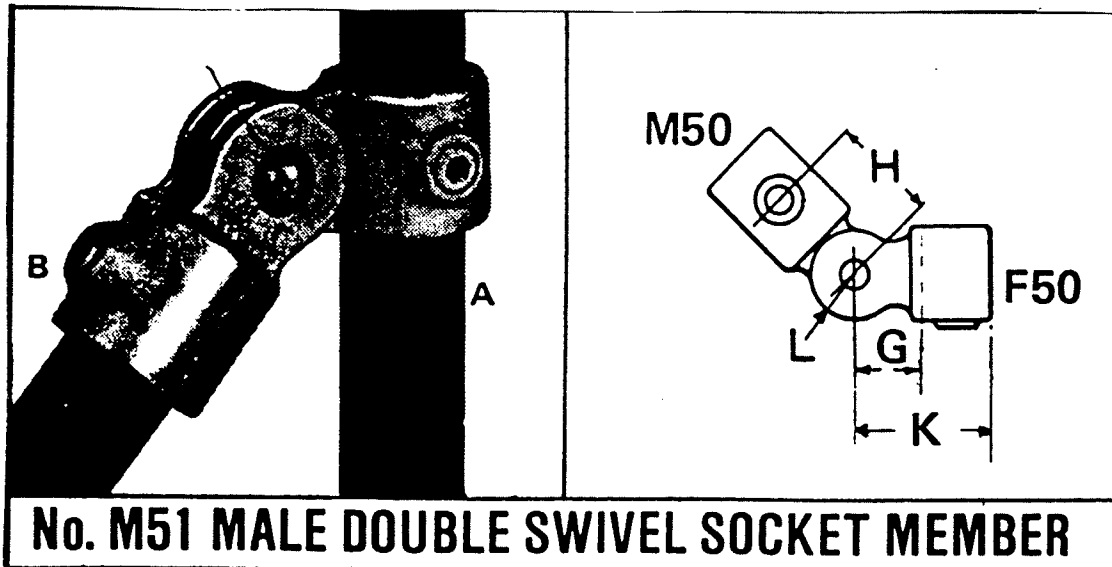


Fig. 4.6. A description of the swivel, from [KeeK86].

CHAPTER V

SOFTWARE DESCRIPTION

The software of the tracking system provides the following tasks (i) controlling the signal processors which are connected to the transducers, (ii) filtering out erroneous data that is caused by noise or reverberation, (iii) implementation of the locating algorithm which is described in Sec. 5.3, (iv) human-machine interfacing.

Because of the difference in the nature of the first task and the others, the software is divided into two major parts. The first part which controls the operation of the signal processors needs relatively fast response and almost no calculations, is written in assembly language. The second part is written on the main computer which is described in Sec. 4.1, and contains powerful microprocessor and graphics capability. Since the software of this part contains complex calculations (for locating the target and filtering the data) and the software needs strong graphics support for the human-machine interface, therefore it is written in a high level language.

5.1 Control Processor Software

As earlier pointed out, the software of the control processor is written in assembly language. This software is implemented on a separate machine because (i) The main computer (Macintosh) is a closed architecture computer and there is no direct access to the bus of the processor, (ii) The main computer is heavily loaded with computation tasks, and there is a danger that the execution of computation jobs will be affected because of the tedious routine operations of the control processor.

5.1.1 General Description

The control processor software is written in the 6800 Macro assembler language, which was developed and emulated on the Tektronix 8002 μ PROCESSOR LAB [Tekt78]. The software was written in four modules according to the main tasks of the control processor which are:

(i) Initialization module, which initializes and defines the operations of the ACIA and PTMs.

(ii) Communication module, which communicates with the main computer. This module translates the command that is received from the main computer, and invokes the proper module (track or search). When the module receives an erroneous message it transmits an error string to the main computer. When the tracking or search cycle is completed, it transmits the values which are measured by the ECHO timers to the main computer.

(iii) As shown by Fig. 5.1, the search module is a one cycle module in which the module provides a single INIT pulse. Simultaneously (as described in Sec 4.2.3) the ECHO timers start to measure the duration of the echo pulses, which are proportional to the distance of the transducers from the tracked limb. At the end of the INIT pulse (whose length is 100 msec) the module reads the echo durations, which were already measured by the ECHO timers. Because the ECHO timers count the clock pulses in descending order (in order to calculate the positive values of the time delay) the values which are obtained from the timers are binary complemented and saved in a buffer. The BLNK timers are loaded with the corresponding values, which are the values of the

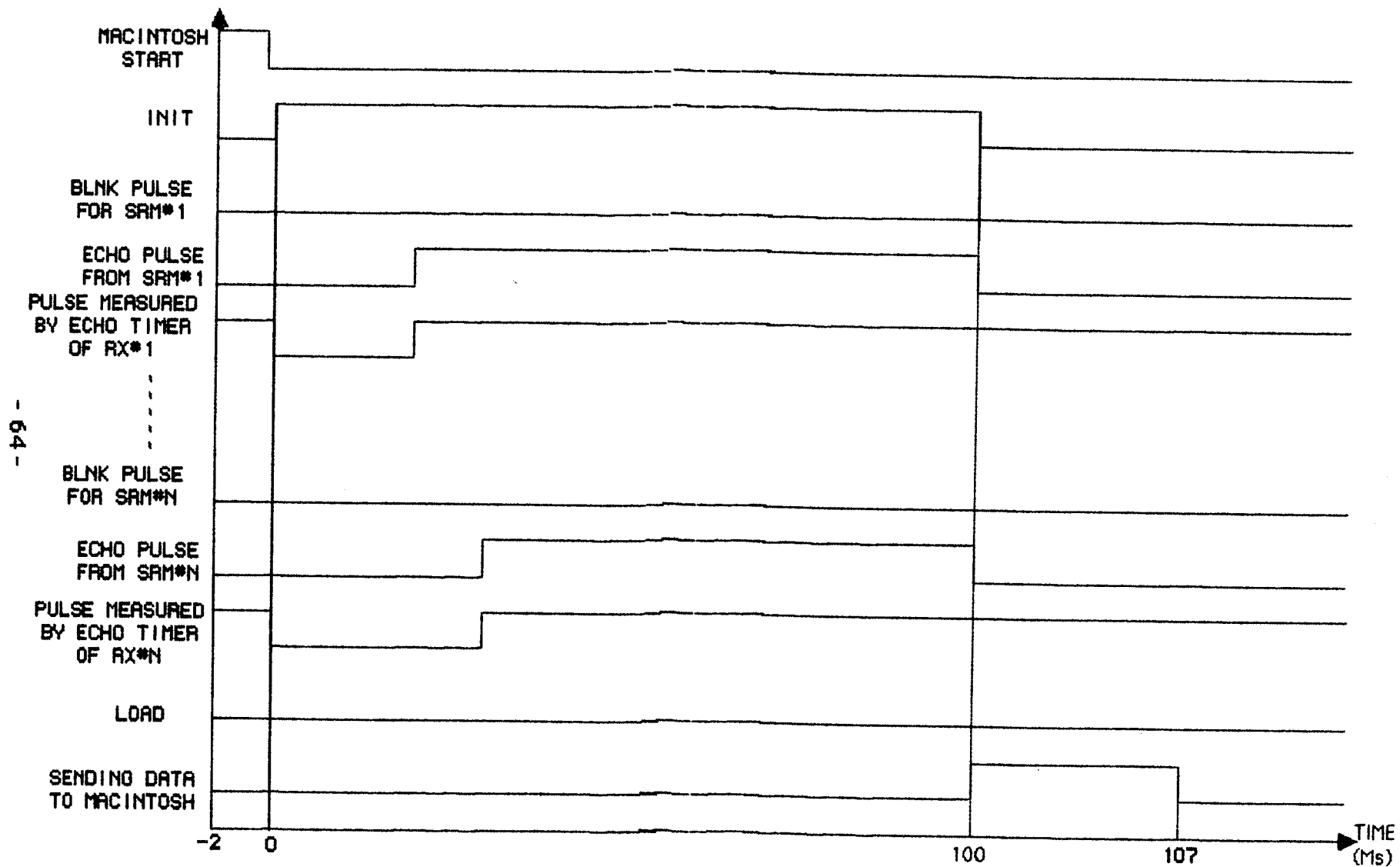


Fig. 5.1. The timing diagram for search mode.

buffer less a constant. After these steps are executed, the program resumes to the communication module which transmits (in about 7 msec) the values of the measured data to the main computer.

(iv) The tracking module tracks the target movement and (as shown in Fig. 5.2) is similar to the search module except (a) the INIT pulse is shorter (10 msec), (b) the BLNK timers which were loaded (as described in the previous paragraph) are activated simultaneously to the INIT pulse and this is done in order to prevent reverberation (and this method can be implemented on transducers with wider beam pattern than these which are used by the tracking system), (c) the number of cycles is usually larger than one (usually three, ten, or 100) according to the command which is received from the main computer. In addition the tracking cycle operates as follows:

At the end of each INIT pulse the program checks if all the tracking cycles were executed, and if not the program starts a LOAD pulse for a duration of 2 msec. This time is used for saving the data from the ECHO timers in a buffer, calculating and writing the related data in the BLNK timers. After the saving and writing are executed, the program resumes to the communication module which transmits the data to the main computer.

At the end of the delay of 2 msec (which was generated by the LOAD timer) the LOAD timer triggers the INIT timer for additional tracking cycle. By this method the tracking cycle has a constant period of 12 msec which is not affected by the software execution time, or the length of the data transmission. After the data is transmitted, the program resumes to the tracking module for an additional tracking cycle. This process lasts till the

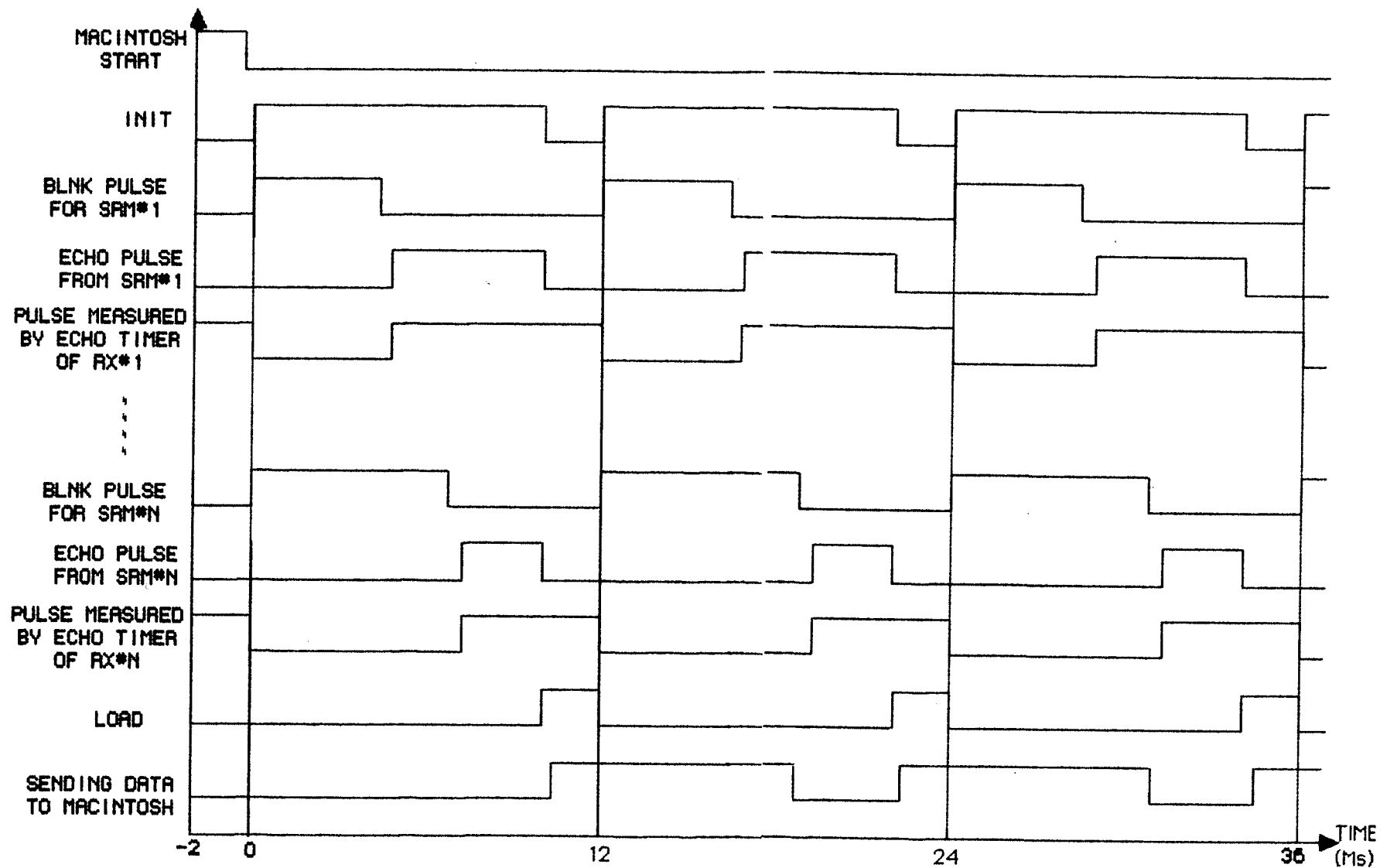


Fig. 5.2. Timing diagram for tracking mode.

program finishes the number of the tracking cycles which were predetermined by the main computer.

5.1.2 Structure of the Software

The software of the control processor is controlled by a state machine. As shown in Fig. 5.3, the states of the machine are defined as modes. When the power is applied to the control processor, the program initializes all the PTMs and the ACIA, the state counter is set to zero, and the control processor is at MODE 0.

The program remains at MODE 0, until four characters are received by the ACIA circuits. The data which was received is checked, and according to the message which was received from the main computer the state machine moves to the next mode.

If the message which was sent by the main computer is not one of these which are defined in TABLE 5.1, then at the end of MODE 0 the program moves to MODE 1. At this mode the ACIA sends to the main computer the ASCII string of "ERROR"+CR. After the six characters are transmitted, the program returns to MODE 0 and waits for further instructions from the main computer.

If the string of "S0"+LF+CR is received from the main computer (at MODE 0), the software moves to MODE 10 and starts the search mode. In this mode the program triggers the INIT timer for duration of 100 msec and stays in this state until pulse of the the INIT timer is completed. When the INIT pulse is terminated, the PTM which contains the INIT timer sends a interrupt to the 6802 CPU and the program moves to MODE 11.

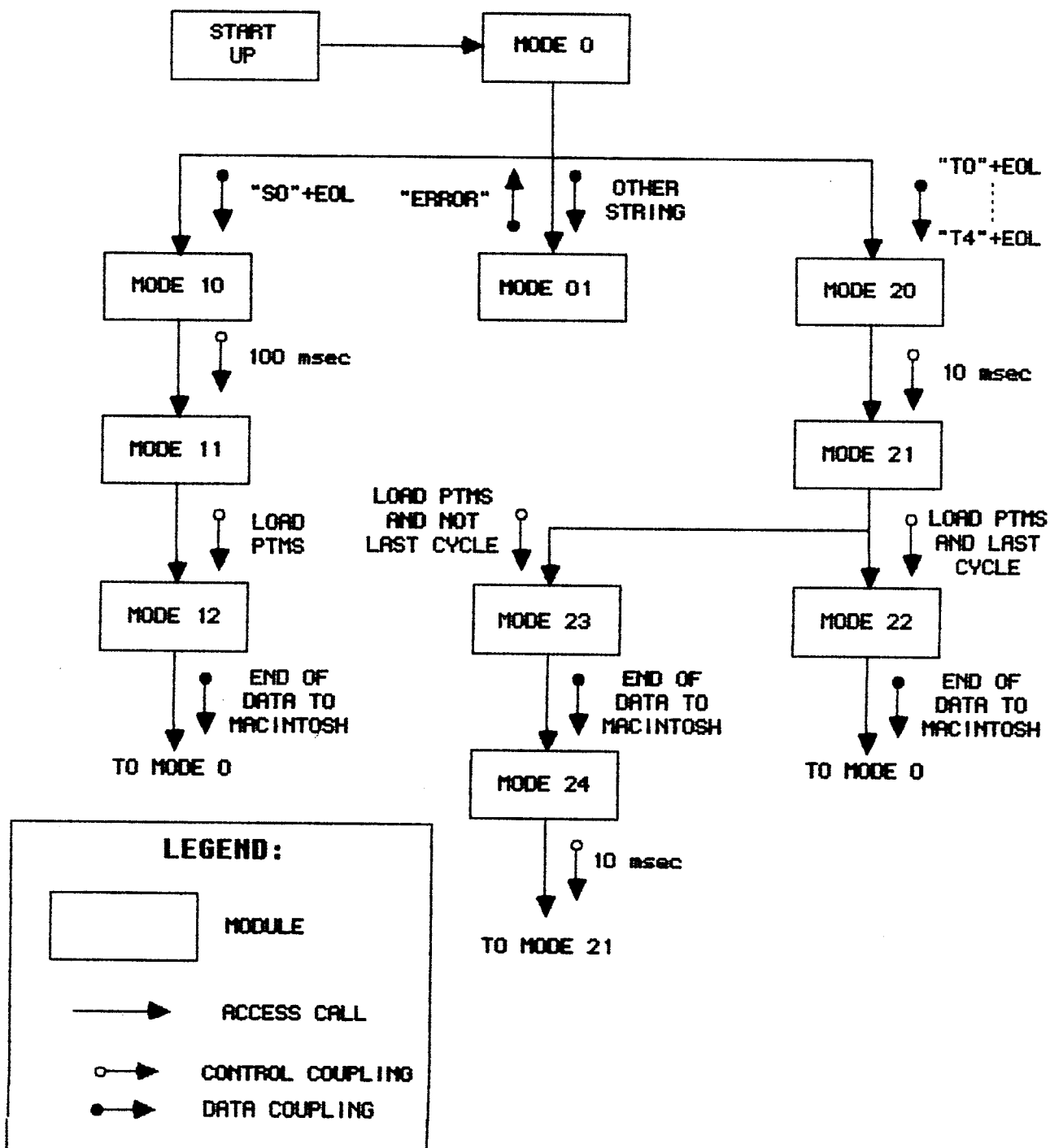


Fig. 5.3. The structure of the control processor software.

At the beginning of MODE 11, the ACIA sends the character "T" to the main computer. Since the ACIA transmits at the rate of 19200 baud, the transmit time of a character is about 0.5 msec. This time period is utilized to transfer the data from the ECHO timers to a buffer and to load the BLNK timers. After the loading, the program moves to MODE 12.

In MODE 12. the ACIA transmits 14 characters to the main computer. The structure and definition of the character string is given in Sec 5.1.3. At the end of this state, the machine resumes MODE 0 to get further instructions from the main computer.

When one of the five strings of "T0"+LF+CR to "T4"+LF+CR is received from the main computer (at MODE 0), the program moves to MODE 20 and starts the tracking mode. In this mode the software (i) triggers the INIT timer for a duration of 10 msec, (ii) loads the cycles counter with the number of the tracking cycles according to the second character of the received string (refer to TABLE 5.1), (iii) sets the control of the LOAD timer to produce automatically a pulse with the width of 2 msec at the end of the INIT pulse. This load pulse is used later on to trigger the INIT pulse for an additional tracking cycle without the software interference. When the INIT pulse is terminated, the INIT timer sends an interrupt to the 6802 CPU, and the program moves to MODE 21.

At MODE 21 the software verifies that the INIT pulse is terminated, then transfers the data from the ECHO timers to a buffer and loads the BLNK timers (similar to MODE 11). At the end of this mode, the software checks whether the cycle counter is empty (which means all the tracking cycles were executed). If the answer is yes then the program moves to MODE 22, else it moves to

MODE 23.

In MODE 22 the software resets the control registers of INIT and LOAD timers. This is done in order to prevent oscillations of these timers. Similarly to MODE 12, the software transmits fourteen characters of data to the main computer. At the end of this mode, the program resumes to MODE 0.

In MODE 23 the software (i) sets the INIT timer to produce an additional 10 msec pulse at the end of the LOAD pulse, (ii) decrements the cycle counter, (iii) the ACIA sends the 14 characters of data to the main computer. At the end of the data transmission, the state machine moves to MODE 24.

In MODE 24 the software verifies that the LOAD PTM has terminated the 2 msec period. If pass it moves to MODE 21 which was discussed earlier. MODE 21 will start only at the end of the 10 msec INIT pulse.

Additional details and the full program listing of the control processor software are given in Appendix B.

5.1.3 Communication Protocols

The string which the main computer sends to the control processor, consists of a four ASCII characters. The first one defines the mode of operation (search or tracking). The second defines the number of tracking cycles. The last two characters (which are LF and CR) are used to cop with the operation of different high level languages which send at the end of transmitted strings, a combination of these two characters. The strings which the main computer can send to control processor are

described in TABLE 5.I.

TABLE 5.I
THE VALID STRINGS FOR THE CONTROL PROCESSOR

The String	Mode	Cycle No.
"S0"+LF+CR	Search	1
"T0"+LF+CR	Tracking	1
"T1"+LF+CR	Tracking	10
"T2"+LF+CR	Tracking	84
"T3"+LF+CR	Tracking	1000
"T4"+LF+CR	Tracking	3

The control processor returns the data which was measured by the ECHO timers in a string of 14 characters. The first character in the string is "T", which can be used for synchronizing the following data. The next six pairs of characters represent data from the six ECHO timers, where each pair of characters relates to the correspondent timer. The value of the data for each transducer, Val, is calculated by

$$\text{Val} = \text{Val1} + 256 \text{ Val2} \quad (5.1)$$

where Val1 is the ASCII value of the first character in the pair, and Val2 is the value of the second character in the pair. The last character in the string is CR, and it notifies the main computer for end of data.

5.2 Main Computer Software

As earlier pointed out, the software of the main computer is written in a high level language. The language which is chosen for this task is BASIC [Micr85] and is selected because (i) the friendliness and the interactive process of the interpreter which allows immediate debugging of the program without the compiling phase (which is common for other languages), (ii) strong support of human-machine interfacing, (iii) graphics support. In addition, at the end of the program debugging phase in order to speed up the execution time of the software the program is compiled by a BASIC compiler [Micr86].

The program is divided to several modules which are described in the following sections. In addition the full program list of the main computer software is given in APPENDIX C.

5.2.1 Main Menu Description

In order to control the operation of the tracking system, the operator has to use menus. When the program is initiated the operator gets a screen where he can select an operation. In order to select an operation, the operator has to point with the mouse on a square (which is drawn on the computer screen close to the operation name), and then to click with the mouse button. As described by Lambert [Lamb85], this menu operation is strongly supported by the special BASIC commands for mouse operation which are (i) BUTTON, (ii) DIALOG.

The menu is described by TABLE 5.II, and the structure of the software is shown by Fig. 5.4.

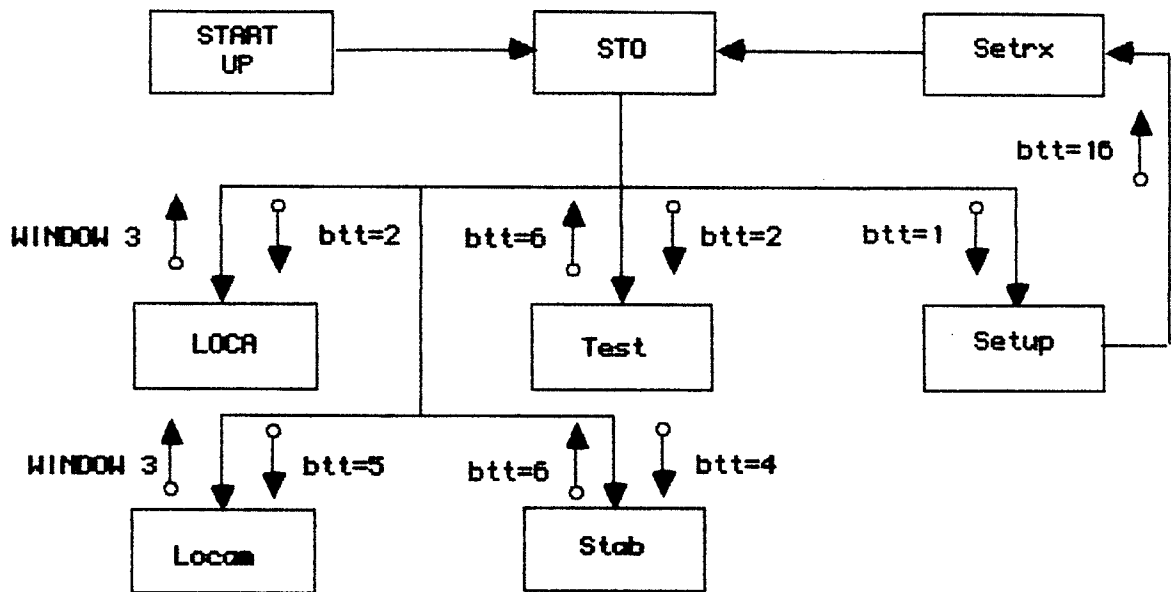


Fig. 5.4. The structure of the main menu.

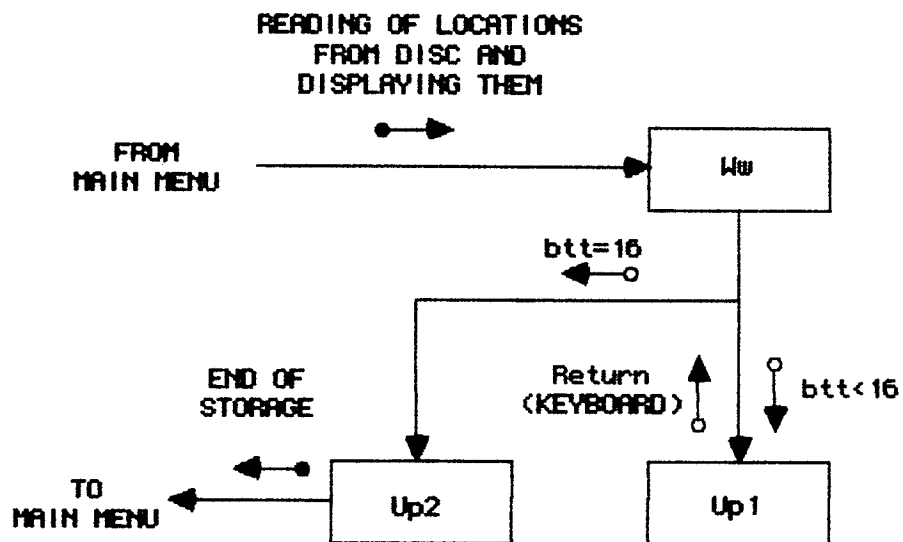


Fig. 5.5. The structure of the receivers set up subroutine.

TABLE 5. II
THE MAIN MENU

Button	Operation
Receivers Set Up	Enables manual entry of the receivers position.
Distance Test	Enables manual calibration of the distance measurements.
Auto Track	Displays the location of the target, the array which is used for locating is selected automatically.
Manual Track	Displays the location of the target, the array which is used for locating is selected manually.
Stability	Measures the stability of the distance measurements.

5.2.2 Receivers Set Up Module

When the Receiver set up button is pressed the program moves to Setup module, whose structure shown in Fig. 5.5. In this module, the location of the five receivers is read from the disk and is displayed according to their three coordinates values. In addition, close to each of the displayed values the program places a button which can be selected by the mouse (Ww module). When one of these buttons is selected by clicking on the mouse, the value of the selected coordinate for the correspondent receiver position has to be reentered by the operator (Up1 module). Each reentry of the data has to be terminated by the Return key. In order to exit from this module, the operator has to press the STOP button (btt=16) which is located at the bottom of the screen. When the STOP button is pressed, the program stores the new locations of

the five transducers as a file on the disk (Up2 module) and resumes to the main module.

5.2.3 Distance Test Module

When the Distance Test button is pressed, the program moves to Test module. In this module, whose structure described in Fig. 5.6 the measured distances of all the receivers from the transmitter are displayed. In order to start the testing, the operator has to select a test from an additional menu which is described in TABLE 5.III.

TABLE 5.III
THE TEST MENU

Button	Operation
Error Mode	Verifies the communication between the main computer and the control processor, by the string of "XYZ"+CR.
Search Mode	The control processor measures the distance in search mode.
Track 1 Mode	The control processor measures the distance in single cycle of track mode.
Track 3 Mode	The distance is calculated by averaging the data, which is received from 3 consecutive tracking cycles.
Track 10 Mode	The distance is calculated by averaging the data, which is received from 10 consecutive tracking cycles.
EXIT	Resumes to Main menu.

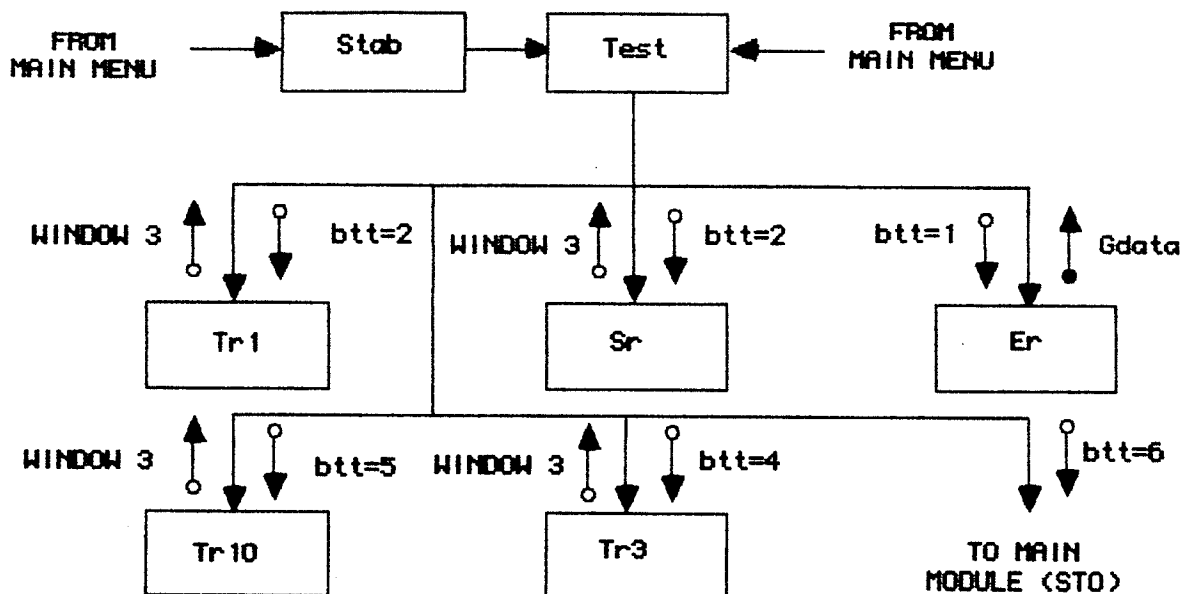


Fig. 5.6. The structure of the test and stability modules.

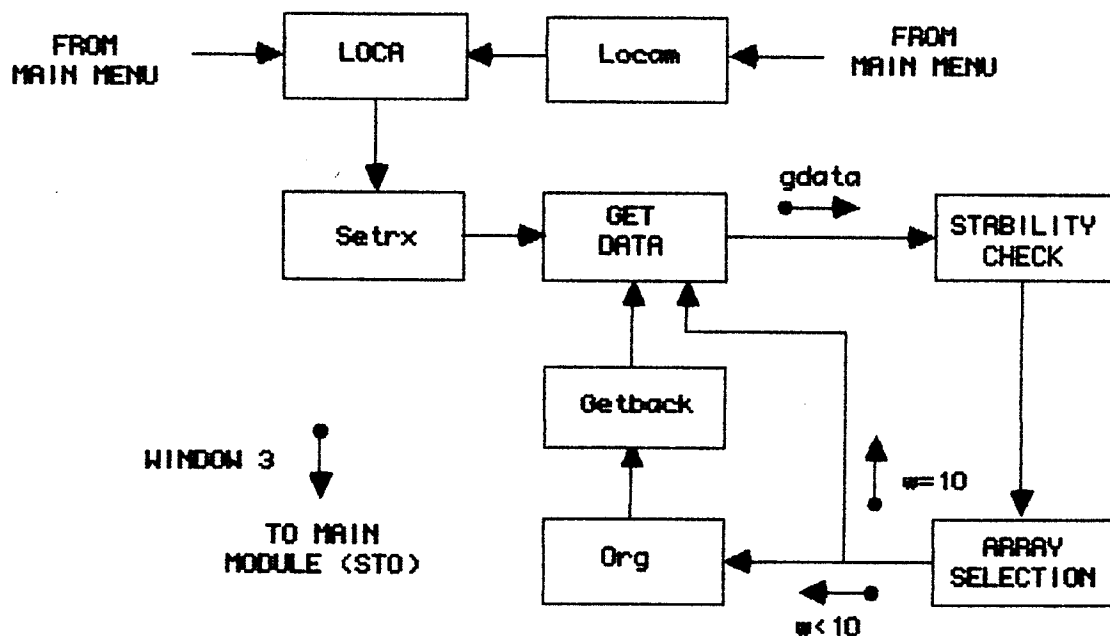


Fig. 5.7. The structure of the locating modules.

When the Error Mode is selected, the main computer sends a string of "XYZ"+CR to the control processor (Er module). Since this string is not included in TABLE 5.I, the control processor sends back the string of "ERROR"+CR (Gdata module) and the "ERROR" message is displayed on the screen of the main computer. This test is executed in order to verify that the communication between the main computer and the control processor is working properly.

If for any reason, the control processor does not respond to the string which was sent by the main computer or send back less than six characters, the program remains in Gdata module for 10 sec and then resumes to Test module.

When the Search or one of the Tracking modes is selected, the main computer sends the correspondent string (which is defined according to the selected mode by TABLE 5.I) to the control processor (Sr, Tr1, Tr2, and Tr10 modules). The measured data is received from the control processor by Gdata module. As earlier pointed out, if the number of characters which are received from the control processor is less than the expected (14 or the number of the tracking cycles multiplied by 14), then the program resumes to test module.

Each pair of data characters (from the received string) is converted to a numeric value by Eq. 5.1, and the distance is evaluated by Eq. 6.1. If for any reason the calculated value of the distance, d_i , is smaller than 0.7 m or greater than 2 m then the measured data from the related receiver is ignored. The valid measured distance for each transducer in Search or Track 1 modes, or the average of the valid measured distances in Track 3 or Track 10 modes is calculated and displayed on the screen of the main computer. The distances are displayed with a resolution of

1 mm, and when the distance measurement of a certain receiver does not obtain a valid value or valid average, then the value of x.xxx m is displayed as the distance for the correspondent receiver.

The program returns to Test module and stops to measure distances when the mouse is pointing on WINDOW 3 and is selected by the operator. In order to resume to Main menu, the operator has to select the EXIT button. The Distance Test is used in order to calibrate the distance measurements of the tracking system (which are described in Sec. 6.2.1).

5.2.4 Stability Test Module

As shown by Fig. 5.6, the Stability module has almost the same structure as the Distance Test module. When the operator starts this module, he has to select the desired mode for operation (Search, or one of the Tracking modes). When the mode is selected, the distances are measured and processed in the same way as the previous module, but instead of displaying them on the screen of the computer they are stored until the next measurement cycle is executed. When the next measurement is executed, the differences between the distances values (for each transducer) of the current cycle and the previous one are calculated and displayed on the screen of the computer. In addition, the maximal differences in the distance measurement values are saved in the main computer memory and are displayed in addition to the display of the temporary differences.

The termination of the Stability test is achieved in the same way as the Distance test. The Stability test is used for measurement of the tracking volume, as described in Sec 6.2.3.

5.2.5 Locating Modules

When Auto or Manual Track button is pressed, the program moves to the locating modules which their structure is shown in Fig. 5.7. In these modules the program measures the distances of the target from the array, and by implementing the locating algorithm which is described in Sec 5.3 the position of the target (by the values of the coordinates) is displayed on the main computer monitor.

The array which is used for tracking is selected manually from TABLE 5.IV in the Manual Track mode (Locam module), or by the first set of the three transducers which the jitter error of its all three transducers is less than 0.0008 m. The sets of the transducers are defined by TABLE 5.IV and checked by STABILITY CHECK and ARRAY SELECTION modules.

Before the tracking system starts to locate the values of the locations of the receivers are read from the disk, and in addition the values of the constants (which are needed for coordinate transformation by the locating algorithm and are described in Sec. 5.3.2) are calculated in Setrx module.

The software measures the distances of the target from the array in the same way as in the Distance Test module (GET DATA module). If the measured distances for the selected array in the Manual locating mode, or for all the sets of arrays in the Auto Track mode are not stable, then the software continues to measure the distances without calculating the location of the target (as shown in Fig. 5.7 for $w=10$).

When the distance measurements of the selected set of transducers is in manual mode, or one of the sets of the arrays in auto modes is stable, then the software calculates the location of the target by Eqs. 5.4 to 5.6 (Org module). In addition, (as described in Sec. 5.3.2) in order to display the position of the target in coordinate system where the axes are parallel to the reference frame, the calculated values of the coordinate are transformed (by the Getback module).

The value of this software is, that the calculated values of the position of the target in the coordinates system of the reference frame are the momentary trajectory of the target. By continuous measurement of the distances of the target from the array, and then implementing the locating software the trajectory of the limb movement is obtained.

5.3 Locating Algorithm

As earlier pointed out, a target can be located when the distances of the target from three receiving transducers are known. The target is located at the intersection point of three spheres that their centre are at the location of the receiving transducers, and their radiuses are the measured distances between the target and the receivers. The coordinate values of the target location, x_s, y_s, z_s , can be calculated by the equations of the spheres which are generated by each of the transducers

$$(x_i - x_s)^2 + (y_i - y_s)^2 + (z_i - z_s)^2 = d_i^2, \quad i=1,3 \quad (5.2)$$

where, d_i , is the distance of each of the transducers from the target, and (as shown in Fig. 5.8) x_i, y_i, z_i are the coordinate values of transducer i which is used for locating.

The solution of Eq. (5.2) requires complex mathematical methods [ScRo87]. The problem is how to derive from Eq. (5.2) a set of three linear equations with the variables x_s, y_s, z_s without generating a singular matrix (where the problem is not solvable). In the following sections a method for solving this problem is described and implemented.

5.3.1 Removing the Square Values

The square values of x_s, y_s, z_s in Eq. (5.2) can be removed by transferring and rotating the coordinates. In order to achieve this solution, the transformation has to fulfill three conditions which are:

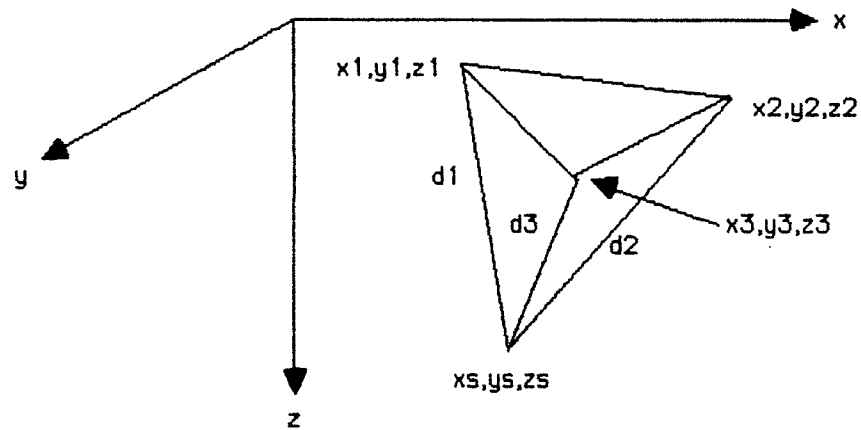


Fig. 5.8. General problem of locating a target with three receivers.

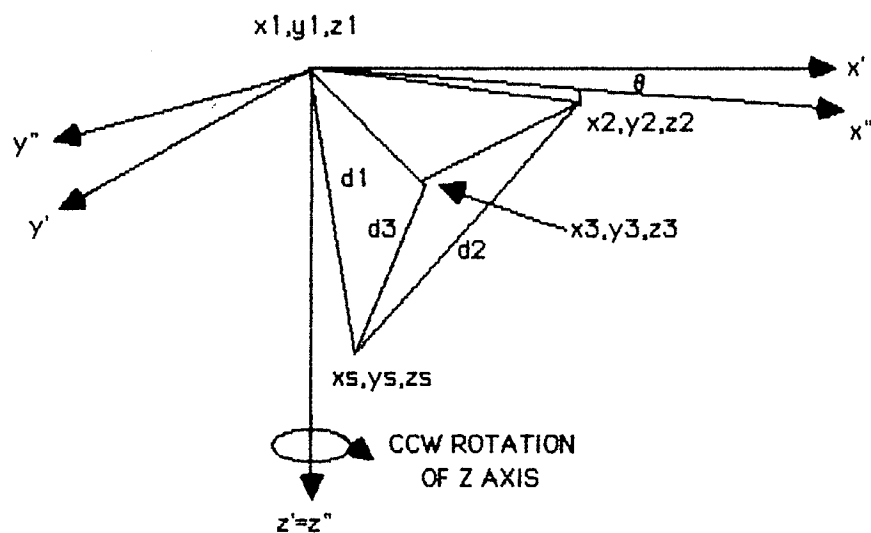


Fig. 5.9. Step*2 rotation of x, y plane around the z axis.

(i) The origin has to be shifted to the location of the first transducer, thus the location of Rx#1 in the transformed coordinate system is 0,0,0.

(ii) The second transducer has to be located on the X axis, thus the location of Rx#2 in the transformed coordinate system is $x_2, 0, 0$.

(iii) The third transducer has to be located on the plane which is generated by $z=0$, thus the location of Rx#3 in the transformed coordinate system is: $x_3, y_3, 0$.

When the transformed values of the locations of transducers are substituted with x_i, y_i, z_i in Eq. (5.2), then

$$\begin{aligned}x_s^2 + y_s^2 + z_s^2 &= d_1^2 \\x_s^2 + y_s^2 + z_s^2 + x_2^2 - 2x_2x_s &= d_2^2 \\x_s^2 + y_s^2 + z_s^2 + x_3^2 - 2x_3x_s + y_3^2 - 2y_3y_s &= d_3^2\end{aligned}\tag{5.3}$$

x_s can be calculated by subtracting the first equation from the second in the set of equations which is generated by Eq. (5.3), and equals to

$$x_s = \frac{d_1^2 + x_2^2 - d_2^2}{2x_2}\tag{5.4}$$

Similarly, y_s can be calculated by subtracting the first equation from the third in the set of equations which is generated by Eq. (5.3), and by substituting the value of x_s with its value that is calculated by Eq. (5.4)

$$y_s = \frac{d_1^2 + x_3^2 + y_3^2 - d_3^2 - 2x_s x_3}{2y_3} \quad (5.5)$$

Finally, z_s is calculated by substituting the values of x_s and y_s in the first equation of Eq. (5.3) with the calculated values that are obtained by Eqs. (5.4), (5.5)

$$z_s = \sqrt{d_1^2 - x_s^2 - y_s^2} \quad (5.6)$$

According to Eq. (5.6), z_s has two solutions. The negative value of z_s is rejected by the assumption that the target is located only in front of the array and not behind it.

Because of conditions (i) and (ii) of the transformation, two of the transducers are located on the transformed X axis. Therefore if all the three receivers are located on the same axis, the third transducer is located on the X axis too, and $y_3=0$. Therefore, Eq. (5.5) has a singular solution. For this reason, the selection of such an array for locating a target is prohibited.

When the location of a target is calculated, the transformed coordinates values of the target location are obtained by Eqs. 5.5 to 5.7. These values are transformed back to the original coordinate system (which is related to the reference frame) in four steps, which are reversed in direction and in order to these which are described in the next section.

5.3.2 Implementation of the Coordinates Transformation

The coordinate transformation, which is described in Sec. 5.3.1 is obtained by four steps:

(i) The origin is moved to the location of Rx#1, which is located at x_1, y_1, z_1 . This is done by

$$x_i' = x_i - x_1, \quad y_i' = y_i - y_1, \quad z_i' = z_i - z_1 \quad i=1,3. \quad (5.7)$$

where, x_i, y_i, z_i are the locations values of the transducers in the old coordinate system, and x_i', y_i', z_i' are the locations values of the transducers in the new coordinate system. Therefore, $x_1' = y_1' = z_1' = 0$ and the first condition of Sec. 5.3.1 are fulfilled. The reverse transformation (which as earlier pointed out, is used for returning to original coordinate system) can be achieved by adding the values of x_1, y_1, z_1 to the calculated values of the target location.

(ii) As shown in Fig. 5.9, the coordinates are rotated around Z axis in ccw direction until $y_2'' = 0$. The definition of cw or ccw direction of the rotation is related to an observer, who is located on a positive value of the spun axis and is looking toward the origin. The rotation angle, θ , is calculated by $\theta = \arctan(y_2/x_2)$. The rotation matrix is calculated by [Anto81]

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (5.8)$$

where x', y', z' are the values in the old coordinate system, and x'', y'', z'' are the values in the rotated coordinate system. After this rotation the second transducer is located at $x_2'', 0, z_2''$ in the

new coordinate system.

The inverse matrix which is used for returning to the previous coordinate system is calculated by [Anto81]

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \quad (5.9)$$

The inverse matrix in Eq. (5.9) is the same as the matrix in Eq. (5.8), when θ is substituted with $-\theta$.

(iii) As shown in Fig. 5.10, the coordinates are rotated around Y axis in cw direction until $z_2''' = 0$. The rotation angle, θ_1 , is calculated by $\theta_1 = \arctan(y_2''/x_2'')$ and the transfer matrix is

$$\begin{bmatrix} x''' \\ y''' \\ z''' \end{bmatrix} = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 \\ 0 & 1 & 0 \\ -\sin \theta_1 & 0 & \cos \theta_1 \end{bmatrix} \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \quad (5.10)$$

where x'', y'', z'' are the values in the old coordinate system, and x''', y''', z''' are the values in the rotated coordinate system. After this rotation, the second transducer is located at $x_2''', 0, 0$ in the new coordinate system and the second condition of Sec. 5.3.1 is fulfilled.

It can be easily proved that the inverse matrix of Eq. (5.10) is the same matrix when θ_1 is substituted with $-\theta_1$.

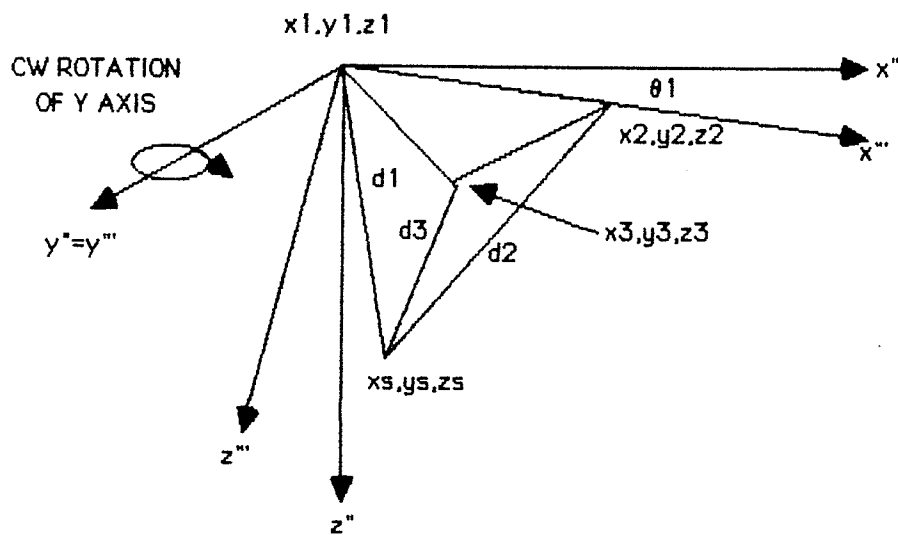


Fig. 5.10. Step*3 Rotation of x,z plane about the y axis.

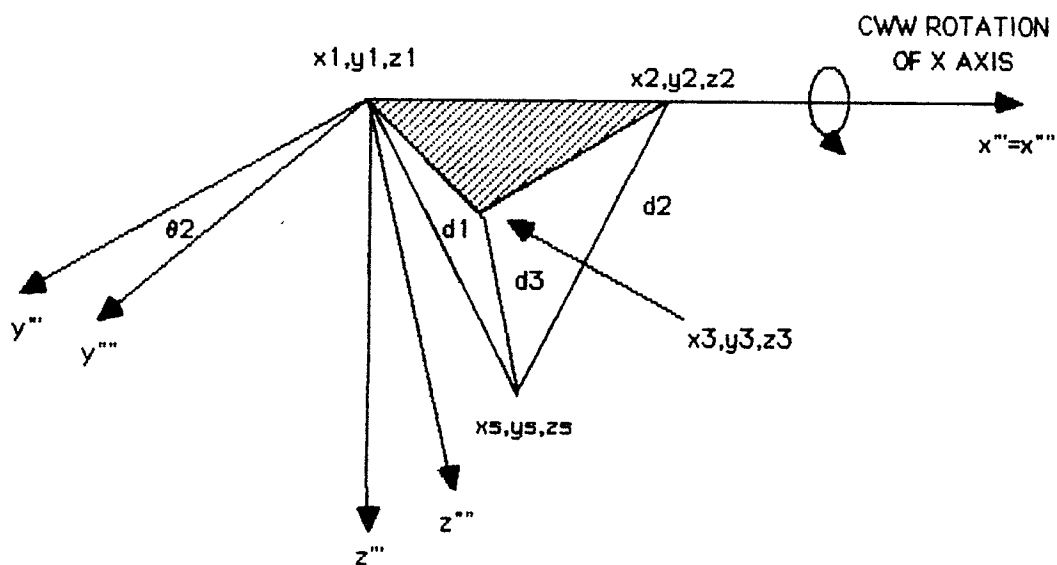


Fig. 5.11. Step*4 Rotation of y,z plane around the z axis.

(iv) As shown in Fig. 5.11, the coordinates are rotated around X axis in cww direction until $z_3''''=0$. The rotation angle, θ_2 , is calculated by $\theta_2=\arctan(z_3''/y_3'')$ and the transfer matrix is

$$\begin{bmatrix} x'''' \\ y'''' \\ z'''' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & \cos \theta_2 & \sin \theta_2 \\ 0 & -\sin \theta_2 & \cos \theta_2 \end{bmatrix} \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \quad (5.11)$$

where x'', y'', z'' are the values in the old coordinate system, and x''', y''', z''' are the values in the rotated coordinate system. After this rotation the third transducer is located at $x_3''', y_3''', 0$ (in the new coordinate system) and all the three conditions of Sec. 5.3.1 are fulfilled. Therefore Eqs. (5.4)-(5.6) can be implemented, and the target can be located.

It can be easily proved that the inverse matrix of Eq. (5.11) is the same matrix when θ_2 is substituted with $-\theta_2$.

5.3.3 Software Implementation

As shown in Fig. 5.12, the array of receivers consists of five receivers. The receivers are located at the distance of about 17 cm from the x,y plane, which is defined as the upper side of the reference frame. This is because the transducers have to be attached to the reference frame by a swivel (that has the size of about 17 cm), and for this reason they are not located at the x,y plane. The main transducer (Rx1) is located at $x=y=0$. Two other transducers are located parallel to the X axis at $x \cong +25$ cm (Rx2) and $x \cong -25$ cm (Rx3), and the last two are located symmetrically parallel to the Y axis at $y \cong +25$ cm (Rx4) and $y \cong -25$ cm (Rx5).

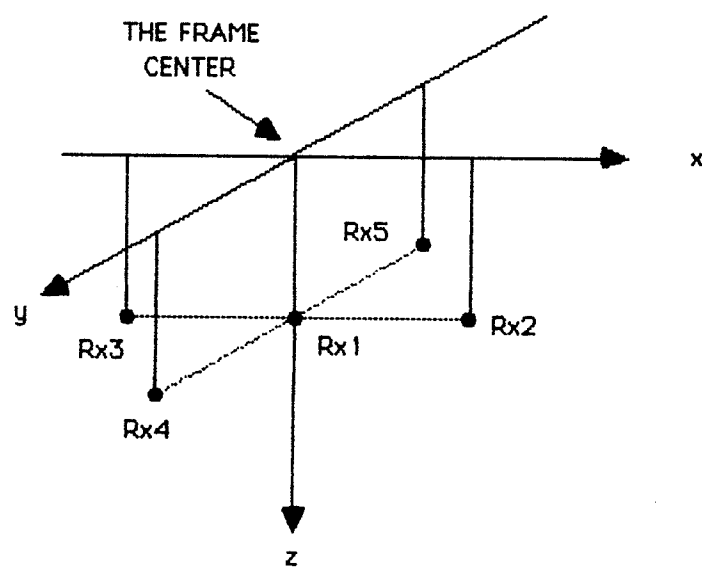


Fig. 5.12. The roughly location of the five transducers.

As described in Sec. 5.2.2, the exact location of the transducers is entered in the computer and is stored as a file on a floppy disk. Since only three transducers are needed for locating the target, the receivers are divided arbitrary to ten groups of three transducers per group. This partition contain all the combinations of selecting three out of five receivers. The partition of the transducers to groups is defined by TABLE 5.IV.

TABLE 5.IV
PARTITION OF TRANSDUCERS TO GROUPS

SET#	Rx#1	RX#2	Rx#3
0	1	2	3
1	1	2	4
2	1	2	5
3	1	3	4
4	1	3	5
5	1	4	5
6	2	3	4
7	2	3	5
8	2	4	5
9	3	4	5

The selection of the SET#0 and SET#5 is not allowed for locating, because in those groupS all the three transducers are located on the same axis. X axis for SET#0 and Y axis for SET#5.

The coordinate values of Rx#1 (for each group) are stored in the memory to obtain the reverse transformation, which is described in step (i) in the previous section. The coordinate values of the transducers for each set, are transformed by Eqs. (5.7),(5.8),(5.10) and (5.11) and saved as a table in the computer memory. In addition, to speed up computations the values

of sine and cosine of $-\theta, -\theta_1, -\theta_2$ for each set of transducers are stored in the computer.

In order to locate a target all the three transducers of a certain group has to measure a correct distance from the target, and the method of how the most proper group is selected is described in Sec. 5.2.5. The values of x_s, y_s, z_s are calculated by substituting in Eqs. (5.4)-(5.6) the precalculated coordinates of the transducers for the selected set with the values of x_2, x_3, y_3 , and by substituting the values of d_1-d_3 with the measured distances.

The values of x_s, y_s, z_s are transformed back to the original coordinate system by (i) substituting in Eq. (5.11) the value of θ_2 with the value of $-\theta_2$ (for the related group of transducers), (ii) substituting in Eq. (5.10) the value of θ_1 with the value of $-\theta_1$ (for the related set of transducers), (iii) substituting in Eq. (5.8) the value of θ with the value of $-\theta$ (for the related set of transducers), (iv) by adding the original coordinate values of $R_{x\#1}$ with the calculated values x_s, y_s, z_s . Since the values of sine and cosine of $-\theta, -\theta_1, -\theta_2$ for the selected set are already stored in the computer memory, the calculations are speeded up.

CHAPTER VI

TESTING

In this chapter the testing of the tracking system is described. The purpose of those tests is to verify the ability of the system to locate and track a target according to the theory which is described in the previous chapters.

In general, the tests are divided into two major groups. The first group consists of purely electronic tests which check the accuracy of the hardware. The second group of tests deals with calibration of the system and involves all the parts of the system, such as electronic circuits, ultrasound transducers, and software.

6.1 Testing Hardware Test Subsystems

6.1.1 Frequency of the Clock

Since, the distance measurement is based on measuring the time period (in which the burst of ultrasound pulses propagates from the transmitter to one of the receivers of the array), it is critical to measure this time accurately. The measurement is done by timers which count clock pulses. As described in Sec. 4.2.3, the clock which is used by the counters of the timers is the same clock which drives the CPU of the control processor. Any error in the frequency of this clock will affect the timers operation, and the readings of the time period will be erroneous. The frequency of the clock was measured with a universal counter/timer (type DC 503A manufactured by Tektronix) at pin 22 of the 44 Pin Edge Connector of the 6802 Thesis Board 2. The frequency of 999.98 kHz was measured and was stable for 15 minutes. The error of the

measured frequency when it is related to 1 Mhz (which is the specified frequency of the circuit) is .002%.

This error is relatively small when compared with the stability of the speed propagation of the sound waves in air, giving an error of 0.175 % per 1 °C [BESS80]. The errors of those two parameters are compared because they have the main affect of the accuracy of the distance measurement.

6.1.2 Measurement of the Timers

As pointed out in Sec. 5.1.1, a pulse is provided by the signal processor. This pulse which its width equals to the time delay in which the ultrasound burst propagates in the air (from the target ot the receiver) is measured by the ECHO timers. In order to verify that those timers are operating properly, the width of the pulse was measured with the Universal Counter Timer, and compared to the one which was counted by the counters in the ECHO timers. The Universal Counter is able to measure the width of a positive pulse but as shown in Fig. 5.2, the ECHO pulse is a negative pulse. The measurement was executed in Track 10 mode, and the width of the pulse, P_w , which is measured by the Universal Timer equals to

$$P_w = t_{INIT} - t_{ECHO} \quad (6.1)$$

where, t_{INIT} is the value of the INIT pulse and equals to 10 ms in the tracking mode, and t_{ECHO} is the time which is measured by the ECHO timer. The readings were taken for each receiver at three points, where the distances of the target from the receiver were about 80 cm, 120 cm, and 160 cm. The values of the measured P_w that are measured by the Universal Counter, versus the calculated

values of P_w (which are achieved by substituting in Eq. (6.1) the measured value of the ECHO timer with t_{ECHO}) are shown in TABLE 6.I. Because the ECHO pulse is a negative one the longest time in TABLE 6.I denotes the shortest distance.

TABLE 6.I
COMPARISON BETWEEN THE UNIVERSAL COUNTER
AND ECHO TIMER READINGS

Receiver No.	Counter Reading [ms]	Computer Reading [ms]
1	5.2965	5.296
	6.3594	6.360
	7.5602	7.560
2	5.3844	5.384
	6.2549	6.255
	7.5666	7.566
3	5.4725	5.473
	6.3223	6.322
	7.5666	7.566
4	5.8441	5.844
	6.3493	6.350
	7.4994	7.499
5	5.5242	5.524
	6.5303	6.530
	7.5342	7.534

From the results it can be observed, that the computer readings are accurate, and the error is less than $1 \mu s$ which is equivalent to 0.34 mm when the sound waves propagate at the speed of 340 m/s. The error is caused by the clock frequency which is 1 MHz. Thus the counter can not measure time periods shorter than $1 \mu s$.

6.2 System Calibration

6.2.1 Distance Measurement

Since the locating algorithm (which is described in Sec. 5.3) is affected by the accuracy of the distance measurement, it is important to measure these distances accurately. The distance of the target from transducer i , d_i , is calculated by

$$d_i = c \cdot t_i + b_i \quad (6.2)$$

where, c is the speed of the sound waves at the measurement time, t_i is the propagation time which is measured by the ECHO timer, and b_i is the delay of the transducer. The assumption is that during the time period when the transducers are calibrated the temperature of the air is stable, and the speed of the sound waves is constant.

Since the dimensions of the transducer are not negligible, the distance measurement is executed when both of the transducers (receiver and transmitter) are adjusted to have a common acoustic axis. This condition can be achieved in two steps (i) adjusting the transmitter and the receiver to be horizontal to ground (with a surface level) (ii) moving the transmitter on the same horizontal plane until the distance which is measured is the shortest one. In addition, the most northern point of the transducers is marked, and then finally the distance between these marked points is measured by a metric tape.

Each receiver transducer is calibrated at three distances, which are located at the center and the two edges of the estimated path of the limb movement. Since the speed of the sound waves in

the air c and the delay of the transducer b_i are unknown, c and b_i are calculated by using Eq. (6.2) and substituting the values of t_i and d_i with the measured values from two points (where the distances are the smallest). The third point is used to verify the results and to check the linearity of the transducer. The software which is used to measure distances is described at Sec 5.2.3, and the results of the measurement are recorded and displayed at TABLE 6.II. Where d_1, d_2, d_3 are the measured distances by a metric tape, and d_1', d_2', d_3' are the distances which are calculated by substituting in Eq. (6.2) b_i and c with the arbitrary values of $b_i=0$ and $c=341$ m/s, and t_i with the measured time by the ECHO timers.

TABLE 6.II
DISTANCE MEASUREMENTS

Rx. #	d_1 [m]	d_1' [m]	d_2 [m]	d_2' [m]	d_3 [m]	d_3' [m]
1	0.962	0.964	1.211	1.213	1.555	1.557
2	0.938	0.940	1.194	1.197	1.531	1.537
3	0.932	0.934	1.183	1.185	1.526	1.530
4	0.935	0.937	1.185	1.187	1.526	1.527
5	0.937	0.939	1.186	1.189	1.532	1.533

As earlier pointed out, the measured values of the distances d_1, d_2 , and time related distances d_1', d_2' are used to calculate by Eq. (6.2) the values of the coefficients c and b_i . In addition, the distance d_3 is calculated by substituting in Eq. (6.2) c and b_i with the pre calculated values, and t_3 with the calculated value from d_3' . TABLE 6.III shows (i) the calculated values of speed of sound wave c , (ii) the calculated values of the transducer delay b_i , (iii) the calculated value of the distance

d3", (iv) the linearity error, LE, is calculated by

$$LE = d3 - d3'' \quad (6.3)$$

where d3 is the measured distance at point#3.

TABLE 6.III
COEFFICIENT AND LINEARITY CALCULATIONS

Rx. #	c [m/s]	b _i [m]	d3'' [m]	LE [m]
1	341.0	-.0020	1.557	0.000
2	342.3	-.0075	1.535	-.004
3	341.0	-.0020	1.528	-.002
4	343.7	-.0113	1.531	-.005
5	339.0	0.0040	1.528	0.004

As shown by TABLE 6.III, the maximum linearity error is 0.5 cm, and the reasons for this error are (i) the dimensions of the transducer are not negligible when compared to the accuracy of the distance measurement, (ii) the linearity of the amplifier in the SRM, (iii) the variations in the speed of the ultrasound waves in the air that occurred because of the long durations of the tests. The effect of the linearity error is discussed in Chap. 7.

6.2.2 Jitter Measurement

The jitter test measures the jitter of the distance measurements of the system from a stationary transmitter. In order to measure the jitter, the transmitter is placed on a horizontal plane which is defined as the measurement platform. Like in the previous section, the transmitter and the receiver are tilted until their surface is parallel to the measurement platform and horizontal to ground. The jitter tests are executed at two distances of the measurement platform from the receivers array. Those distances are the two edges of the path of the limb movement.

The distance of the measurement platform from the tested receiver can be found by moving horizontally the transmitter on the platform until the shortest distance is measured, and at this point the measured distance is the distance of measurement platform from the receiver. In addition, the jitter is measured as a function of the tilt angle θ (of the transmitter from the acoustic axis of the receiver), and can be calculated by

$$\theta = \arccos(D/D_1) \quad (6.2)$$

where, D is the distance of the measurement platform from the receiver, and D_1 is the measured distance of the transmitter from the receiver.

The software which executes the jitter test is described in Sec. 5.2.4, and is calculating the maximal and the jitter of the distance readings. The jitter test is performed in two steps (i) checking the jitter with a single transducer at a single distance

in order to verify which mode of operation is the most suitable for tracking, (ii) checking the jitter of all the transducers at two distances in the pre-selected mode of operation. The first step is performed on Rx#1 at D=1.544 m, which is near the edge of the limb movement. The jitter in the distance reading as function of the tilt angle θ are recorded for all the operating modes and are shown by TABLE 6.IVa and TABLE 6.IVb.

TABLE 6.IVa
THE AVERAGE JITTER IN DISTANCE READINGS
FOR ALL OPERATING MODES

Mode	0° [cm]	3° [cm]	6° [cm]	9° [cm]	12° [cm]	15° [cm]
Search	0.0	0.0	0.0	0.2	0.2	0.4
Track 1	0.0	0.0	0.0	0.2	0.2	0.4
Track 3	0.0	0.0	0.0	0.0	0.1	0.2
Track 10	0.0	0.0	0.9	0.0	0.0	0.2

TABLE 6.IVb
THE MAXIMAL JITTER IN DISTANCE READINGS
FOR ALL OPERATING MODES

Mode	0° [cm]	3° [cm]	6° [cm]	9° [cm]	12° [cm]	15° [cm]
Search	0.0	0.1	0.1	0.5	0.6	1.2
Track 1	0.0	0.0	0.1	0.5	0.8	0.8
Track 3	0.0	0.0	0.0	0.3	0.4	0.7
Track 10	0.0	0.0	0.0	0.2	0.2	0.4

As expected, the smallest jitter is achieved at Track 10 Mode because of the longest integration time. On the other hand,

in Track 3 Mode the average error is greater than 0.1 cm only for $\theta=15^\circ$, and the sampling rate is three times faster than in Track 10 Mode. For those reasons the Track 3 Mode is the mode which is used for tracking. The jitter of all the transducers was tested, and the measurement time at each point was 20 s. which is relatively long time when compared to the tracking cycle which is 36 ms for Track 3 mode. The average jitter in the distance readings for all the receivers are shown in TABLE 6.V, and the maximal jitter in TABLE 6.VI.

The average error for all the transducers for tilt angles θ smaller than 15° is less than 0.1 cm, except Rx#4 which is slightly out of range.

TABLE 6.Va
THE AVERAGE JITTER IN DISTANCE READINGS
FOR SHORT DISTANCES

Rx. #	D [m]	0° [cm]	3° [cm]	6° [cm]	9° [cm]	12° [cm]	15° [cm]	θ_{\max} [deg]
1	0.981	0.00	0.00	0.00	0.10	0.10	0.20	0.40
2	0.956	0.00	0.00	0.00	0.10	0.10	0.10	1.20
3	0.956	0.00	0.00	0.05	0.05	0.05	0.10	1.20
4	0.953	0.00	0.00	0.00	0.00	0.10	0.20	0.90
5	0.956	0.00	0.00	0.05	0.05	0.10	0.10	0.60

TABLE 6. Vb
THE AVERAGE JITTER IN DISTANCE READINGS
FOR LONG DISTANCES

Rx. #	D [m]	0° [cm]	3° [cm]	6° [cm]	9° [cm]	12° [cm]	15° [cm]	θ_{max} [deg]
1	1.545	0.00	0.00	0.00	0.10	0.10	0.20	0.50
2	1.524	0.00	0.00	0.00	0.10	0.10	0.20	0.70
3	1.518	0.00	0.00	0.00	0.00	0.05	0.20	0.90
4	1.517	0.00	0.00	0.10	0.10	0.20	0.20	0.60
5	1.520	0.00	0.00	0.00	0.00	0.00	0.20	1.20

TABLE 6. VIa
THE MAXIMAL JITTER IN DISTANCE READINGS
FOR SHORT DISTANCES

Rx. #	D [m]	0° [cm]	3° [cm]	6° [cm]	9° [cm]	12° [cm]	15° [cm]	θ_{max} [deg]
1	0.981	0.00	0.00	0.00	0.30	0.50	0.50	1.60
2	0.956	0.00	0.00	0.00	0.30	0.30	0.40	3.80
3	0.956	0.00	0.00	0.30	0.30	0.30	0.40	3.80
4	0.953	0.00	0.00	0.00	0.10	0.30	0.40	2.40
5	0.956	0.00	0.10	0.30	0.30	0.30	0.40	1.60

TABLE 6. VIb
THE MAXIMAL JITTER IN DISTANCE READINGS
FOR LONG DISTANCES

Rx. #	D [m]	0° [cm]	3° [cm]	6° [cm]	9° [cm]	12° [cm]	15° [cm]	θ_{max} [deg]
1	1.545	0.00	0.00	0.20	0.30	0.70	0.70	1.10
2	1.524	0.20	0.30	0.30	0.30	0.30	0.60	2.50
3	1.518	0.00	0.10	0.20	0.20	0.20	0.70	2.80
4	1.517	0.20	0.30	0.30	0.30	0.50	0.90	1.70
5	1.520	0.00	0.00	0.05	0.10	0.10	0.40	3.30

θ_{\max} is defined as the tilt angle where the system stops to measure the distances. Since the signal to noise ratio at this point is high, it is hard to find accurately the value of θ_{\max} , but θ_{\max} is greater than 15° (the angle which is used to calculate the tracking volume). The values of θ_{\max} for each transducer at the two extreme distances are shown in TABLE VII. Where $\theta_{S\max}$ denotes the value of θ_{\max} for a short distance, and $\theta_{L\max}$ denotes the value of θ_{\max} for a long distance. In addition the jitter in the distance readings for θ_{\max} are shown in TABLES 6.V and 6.VI.

TABLE 6.VII
THE MAXIMAL TILT ANGLE

Rx. #	$\theta_{S\max}$ [deg]	$\theta_{L\max}$ [deg]
1	19.40	17.50
2	21.85	19.20
3	19.50	20.54
4	20.88	20.20
5	17.00	20.10

6.2.3 Tracking Volume Measurement

In this section the size the tracking volume which is calculated in Sec. 3.2 is verified. The tracking system uses the array which is shown in Fig. 5.12. In addition (as described in Sec. 3.2.2) each of the side transducers is tilted from the horizontal position toward Z axis in 10.7° , so that the acoustic axis of the transducer is pointing toward the estimated center of the limb movement. The tilt of the transducers is measured by a level angle finder, and can be adjusted by the swivel which attaches him to the frame. The locations of the transducers (that are shown in TABLE 6.VIII) are measured by a tape, and are entered to the computer as described in Sec. 5.2.2.

TABLE 6.VIII
THE LOCATION OF THE TRANSDUCERS

	X [m]	Y [m]	Z [m]
Rx. #1	0.000	-.001	0.125
Rx. #2	0.265	0.000	0.150
Rx. #3	-.265	0.000	0.155
Rx. #4	0.000	0.260	0.153
Rx. #5	0.000	-.268	0.155

The transmitter is located on the measurement platform which is described in the previous section. During the measurements, the platform is placed at the two extreme levels (that are close as possible to the edges of the path of the limb movement) and are at the distances $z \approx 1.80$ m and $z \approx 1.10$ m from the array. Since the transducers of the array are located at $12.5 \text{ cm} \leq z \leq 15.8 \text{ cm}$, the distance of the measurement platform from the array is about 1.65 m and 0.95 m.

The boundaries of the tracking volume are tested by moving the transmitter on the measurement platform parallel to the X or Y axis, from $x=y=0$ until the point where the distance readings from at least two transducers are jittering. The location of this point is measured by a weight which is lowered from the correspondent beams of the reference frame that is by definition the X or Y axis. The locations of each of the extreme points are displayed at TABLE 6. IX.

TABLE 6. IX
THE LOCATION OF THE EXTREME POINTS

Point	X [m]	Y [m]	Z [m]
P1	0.18	0.00	1.10
P2	-.18	0.05	1.11
P3	0.03	0.18	1.12
P4	0.02	-.18	1.10
P5	0.25	0.00	1.81
P6	-.26	0.00	1.79
P7	0.00	0.245	1.81
P8	0.00	-.25	1.81

In addition, the jitter in the distance readings at each of the eight extreme points (four on each plane) are shown by TABLE 6. X. The value of X.X in TABLE 6. X denotes that the measured distance at this point is out of the range (which is described in Sec. 5.2.2) and is rejected by the software.

TABLE 6.X
THE CHANGES IN DISTANCE READINGS
AT THE EXTREME POINTS

	Rx. #1 [cm]	Rx. #2 [cm]	Rx. #3 [cm]	Rx. #4 [cm]	Rx. #5 [cm]
P1Ave	0.0	0.2	X.X	0.3	0.1
P1Max	0.0	0.7	X.X	1.4	0.4
P2Ave	0.1	X.X	0.0	0.2	0.8
P2Max	0.2	X.X	0.0	1.1	2.8
P3Ave	0.1	0.4	0.3	0.0	X.X
P3Max	0.6	1.1	0.8	0.0	X.X
P4Ave	0.0	0.1	0.2	X.X	0.0
P4Max	0.2	0.4	0.6	X.X	0.0
P5Ave	0.0	0.0	X.X	0.0	0.2
P5Max	0.0	0.1	X.X	0.2	0.5
P6Ave	0.0	0.3	0.0	0.0	0.1
P6Max	0.2	0.9	0.0	0.2	0.4
P7Ave	0.0	0.0	0.1	0.0	0.5
P7Max	0.2	0.4	0.5	0.2	2.6
P8Ave	0.0	0.0	0.1	0.8	0.0
P8Max	0.3	0.4	0.5	3.5	2.6

As can be observed from TABLE 6.IX, for a constant value of z the extreme value of x on y axis equals almost to the extreme value of x on y axis, therefore, for a constant value of z the tracking area is limited by a circle which its radius is the absolute value of the extreme value of x on y axis (or y on z axis). In addition, the assumption is that the radius of the circle is proportional to the value of z , and the tracking volume has the shape of a frustum which its volume, V , is calculated by

$$V = 1/3(\pi h(R_1^2 + R_2^2 + R_1 R_2)) \quad (6.3)$$

where h is the height of the frustum, R_1 and R_2 are the radiuses of the bases of the frustum. The height of the frustum, h , is calculated by

$$h = z_1 - z_2 \quad (6.4)$$

where z_1 and z_2 are the extreme points of the limb movement. Since the range of the limb movement is $96 \text{ cm} \leq z \leq 181 \text{ cm}$, h is calculated from Eq. (6.4) and equals to 85 cm. R_1 is given by TABLE 6.IX and equals to 25 cm. Since the size of the radius is proportional to the value of z , and for values of z which equal to 111 cm and 181 cm the radius sizes are 18 cm and 25 cm respectively, the size of the radius R_2 at $z=96 \text{ cm}$ can be calculated by rule of three and equals to 16.5 cm. Finally, the tracking volume is calculated by substituting in Eq. 6.3 the values of h with 0.85 m, R_1 with 0.25 m, R_2 with 0.165 m, then the rounded value of the volume $V=0.1165 \text{ m}^3$, (which is 43% of the theoretical volume which is calculated in Sec. 3.2.4).

The main reason for the decrease in the volume is because the readings of the distance measurement at the edges of the ultrasound beam are jittering. Thus, this part of the volume has to be deleted and the tracking volume becomes smaller.

6.2.4 Verification of the Location of a Target

In order to verify that a target is located properly in the tracking volume, the location of a target is checked at the eight extreme points (which are described in the previous section).

The groups of transducers, which are used to locate the target at each extreme point are selected by the following criterions (i) since the most distant receiver from the extreme point does not measure a stable distance the groups which include this receiver are eliminated, (ii) as described in Sec. 5.3.3 the groups in which all the three transducers are located on the same axis are not allowed.

Because of condition (i) there are only four receivers that can locate a target at an extreme point, and there are four possibilities to select three out of four receivers. One of the four possibilities is eliminated because of condition (ii). The locating with the three remaining groups is tested, and the results are displayed in TABLE 6.XIa to TABLE XIh. The position of the related points is given in TABLE 6.IX, and the list of the related transducers to a group (set) is given in TABLE 5.IV.

TABLE 6.XIa
THE LOCATION OF THE TARGET AT P1

	X [m]	Y [m]	Z [m]
Set1	0.18	0.04	1.10
Set2	0.19	-.02	1.11
Set8	0.18	-.00	1.10

TABLE 6.XIb
THE LOCATION OF THE TARGET AT P2

	X [m]	Y [m]	Z [m]
Set3	-.18	0.04	1.10
Set4	-.18	0.02	1.11
Set9	-.19	0.05	1.11

TABLE 6.XIc
THE LOCATION OF THE TARGET AT P3

	X [m]	Y [m]	Z [m]
Set1	-.03	0.17	1.12
Set3	0.03	0.18	1.12
Set6	0.00	0.20	1.12

TABLE 6.XId
THE LOCATION OF THE TARGET AT P4

	X [m]	Y [m]	Z [m]
Set2	0.01	-.19	1.10
Set4	0.06	-.19	1.10
Set7	0.04	-.20	1.10

TABLE 6.XIe
THE LOCATION OF THE TARGET AT P5

	X [m]	Y [m]	Z [m]
Set1	0.25	0.00	1.80
Set2	0.25	0.03	1.80
Set8	0.24	-.00	1.80

TABLE 6.XIf
THE LOCATION OF THE TARGET AT P6

	X [m]	Y [m]	Z [m]
Set3	-.27	0.01	1.80
Set4	-.26	0.00	1.80
Set9	-.26	-.01	1.79

TABLE 6.XIg
THE LOCATION OF THE TARGET AT P7

	X [m]	Y [m]	Z [m]
Set1	-.02	0.24	1.81
Set3	-.00	0.25	1.81
Set6	0.02	0.25	1.81

TABLE 6.XIh
THE LOCATION OF THE TARGET AT P8

	X [m]	Y [m]	Z [m]
Set2	0.03	-.26	1.81
Set4	0.01	-.26	1.81
Set7	0.00	-.25	1.81

It is obvious from TABLEs XI that the tracking system calculates properly the Z axis projection, and the locating error is not greater than 1 cm. On the other axes it is very crucial not to tilt the transmitter. Any slight tilt of the transmitter caused to erroneous and jittered readings. The reason for it is that when

$$\begin{aligned} d_i &= d_2 + dd, \quad d_i \gg dd, \\ d_i &\gg x_2, \quad i=1,2 \end{aligned} \quad (6.5)$$

where x_2 is the distance between two transducers, and d_i is the measured distance of transducer i from the target, then the projection on X axis, x_s , can be calculated by substituting in Eq. (5.4) the value of d_1 with d_2+dd , and equals to

$$x_s = \frac{d_2^2 + 2d_2dd + dd^2 + x_2^2 - d_2^2}{2x_2} \quad (6.6)$$

Since $d_2 \gg dd$ and $d_2 \gg x_2$, Eq. (6.6) can be simplified to

$$x_s \cong \frac{2d_2 dd}{2x_2} \quad (6.7)$$

If for some reason the distance measurements of d_1 are jittering, then value of dd is not stable too and the error in the projection on X axis is amplified by the relation of d_2/x_2 .

In the tracking system $x_2 \cong 25$ cm (dependent on the set of transducers which is selected for tracking) and the target is located usually further than 1 m from the array, therefore, any error in the measurement of d_1 can be amplified four times at least when x_s is calculated. Similarly, it can be proved that the projection on Y axis has the same problem. Solutions to this problem are discussed in Chap VII.

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

This project relates to a multiple-processor ultrasonic measurement and analysis system designed to track slow-moving objects within a small volume. The system was designed, implemented, tested with respect to its basic operation. It includes (i) a transmitter and an array of ultrasound receivers, (ii) control and signal processors, and (iii) a graphics oriented computer.

The system is capable of measuring the distances between the target and the array of six receiving transducers at the rate of 80 Hz. Since the system is designed to track the limb movement of disabled patients, the speed of their limb movement is relatively slow. If one assumes that the maximum velocity of the movement is less than 20 cm/s, then the target can change its location within 2.5 mm between two subsequent readings. This distance is half in its magnitude than the linearity error which was measured in Sec. 6.2.1. Therefore, the selection of 80 Hz as repetition rate for distance measurements, and the selection of the control processor is justified.

The target locating error along the Z coordinate has the same magnitude as the maximum linearity error of the distance measurement, and is 0.5 cm which is satisfactory for the system operation. Since the single ultrasound transmitter used have a narrow beam pattern, the receiving transducers have to be positioned close to one another. This requirement has resulted in a problem related to the target location along the X and Y coordinates, due to the linearity and jitter error amplification. This problem should be treated first by software (by heuristic

assumptions) and then in hardware by finding a transmitter with a wider beam pattern.

The target is located by triangulation, using the distances between the target and at least three points where the receiving transducers are located. The triangulation is done by the main computer. However, the present implementation of the locating algorithm requires more time than the time period between two distance measurements, the system must to collect the measured data first, and followed by the triangulation and the display the target trajectory. Since the data points from the control processor are transmitted to the main at the rate of 67200 characters per minute, the measuring time is limited by the memory which is available on the main computer for accumulation of the data. Since the present implementation of the BASIC interpreter used [Micr85] limits a string of characters to 32 K, a single string represents close to a 0.5 min period of limb movement.

Human interfacing can be simplified by the use of graphics interface embedded in a specific operating system, as available in the main computer selected for this project. Appropriate operating menus for human interface are described in Sec. 5.2. The main computer pixel and object drawing routines can also be used to simplify the display of both the tracking volume and the trajectory of the limb movement.

Although the system must cope with the narrow beam pattern of the transmitter, its implementation is relatively simple when compared to other systems with other tracking methods.

REFERENCES

- [AlIn86] J. Allin, and G.F. Inbar, "FNS Parameter Selection and Upper Limb Characterization," *IEEE Trans. Biomed. Eng.*, vol. BME-33, No. 9, pp. 809-817, 1986.
- [AmLA84] A. Amit, G. Langholtz, and M. Acran, "An Approach to Control Laws for Arm Motion," *IEEE Trans. Biomed. Eng.*, vol. BME-31, No. 9, pp. 605-610, 1984.
- [Anto81] H. Anton, *Elementary Linear Algebra*. New York: John Wiley and Sons, 1981, (pp. 196-211) 339 pp.
- [BESS80] G. Biber, S. Ellin, E. Shenk, and Stempeck, "The Polaroid Ultrasonic Ranging System," *AES 67th Convention*; New York, Oct31-Nov3, 1980, 11 pp.
- [Beye78] W. H. Beyer, *CRC Standard Mathematical Tables*, CRC Press, West Palm Beach, pp. 416-422, April 1978.
- [Ciar80] S. Ciarcia, "Home In on the Range! An Ultrasonic Ranging System," *Byte*, pp. 32-58, November 1980.
- [Ciar84] S. Ciarcia, "An Ultrasonic Ranging System," *Byte*, pp. 113-123, October 1984.
- [DoFf83] A. P. Dowling, and J. E. Ffowcs Williams, *The Sound and Sources of Sound*, West Sussex, Chichester: Ellis Horwood, 1983, (pp. 139-144), 321 pp..
- [GrWr81] J. D. Greenfield, and W. C. Wray, *Using Microprocessors and Microcomputers: The 6800 Family*, John Wiley and Sons, 1981, 460 pp.
- [Inte79] "The 8086 and 8088 Central Processing Unit," Intel, *The 8086 Family User's Manual*, Oct 1979, pp. (2)1-128.
- [John86] T. L. Johnson, "A Comparison of MC68000 Family Processors," *Byte*, pp. 205-218, September 1986.

- [KeeK80] "The KEE KLAMP System of Pipe Construction," Kee Klamp Division Gascoigne Industrial Products Ltd., KK180, 24 pp., 1980.
- [KFCS82] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders, *Fundamentals of Acoustics*, New York: John Wiley and Sons, 1982, (pp. 163-197) 480 pp.
- [KlKP87] R. Klein, W. Kinsner, and J. J. Pear, "A System for 3D Tracking of Limb Movement," *Ninth Annual Conference of the IEEE Engineering in Medicine and Biology Society*; Boston, Nov 13-16, 1987, pp. 1990-1991.
- [KnCa76] C. H. Knapp, and G. C. Carter, "The Generalized Correlation Method for Estimation of Time Delay," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 320-327, Aug. 1976.
- [Lamb85] S. Lambert, *Creative Programming in Microsoft Basic for Optimal Macintosh Performance*, Bellevue, Washington: Microsoft Press, 1985, 535 pp.
- [Mass65] F. Massa, "Ultrasonic transducers for use in the air," *IEEE Proc. (Special Issue on Ultrasonics)*, vol. 55, no. 10, October 1965, pp. 1363-1371.
- [Micr85] "Microsoft BASIC, Interpreter for Apple Macintosh," Microsoft Corp., Bellevue, Washington, 1985, 311 pp.
- [Micr86] "Microsoft BASIC, Compiler for Apple Macintosh," Microsoft Corp., Bellevue, Washington, 1986, 466 pp.
- [NgBa86] L. C. Ng, and Y. Bar-Shalom, "Multisensor Multitarget Time Delay Vector Estimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 4, pp. 669-678, Aug. 1986.
- [Pear85] Joseph J. Pear, "Spatiotemporal Patterns of Behavior Produced by Variable-Interval Schedules of Reinforcement," *Journal of the experimental Analysis of Behavior*, vol. 44, no. 2, 1985, pp. 217-231.

- [PeLe87] J. J. Pear, and J. A. Legris, "Shaping by automated tracking of an arbitrary operant response," *Journal of the experimental Analysis of Behavior*, vol. 47, March 1987, pp. 241-247.
- [Pola85] "Environmental Ultrasonic Transducer," Polaroid, P2759, ITP-49, (11/85).
- [ScRo87] H. C. Schau, and A. Z. Robinson, "Passive Source Localization Employing Intersecting Spherical Surfaces from Time-of-Arrival Differences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 8, pp. 1223-1225, Aug. 1987.
- [ShHs79] W. Shacse, and N. N. Hsu, "Ultrasonic Transducers for Materials Testing and Their Characterizations," in *Physical Acoustic*, Mason and Thurson, vol. XIV, pp. 277-406, New York: Academic Press, 1979.
- [Tek78] "8002A μ Processor Lab," Tektronix Inc., *System User Manual*, Beaverton, Oregon, July 87.
- [Texa84] "Type SN28827 Sonar Ranging Module," Texas Instruments, *Linear Circuits Data Book*, 1984, pp. (5)45-47.
- [UofM86] "6802 Thesis Board 2," University of Mantoba, Electrical Engineering, Winnipeg, Canada, 25/02/86, 3 pp..
- [Webb86] A. D. Webber, "Amiga Vs. Macintosh," *Byte*, pp. 205-218, September 1986.

Appendix A

Control Processor Diagrams

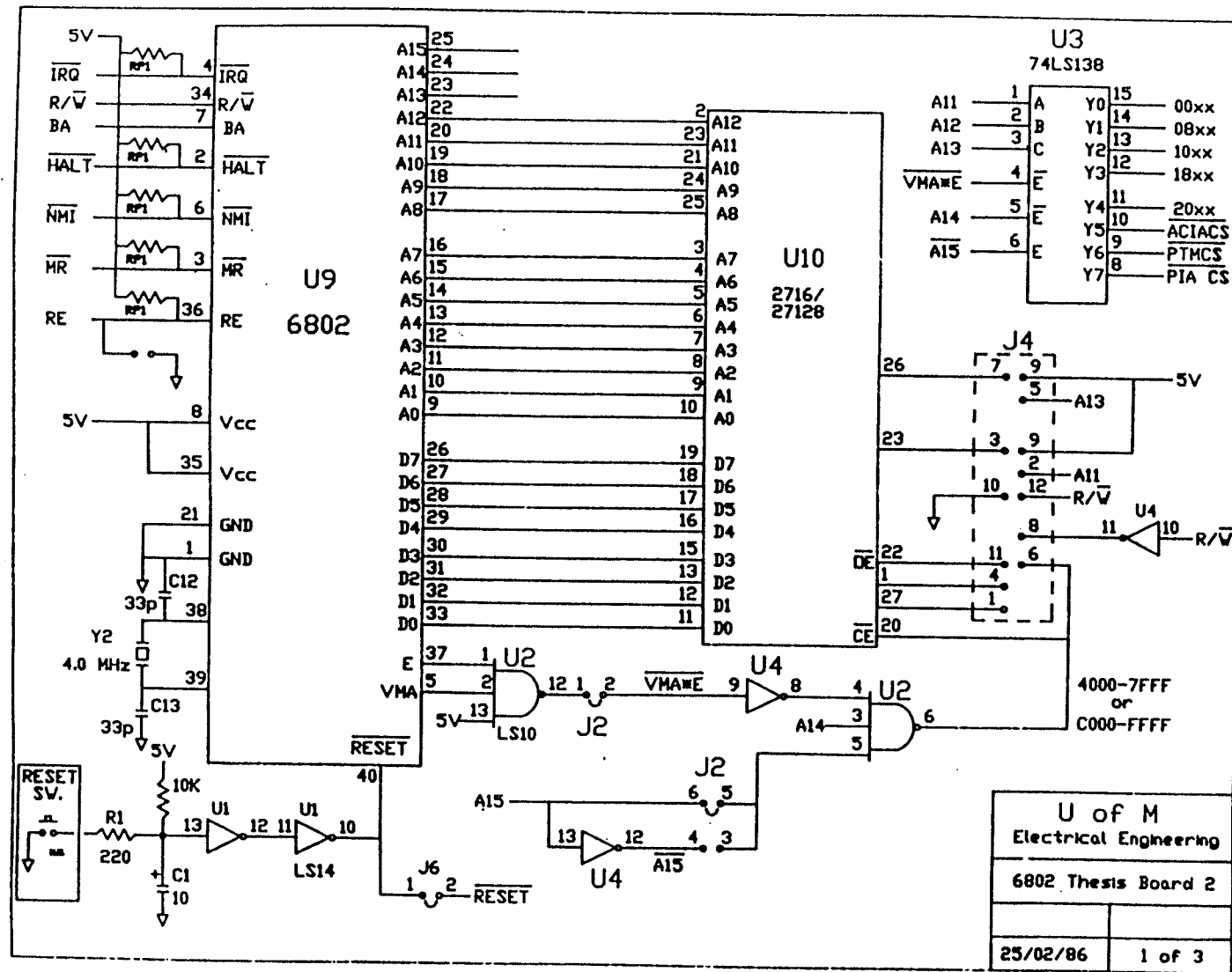


Fig. A.1.1. 6802 Thesis Board 2 (sheet 1 of 3), From [UofM86].

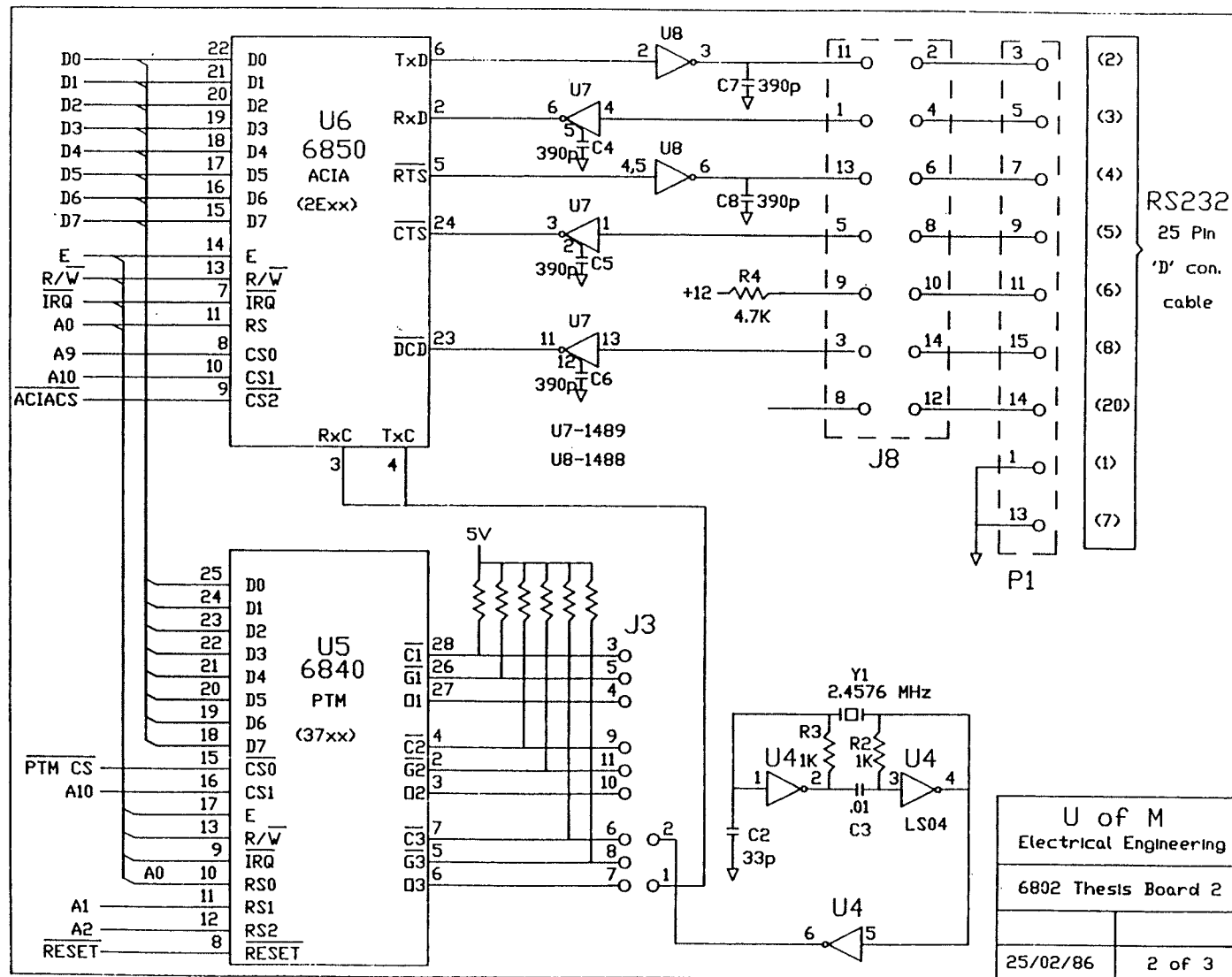


Fig. A.1.2. 6802 Thesis Board 2 (sheet 2 of 3), From [UofM86].

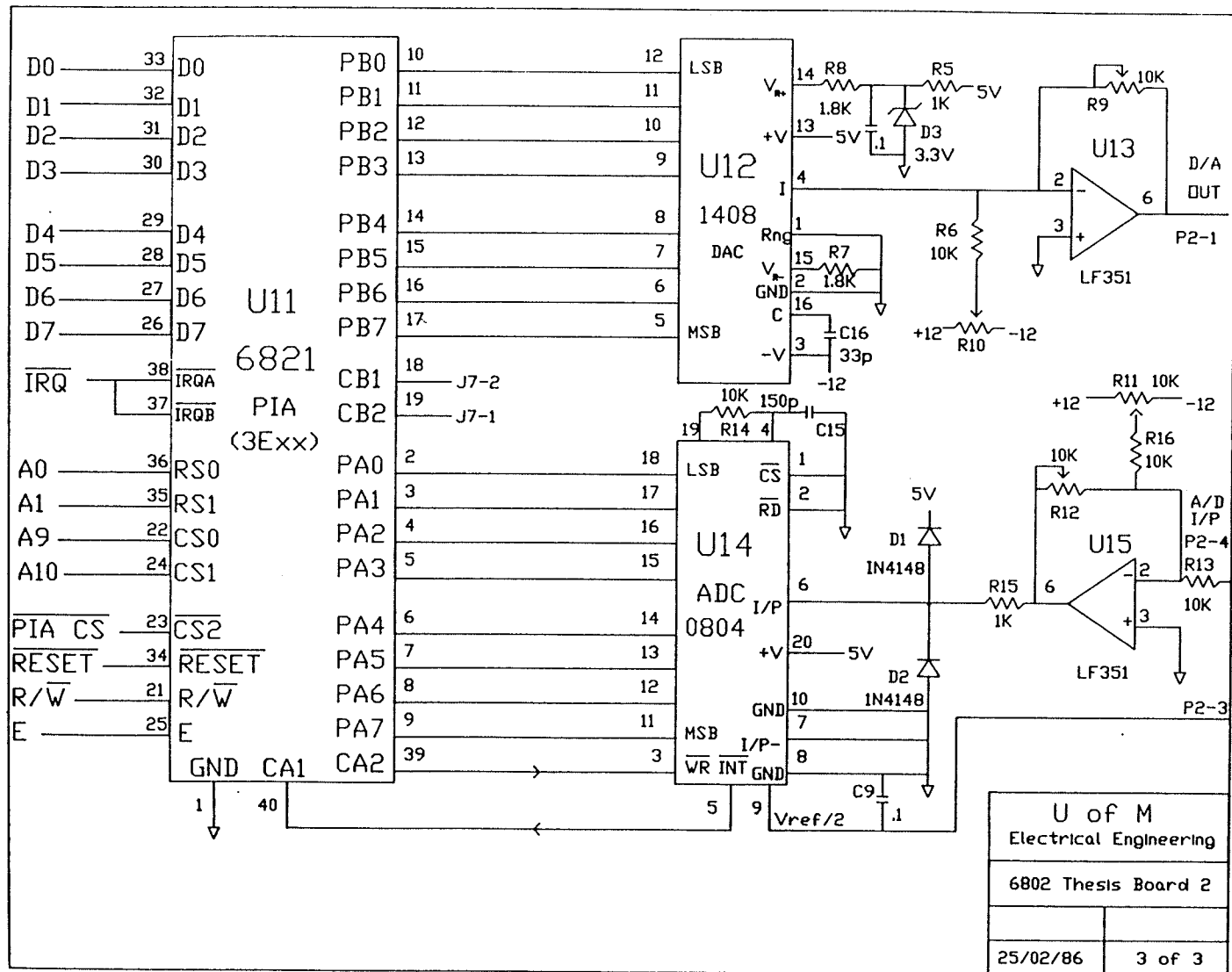


Fig. A.1.3. 6802 Thesis Board 2 (sheet 3 of 3), From [UofM86].

TO SH2-H5 << D0-D7

44 PIN
EDGE CONNECTOR

Y << 9
U << 25
U << 24
T << 23
S << 22
R << 21
P << 20
N << 19
M << 18

21 << 17
19 << 13
B << 16
A0 << 10
A1 << 11
A2 << 12
6 << 8

RESET

BLOAD

FROM SH3-A3 << RESET

TO SH2-H3 << CLINE
TO SH2-H3 << A0-A2

44 PIN
EDGE CONNECTOR

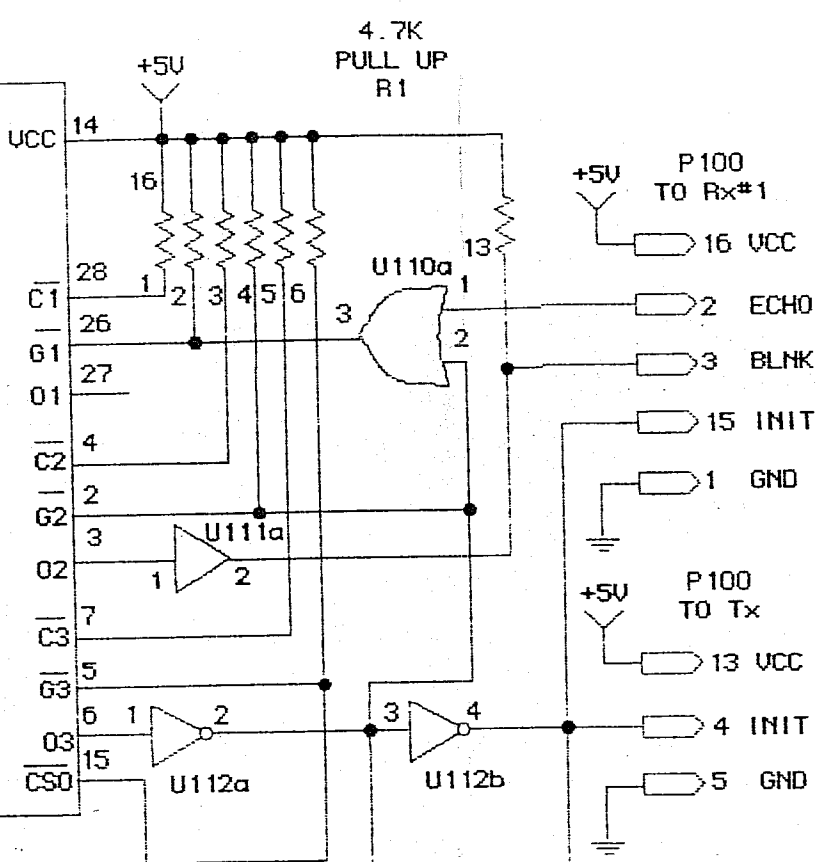
K << A10
8 << A11
H << A12

7 << A13
J << A14
L << A15

U101-PTM
MC6840
ADDRESS
8000-8007

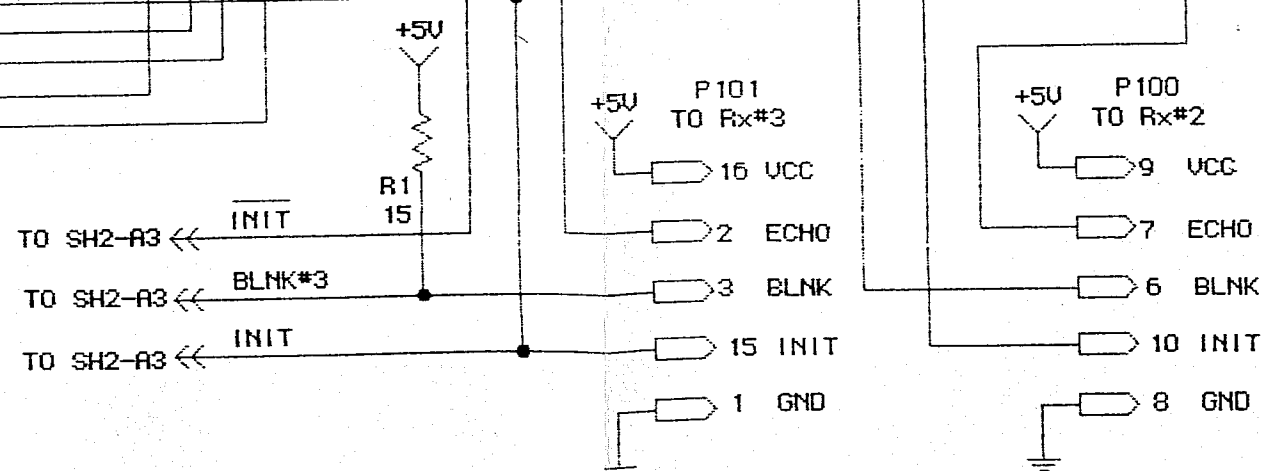
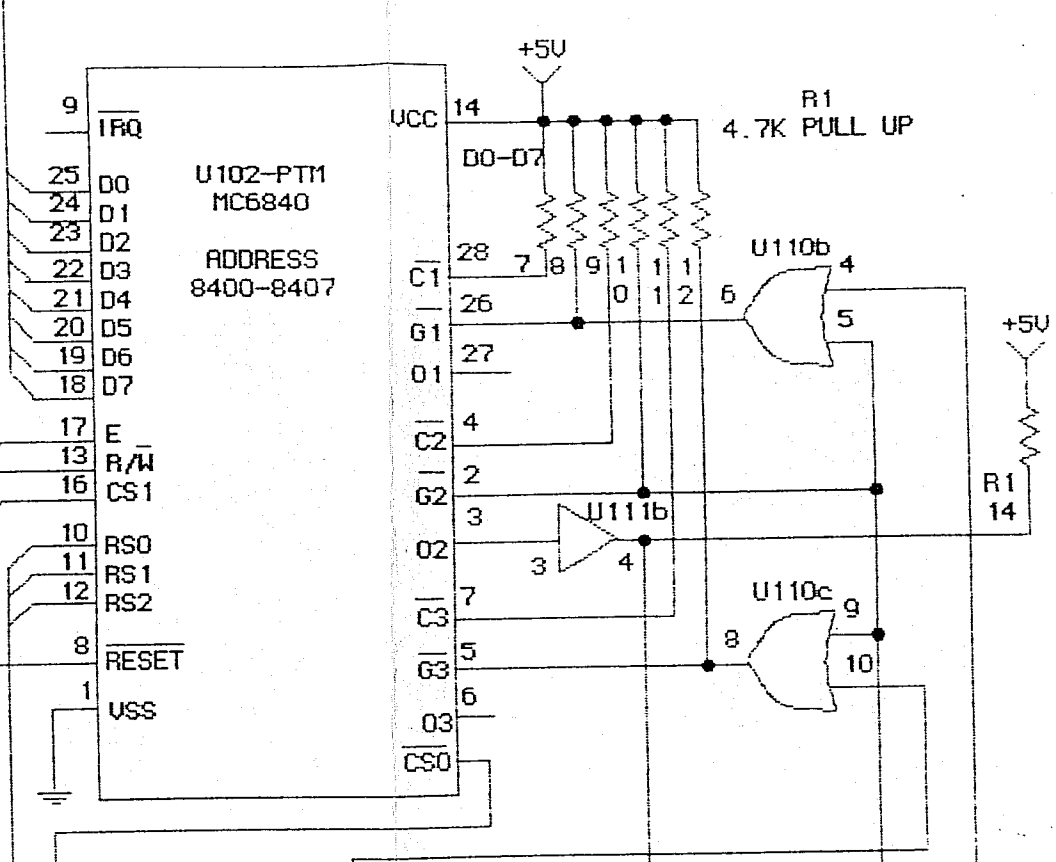
17 E
13 R/W
16 CS1
10 RS0
11 RS1
12 RS2
8 RESET
1 USS

U100
DECODER
74LS138
A0 00
A1 01
A2 02
03
04
05
06
07
E1
E2
E3
GRD 8



U102-PTM
MC6840
ADDRESS
8400-8407

9 IRQ
25 D0
24 D1
23 D2
22 D3
21 D4
20 D5
19 D6
18 D7
17 E
13 R/W
16 CS1
10 RS0
11 RS1
12 RS2
8 RESET
1 USS



UNIVERSITY OF MANITOBA

Fig. A.2.1. Timers Board.

PTM#1, PTM#2	1	0	0
SHEET 1 OF 3	28 JULY 86		

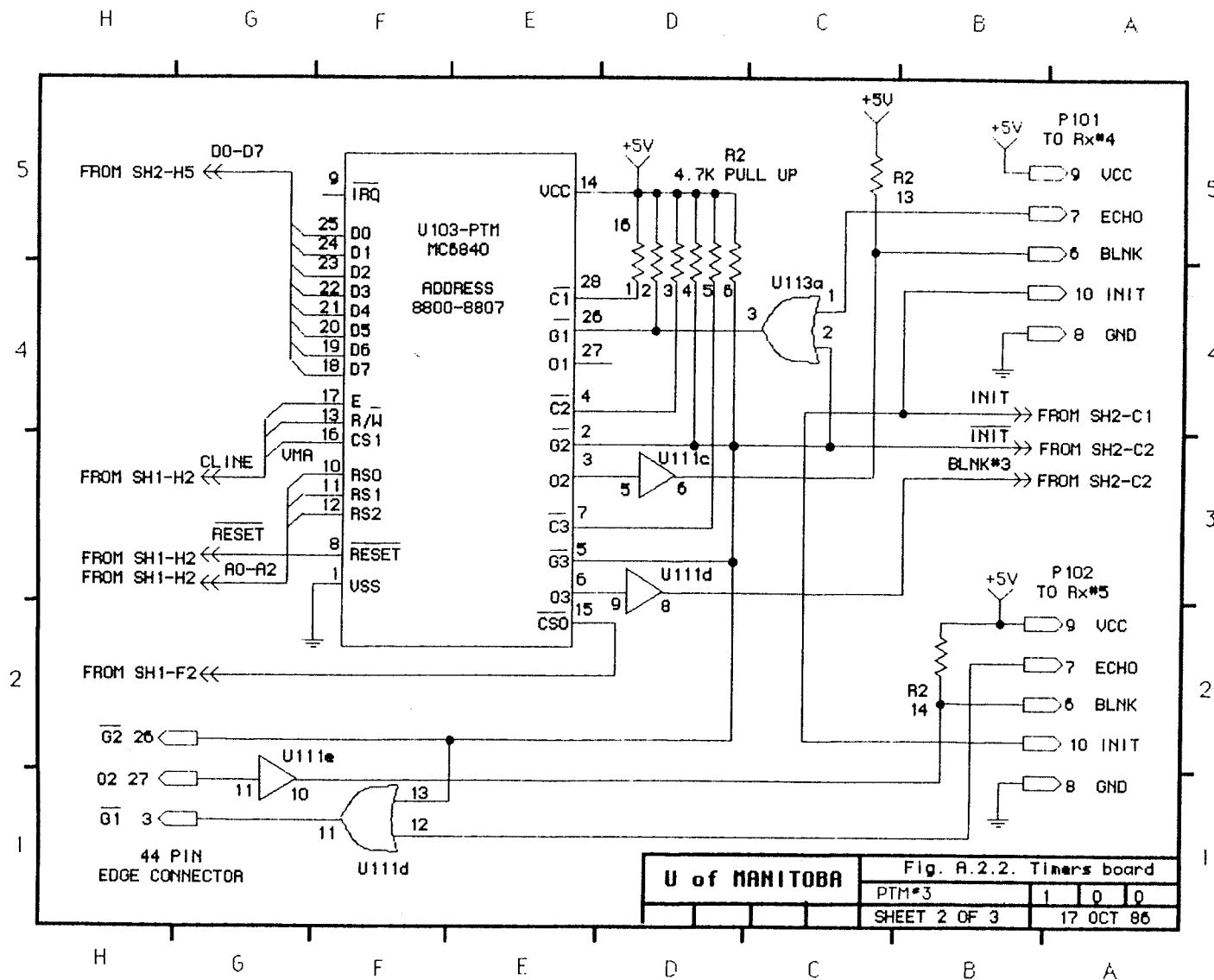


Fig. A.2.2. Timers board (sheet 2 of 3).

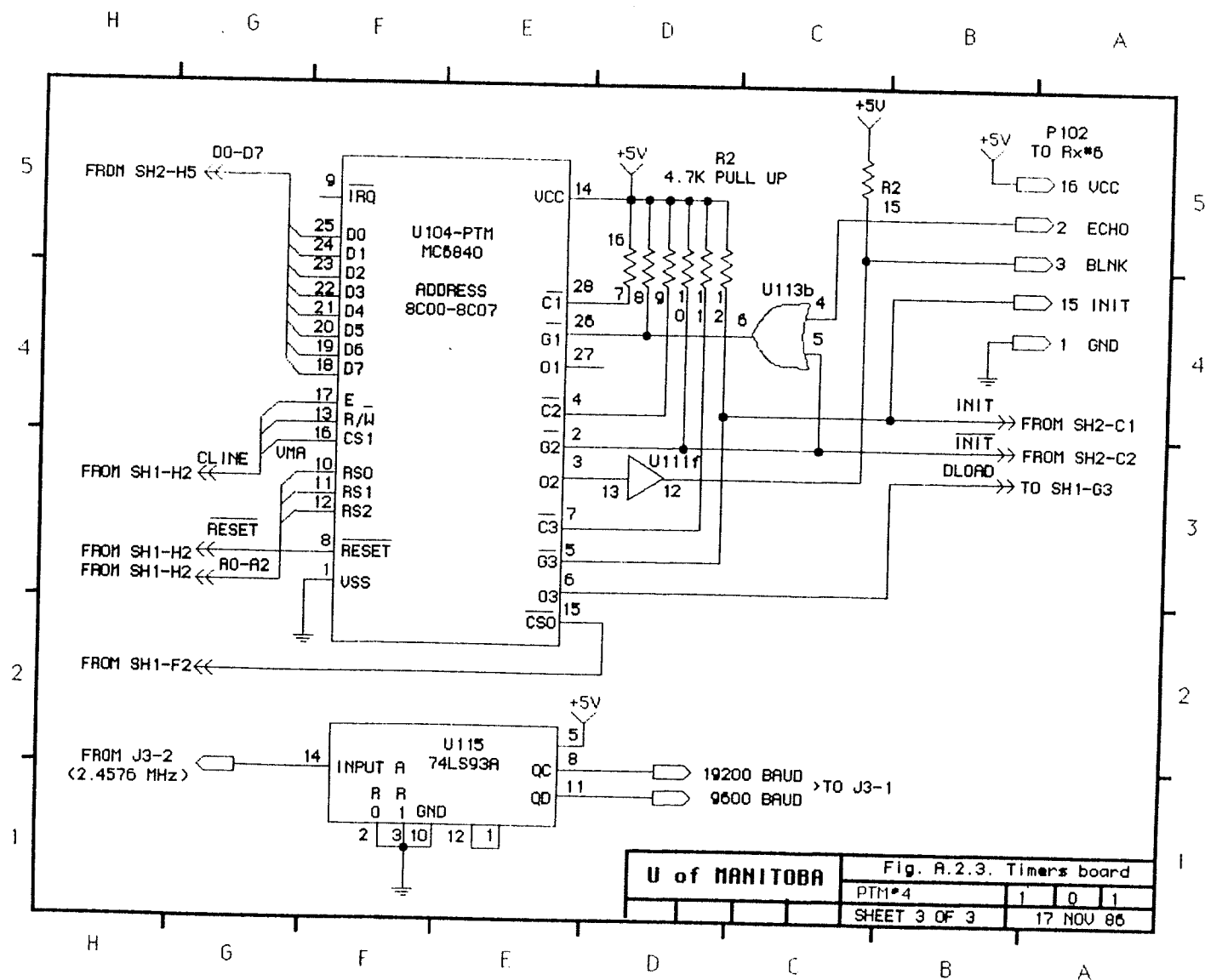


Fig. A.2.3. Timers board (sheet 3 of 3).

Appendix B

Control Processor Program Listings

```

00001          NOLIST ME
00003          ;
00004          ;*****
00005          ;
00006          ;   SYSTEM CONSTANTS
00007          ;
00008          ;*****
00009          ;
00010          GLOBAL  IRQ,ACIA,MODE,ECHOB
00011          GLOBAL  ACCNT,ACIAB,BLANKB,TRCNT
00012          GLOBAL  TXCNT,TXMAX,TXPONT
00013          GLOBAL  TIMER,PTM1,PTM2,PTM3,PTM4
00014          ;
00015          2E00  ACIA  EQU  2E00H          ; ACIA CONTROL REG AT 2E00H
00016          0006  NRX   EQU  6             ; NUMBER OF RX IS 6
00017          8000  PTM1  EQU  8000H          ; U101 PTM BASE ADR. IS 8000H
00018          8400  PTM2  EQU  8400H          ; U102 PTM BASE ADR. IS 8400H
00019          8800  PTM3  EQU  8800H          ; U103 PTM BASE ADR. IS 8800H
00020          8C00  PTM4  EQU  8C00H          ; U104 PTM ADR. IS 8C00H
00021          F800  START EQU  0F800H         ; THE PROGRAM WILL START AT F800H
00022          3700  TIMER EQU  3700H          ; PTM BASE ADDR IS 3700H
00023          000C  TNRX  EQU  12             ; NRX*2=12
00024          ;
00025          SECTION A1,ABSOLUTE
00026          0000  RAM   EQU  0             ; RAM DEFINITIONS
00027          0000  ORG   RAM
00028          0000 0000  ACCNT WORD  0         ; ACIA RX WORD COUNTER
00029          0002 0004  ACIAB BLOCK  4        ; FOUR BYTES ARE RESERVED FOR
00030                                     ; ACIA RECEIVER
00031          0006 000C  BLANKB BLOCK  12       ; 12 BYTES ARE RESERVED FOR
00032                                     ; BLANK DATA
00033          0012 000D  ECHOB BLOCK  13       ; 13 BYTE ARE RESERVED FOR
00034                                     ; ECHO DATA
00035          001F 00    MODE  BYTE  0         ; MODE DEFINITIONS
00036                                     ; 0 WAITING FOR MACINTOSH
00037          0020 0000  TRCNT WORD  0         ; TRACK COUNTER
00038          0022 00    TXCNT BYTE  0        ; ACIA TX CHAR. COUNTER
00039          0023 00    TXMAX BYTE  0        ; NUMBER OF WORDS TO BE TRANSMITTED
00040          0024 0000  TXPONT WORD  0       ; ADRESS OF THE CHAR. FOR TX

```

```
00042      ;*****
00043      ;
00044      ;      MACRO INSTRUCTIONS
00045      ;
00046      ;*****
00047      ;
00048      MACRO   ISRM      ; INITIALIZE SONAR RANGING
00049      LDA     A         #9AH  ; THE ECHO TIME IS MEASURED
00050      STA     A         '1'   ; BY OUTPUT ENABLE
00051      ;
00052      ; INT DISABLED
00053      ; SINGLE PULSE IN TRIGERED
00054      ; BY FALLING EDGE OF GATE (INIT)
00055      ; USES INTERNAL CLOCK 16 BIT
00056      LDA     A         #0B2H ; THE BLANK TIME IS ACTIVATED
00057      STA     A         '2'   ; BY OUTPUT ENABLE
00058      ;
00059      ; INT DISABLED
00060      ; SINGLE PULSE OUT TRIGERED
00061      ; BY FALLING EDGE OF GATE (INIT)
00062      ; USES INTERNAL CLOCK 16 BIT
00063      ; POINTER TO MODULE 3
00064      ENDM
00065      ;
```

```

00065 ;*****
00066 ;
00067 ;   MAIN PROGRAM
00068 ;
00069 ;*****
00070 ;
00071 ;   SECTION A2,ABSOLUTE
00072     FFF8     ORG     0FFF8H      ; IRQ VECTOR PLACE
00073 FFF8 0000 >   WORD     IRQ
00074 ;
00075 ;   SECTION A3,ABSOLUTE
00076     FFFE     ORG     0FFFEH      ; RESET VECTOR PLACE
00077 FFFE F800     WORD     START
00078 ;
00079 ;   SECTION A4,ABSOLUTE
00080     F800     ORG     START      ; MAIN PROGRAM STARTS HERE
00081 F800 8E007F   START1 LDS     #7FH      ; SP THE LAST RAM
00082 F803 0F      SEI              ; DISABLE IRQ
00083 F804 8600    LDA     A        #0      ; SET POINTER TO CONTROL
00084 F806 B73701 STA     A        TIMER+1 ; REGISTER#3 OF TIMER
00085 F809 8681    LDA     A        #81H    ; SET PTM#3 TO /8
00086 F80B B73700 STA     A        TIMER    ; AND TIMER ENABLE
00087 F80E CE000F  LDX     #0FH      ; PTM#3 WILL DEVIDE BY 32
00088 F811 FF3706 STX     TIMER+6 ; FOR 9600 BAUDS
00089 F814 8601    LDA     A        #1      ; SET POINTER TO U5 PTM#1
00090 F816 B73701 STA     A        TIMER+1
00091 ;
00092 F819 8603     LDA     A        #03      ; RESET ACIA
00093 F81B B72E00  STA     A        ACIA
00094 F81E 86B5    LDA     A        #0B5H   ; ACIA IS SET AS : TX,RX INTERRUPT
00095 F820 B72E00  STA     A        ACIA     ; 8 BIT+N*STOP DEVIDE BY 16
00096 F823 CE0002 LDX     #ACIAB    ; RESET RX COUNTER
00097 F826 DF00    STX     ACCNT
00098 ;
00099 F828 8601     LDA     A        #01      ; SET POINTER TO CONTROL
00100 F82A B78001  STA     A        PTM1+1 ; REG#1 OF U101
00101     ISRM     PTM1,PTM1+1 ; INITIALIZE PTM FOR P101
00102 ;
00103 F837 8601     LDA     A        #01      ; SET POINTER TO CONTROL
00104 F839 B78401  STA     A        PTM2+1 ; REG#1 OF U102
00105     ISRM     PTM2,PTM2+1 ; INITIALIZE PTM FOR P102
00106 ;
00107 F846 8601     LDA     A        #01      ; SET POINTER TO CONTROL
00108 F848 B78801  STA     A        PTM3+1 ; REG#1 OF U103
00109     ISRM     PTM3,PTM3+1 ; INITIALIZE PTM FOR P104
00110 ;
00111     ISRM     PTM2,PTM3      ; INITIALIZE PTM FOR P103
00112 ;
00113     ISRM     TIMER,TIMER+1 ; INITIALIZE PTM FOR P105
00114 ;
00115 F869 8601     LDA     A        #01      ; SET POINTER TO CONTROL
00116 F86B B78C01  STA     A        PTM4+1 ; REG#1 OF U104
00117     ISRM     PTM4,PTM4+1 ; INITIALIZE PTM FOR P106

```

```
00118 F878 86B2          LDA    A      #0B2H    ; LOAD TIMER IS WAITING
00119 F87A B78C00        STA    A      PTM4     ; FOR END OF INIT
00120                      ;
00121 F87D BDF888          JSR    ZBLKBU        ; ZERO BLANK BLOCK
00122 F880 4F             CLR    A
00123 F881 971F          STA    A      MODE     ; MODE=0
00124 F883 0E             CLI
00125 F884 20FE          LOOP  BRA    LOOP     ; ENABLE INTERRUPT
00126 F886 20FC          BRA    LOOP     ; WAIT FOR IRQ
00127                      ;
```

```
00129      ;*****
00130      ;
00131      ;      SUBROUTINE INSTRUCTIONS
00132      ;
00133      ;*****
00134      ;
00135 F888 CE0006 ZBLKBU LDX      #BLANKB      ; BLANK BLOCK IS ZEROED BY
00136 F88B C60C   LDA      B      #TNRX      ; THIS ROUTINE
00137 F88D 4F     CLR      A
00138 F88E A700   L1      STA      A      0,X      ; LOAD TILL B IS 0
00139 F890 08     INX
00140 F891 5A     DEC      B
00141 F892 26FA   BNE      L1
00142 F894 39     RTS
00143      ;
00144      END
```

Strings and Macros

ISRMI --- 0096 M

Scalars

ACIA --- 2E00 G
PTM2 --- 8400 G
PTM4 --- 8C00 G
TIMER -- 3700 G

NRX ---- 0006
PTM3 --- 8800 G
RAM ---- 0000
TNRX --- 000C

PTM1 --- 8000 G
START -- F800

A1 Section Absolute (0026)

ACCNT -- 0000 G
ECHOB -- 0012 G
TRCNT -- 0020 G
TXPONT - 0024 G

ACIAB -- 0002 G
MODE --- 001F G
TXCNT -- 0022 G

BLANKB - 0006 G
TXMAX -- 0023 G

A2 Section Absolute (FFFA)

A3 Section Absolute (10000)

A4 Section Absolute (F895)

L1 ----- F88E
ZBLKBU - F888

LOOP --- F884

START1 - F800

IRQ Unbound Global

144 Source Lines 174 Assembled Lines 14039 Bytes available

>>> No assembly errors detected <<<


```
00001          NOLIST ME
00003          ;
00004          ;*****
00005          ;
00006          ;      SYSTEM CONSTANTS
00007          ;
00008          ;*****
00009          ;
00010          GLOBAL  IRQ,MODE,ACIA,TXON
00011          GLOBAL  ACCNT,ACIAB,SRCHM,TRACKM
00012          GLOBAL  TXCNT,TXMAX,TXPONT
00013          ;
```

```

00015      ;*****
00016      ;
00017      ;      MACRO INSTRUCTIONS
00018      ;
00019      ;*****
00020      ;
00021      MACRO   LETCH      ; CHECK IF THE CHARACTER IS
00022      LDA     A          0,X      ; EQUALE TO VAR 1
00023      CMP     A          #'1'    ; IF YES JUMP TO VAR 2
00024      BEQ     '2'
00025      ENDM
00026      ;
00027      MACRO   NOTCH
00028      INX
00029      LDA     A          0,X      ; INCREMENT ADRESS COUNTER
00030      CMP     A          #'1'    ; LOAD DATA
00031      BNE     ERROR        ; CHECK EQUALITY TO VAR 1
00032      ENDM        ; IF NOT JUMP TO ERROR
00033      ;
00034      MACRO   EOL          ; CHECK FOR END OF LINE
00035      NOTCH   0AH          ; LF IS CHECKED
00036      NOTCH   0DH          ; CR IS CHECKED
00037      ENDM
00038      ;
00039      MACRO   CHMODE
00040      LDA     A          MODE      ; CHECK FOR MODE
00041      AND     A          #0F0H    ; AND JUMP ACCORDINGLY
00042      CMP     A          #'1'
00043      BNE     NEXT'3'
00044      JMP     '2'
00045      NEXT'3'      NOP
00046      ENDM
00047      ;

```

```

00049      ;*****
00050      ;
00051      ;      MAIN PROGRAM
00052      ;
00053      ;*****
00054      ;
00055      SECTION B1,ABSOLUTE
00056      F900      ORG      0F900H      ; POLL PROGRAM STARTS HERE
00057 F900 B60000 >  IRQ      LDA      A      MODE      ; CHECK FOR MODE
00058 F903 85F0      BIT      A      #0F0H
00059 F905 2606      BNE      OTHER1      ; IF NOT MODE=0X JUMP
00060 F907 8501      BIT      A      #01      ; CHECK FOR TX
00061 F909 2605      BNE      TXON2
00062 F90B 2006      BRA      IRQ1      ; RANGE ADJUSTMENT
00063 F90D 7EF9E0      OTHER1 JMP      OTHER
00064 F910 7EF9A2      TXON2 JMP      TXON
00065      ;
00066 F913 B60000 >  IRQ1      LDA      A      ACIA      ; CHECK FOR ACIA RX
00067 F916 8481      AND      A      #81H      ; INTERRUPT
00068 F918 8181      CMP      A      #81H
00069 F91A 2628      BNE      IRQEND
00070 F91C FE0000 >      LDX      ACCNT      ; CHECK FOR MAX ADDRESS OF RX
00071 F91F 8C0004 >      CPX      #ACIAB+4      ; BUFFER.
00072 F922 2720      BEQ      IRQEND
00073 F924 B60001 >      LDA      A      ACIA+1      ; LOAD DATA FROM ACIA AND
00074 F927 A700      STA      A      0,X
00075 F929 08      INX      ; INCREMENT ADDRESS COUNTER
00076 F92A FF0000 >      STX      ACCNT
00077 F92D 8C0004 >      CPX      #ACIAB+4
00078 F930 2612      BNE      IRQEND      ; END OF CHARACTER LOADING
00079      ;
00080 F932 CE0000 >      LDX      #ACIAB      ; CHECK THE RECEIVED WORD
00081      LETCH      "S",SEARCH      ; IF THE FIRST CHARACTER IS
00082      ; S GO TO SEARCH
00083      LETCH      "T",TRACK      ; IF THE FIRST CHARACTER IS
00084      ; T GO TO TRACK
00085 F941 7EF986      JMP      ERROR      ; NOT VALID CODE
00086 F944 3B      IRQEND RTI      ; END OF POLL ROUTINE
00087      ;
00088      ; SEARCHING ROUTINE
00089      SEARCH NOTCH      "0"      ; IF SECOND CHARACTER IS NOT
00090      ; 0 THEN GOTO ERROR
00091      EOL      ; CHECK FOR END OF LINE
00092 F95A 8610      LDA      A      #10H      ; LOAD MODE=10H
00093 F95C B70000 >      STA      A      MODE
00094 F95F 7E0000 >      JMP      SRCHM      ; GOTO SEARCHING MODE
00095      ;
00096 F962 08      TRACK INX      ; TRACKING ROUTINE
00097 F963 A600      LDA      A      0,X      ; LOAD RANGE AND CHECK IF
00098 F965 16      TAB      ; IT IS BETWEEN 0-3
00099 F966 84F0      AND      A      #0F0H      ; CHECK IF LSB NIBBLE IS 3
00100 F968 8130      CMP      A      #30H
00101 F96A 261A      BNE      ERROR

```

```

00102 F96C C035          SUB      B      #35H      ; CHECK IF NUMBER IS GREATER
00103 F96E 2C16          BGE      ERROR          ; THEN 4 IF YES JUMP
00104                   EOL                      ; CHECK FOR END OF LINE
00105 F97E 8620          LDA      A      #20H      ; LOAD MODE=20H
00106 F980 B70000 >      STA      A      MODE
00107 F983 7E0000 >      JMP      TRACKM          ; GOTO TRACKING MODE
00108                   ;
00109 F986 8601          ; ERROR LDA      A      #01      ; CHANGE MODE TO 01
00110 F988 B70000 >      STA      A      MODE
00111 F98B C606          LDA      B      #06      ; TRANSMIT 6 CHARACTERS
00112 F98D F70000 >      STA      B      TXMAX
00113 F990 7F0000 >      CLR      TXCNT          ; TX COUNTER=0
00114 F993 CEF99C          LDX      #ERPNT        ; LOAD ERROR POINTER
00115 F996 FF0000 >      STX      TXPONT
00116 F999 7EF9AB          JMP      TXON3          ; SEND TO HOST COMPUTER
00117                   ; ERROR
00118 F99C 4552524F      ERPNT ASCII "ERROR"
00118 F9A0 52
00119 F9A1 0D           BYTE      0DH          ; CARRIGE RETURN
00120                   ;
00121 F9A2 B60000 >      TXON  LDA      A      ACIA      ; CHECK FOR ACIA TX
00122 F9A5 8482          AND      A      #82H      ; INTERRUPT
00123 F9A7 8182          CMP      A      #82H
00124 F9A9 2699          BNE      IRQEND
00125 F9AB F60000 >      TXON3 LDA      B      TXCNT      ; CHECK FOR MAX WORD
00126 F9AE F10000 >      CMP      B      TXMAX
00127 F9B1 261A          BNE      TXON1
00128 F9B3 B60000 >      LDA      A      MODE      ; IF MODE IS 23 THEN
00129 F9B6 8123          CMP      A      #23H      ; IT IS INCREMENTED
00130 F9B8 2606          BNE      MODNE
00131 F9BA 7C0000 >      INC      MODE
00132 F9BD 7EF944          JMP      IRQEND
00133 F9C0 4F           MODNE CLR      A          ; ZERO MODE IF END OF
00134 F9C1 B70000 >      STA      A      MODE      ; TX
00135 F9C4 CE0000 >      LDX      #ACIAB          ; RESET ACIA RX
00136 F9C7 FF0000 >      STX      ACCNT          ; COUNTER
00137 F9CA 7EF944          JMP      IRQEND
00138 F9CD FE0000 >      TXON1 LDX      TXPONT      ; LOAD POINTER
00139 F9D0 A600          LDA      A      0,X      ; LOAD DATA
00140 F9D2 B70001 >      STA      A      ACIA+1    ; SEND DATA
00141 F9D5 5C           INC      B          ; INCREMENT RX COUNTER
00142 F9D6 F70000 >      STA      B      TXCNT
00143 F9D9 08           INX
00144 F9DA FF0000 >      STX      TXPONT          ; INCREMENT POINTER
00145 F9DD 7EF944          JMP      IRQEND
00146                   ;
00147                   ; OTHER CHMODE 10H,SRCHM      ; CHECK OTHER MODES
00148                   CHMODE 20H,TRACKM
00149 F9FA 7EF944          JMP      IRQEND
00150                   ;
00151                   END

```

Strings and Macros

CHMODE - 0076 M	EOL ---- 003F M	LETC - 0050 M
NOTCH -- 0089 M		

B1 Section Absolute (F9FD)

ERPNT -- F99C	ERROR -- F986	IRQ ---- F900 G
IRQ1 --- F913	IRQEND - F944	
MODNE -- F9C0	NEXTA000 F9EC	NEXTB000 F9F9
OTHER -- F9E0	OTHER1 - F90D	
SEARCH - F945	TRACK -- F962	TXON --- F9A2 G
TXON1 -- F9CD	TXON2 -- F910	
TXON3 -- F9AB		

ACCNT Unbound Global

ACIA Unbound Global

ACIAB Unbound Global

MODE Unbound Global

SRCHM Unbound Global

TRACKM Unbound Global

TXCNT Unbound Global

TXMAX Unbound Global

TXPONT Unbound Global

151 Source Lines 204 Assembled Lines 13786 Bytes available

>>> No assembly errors detected <<<

```
00001          NOLIST  ME
00003          ;
00004          ;*****
00005          ;
00006          ;      SYSTEM CONSTANTS
00007          ;
00008          ;*****
00009          ;
00010          GLOBAL  MODE,SRCHM,TIMER,PTMINT
00011          GLOBAL  PTM1,PTM2,PTM3,ACIA,PTM4
00012          GLOBAL  TXCNT,TXMAX,TXPONT,LDPTM
00013          GLOBAL  TXON,ECHOB,BLANKB,FLT
00014          ;
00015          0032    FLT    EQU    50
00016          30D4    MS100  EQU    30D4H          ; 100MS TIME CONSTANT
00017          ;
```

```

00019      ;*****
00020      ;
00021      ;      MACRO INSTRUCTIONS
00022      ;
00023      ;*****
00024      ;
00025      MACRO      CHMODE      ; CHECK IF MODE IS VAR1
00026      LDA      A      MODE      ; IF YES JUMP TO VAR2
00027      CMP      A      #'1'      ; ELSE CONTINUE
00028      BEQ      '2'
00029      ENDM
00030      ;
00031      MACRO      GET      ; LOADING BUFFERS
00032      LDX      '1'      ; LOAD TIME FROM PTM
00033      STX      '2'      ; STORE IT IN ECHO BUFFER
00034      COM      '2'      ; GET POSITIVE VALUE
00035      COM      '2'+1
00036      LDA      A      '2'      ; CHECK FOR NO ECHO
00037      CMP      A      #27H      ; IF NO ECHO
00038      BEQ      G1'2'      ; LOAD ZERO
00039      LDA      A      '2'+1      ; CALCULATE BLANK TIME
00040      SUB      A      #FLT      ; BY SUBTRACTING FILTER
00041      STA      A      '3'+1      ; FILTER TIME
00042      LDA      A      '2'      ; MSB CALCULATING
00043      SBC      A      #HI<FLT>
00044      BMI      G1'2'      ; CHECK FOR POLARITY
00045      STA      A      '3'
00046      BRA      G2'2'      ; REGULAR TERMINATION
00047      G1'2' LDX      #0      ; IF BLANK TIME IS NEGATIVE
00048      STX      '3'      ; THEN THE DELAY IS 0
00049      G2'2' NOP
00050      ENDM
00051      ;
00052

```

```

00054      ;*****
00055      ;
00056      ;      MAIN PROGRAM
00057      ;
00058      ;*****
00059      ;
00060      SECTION C1,ABSOLUTE
00061      FE00      ORG      0FE00H      ; SEARCH MODULE STARTS HERE
00062      SRCHM     CHMODE   10H,MODE10  ; CHECK FOR MODE & JUMP.
00063      CHMODE   11H,MODE11
00064      CHMODE   12H,MOD12A
00065      FE15 3B      IRQEND RTI
00066      FE16 7EFE73   MOD12A JMP      MODE12      ; DUMMY JUMP (RANGE PROBLEM)
00067      ;
00068      FE19 CE0000   MODE10 LDX      #0          ; ZERO BLANK TIMERS
00069      FE1C FF0004 >      STX      PTM1+4        ; P101 BLNK IS 0
00070      FE1F FF0004 >      STX      PTM2+4        ; P102 BLNK IS 0
00071      FE22 FF0004 >      STX      PTM3+4        ; P104 BLNK IS 0
00072      FE25 FF0006 >      STX      PTM3+6        ; P103 BLNK IS 0
00073      FE28 FF0004 >      STX      PTM4+4        ; P106 BLNK IS 0
00074      FE2B FF0004 >      STX      TIMER+4       ; P105 BLNK IS 0
00075      FE2E FF0006 >      STX      PTM4+6        ; LOAD TIME IS 0
00076      ;
00077      FE31 BDFF8B      JSR      PTMINT          ; INITIATE COUNTER WITH FFFFH
00078      ;
00079      FE34 86E3      LDA      A      #0E3H      ; CONTROL WORD FOR INIT TIMER
00080      FE36 B70000 >   STA      A      PTM1      ; BIT 7=1 OUTPUT ENABLE
00081      ;                               ; BIT 6=1 INTERRUPT ENABLE
00082      ;                               ; BITS 543=100 FOR SINGLE SHOT
00083      ;                               ; BIT 2=0 16 BIT OPERATION
00084      ;                               ; BIT 1=1 USE E CLOCK
00085      ;                               ; BIT 0=1 DIVIDE BY 8 COUNTER
00086      FE39 CE30D4      LDX      #MS100        ; X IS LOADED WITH 100MS VALUE
00087      FE3C FF0006 >   STX      PTM1+6        ; START INIT PULSE FOR 100MS
00088      FE3F 8611      LDA      A      #11H      ; CHANGE MODE TO 11
00089      FE41 B70000 >   STA      A      MODE
00090      FE44 7EFE15      JMP      IRQEND        ; WAIT 100MS
00091      ;
00092      FE47 B60001 >   MODE11 LDA      A      PTM1+1  ; CHECK FOR 100MS INTERRUPT
00093      FE4A 8484      AND      A      #84H      ; FROM INIT PTM
00094      FE4C 8184      CMP      A      #84H      ; IF NOT GOTO IRQEND
00095      FE4E 26C5      BNE      IRQEND
00096      ;
00097      FE50 8654      LDA      A      #"T"      ; TRANSMIT T
00098      FE52 B70001 >   STA      A      ACIA+1
00099      FE55 BDFE76      JSR      LDPTM          ; LOAD TIMERS DATA
00100      ;
00101      FE58 8612      LDA      A      #12H      ; CHANGE MODE TO 12
00102      FE5A B70000 >   STA      A      MODE
00103      FE5D C60D      LDA      B      #13      ; TRANSMIT 13 CHARACTERS
00104      FE5F F70000 >   STA      B      TXMAX
00105      FE62 7F0000 >   CLR      TXCNT          ; TX COUNTER=0
00106      FE65 CE0000 >   LDX      #ECHOB          ; LOAD ECHO POINTER

```



```

00107 FE68 FF0000 >      STX      TXPONT
00108 FE6B 860D          LDA      A          #0DH      ; TRANSMIT AT THE END
00109 FE6D B7000C >      STA      A          ECHOB+12      ; (CR)
00110 FE70 7EFE15          JMP      IRQEND
00111                      ;
00112 FE73 7E0000 >      MODE12 JMP      TXON          ; TRANSMIT ECHO BUFFER
00113                      ;
00114                      ;*****
00115                      ;
00116                      ;      SUBROUTINES PROGRAM
00117                      ;
00118                      ;*****
00119                      ;
00120          LDPTM GET      PTM1+2,ECHOB,BLANKB      ; P101 LOAD
00121                      GET      PTM2+2,ECHOB+2,BLANKB+2      ; P102 LOAD
00122                      GET      PTM2+6,ECHOB+4,BLANKB+4      ; P103 LOAD
00123                      GET      PTM3+2,ECHOB+6,BLANKB+6      ; P104 LOAD
00124                      GET      TIMER+2,ECHOB+8,BLANKB+8      ; P105 LOAD
00125                      GET      PTM4+2,ECHOB+10,BLANKB+10      ; P106 LOAD
00126 FF8A 39            RTS
00127                      ;
00128 FF8B CEFFFF          PTMINT LDX      #0FFFFH      ; PREPARE COUNTERS OF PTMS
00129 FF8E FF0002 >          STX      PTM1+2      ; P101
00130 FF91 FF0002 >          STX      PTM2+2      ; P102
00131 FF94 FF0006 >          STX      PTM2+6      ; P103
00132 FF97 FF0002 >          STX      PTM3+2      ; P104
00133 FF9A FF0002 >          STX      TIMER+2      ; P105
00134 FF9D FF0002 >          STX      PTM4+2      ; P106
00135 FFA0 39            RTS
00136                      ;
00137                      END

```

Strings and Macros

CHMODE - 004E M GET ---- 01F6 M

Scalars

FLT ---- 0032 G MS100 -- 30D4

C1 Section Absolute (FFA1)

G14000 - FE9D	G15000 - FECB	G16000 - FEF9
G17000 - FF27	G18000 - FF55	
G19000 - FF83	G24000 - FEA3	G25000 - FED1
G26000 - FEF9	G27000 - FF2D	
G28000 - FF5B	G29000 - FF89	IRQEND - FE15
LDPTM -- FE76 G	MOD12A - FE16	
MODE10 - FE19	MODE11 - FE47	MODE12 - FE73
PTMINT - FF8B G	SRCHM -- FE00 G	

ACIA Unbound Global

BLANKB Unbound Global

ECHOB Unbound Global

MODE Unbound Global

PTM1 Unbound Global

PTM2 Unbound Global

PTM3 Unbound Global

PTM4 Unbound Global

TIMER Unbound Global

TXCNT Unbound Global

TXMAX Unbound Global

TXON Unbound Global

TXPONT Unbound Global

137 Source Lines 263 Assembled Lines 13479 Bytes available

>>> No assembly errors detected <<<

```
00001          NOLIST ME
00003          ;
00004          ;*****
00005          ;
00006          ;   SYSTEM CONSTANTS
00007          ;
00008          ;*****
00009          ;
00010          GLOBAL MODE,TRACKM,TIMER,PTMINT
00011          GLOBAL PTM1,PTM2,PTM3,ACIA,PTM4
00012          GLOBAL TXCNT,TXMAX,TXPONT,LDPTM
00013          GLOBAL TXON,ECHOB,BLANKB,FLT
00014          GLOBAL ACIAB,TRCNT
00015          ;
00016          07D0      MS2    EQU    2000          ; 2MS TIME CONSTANT
00017          2710      MS10   EQU    10000         ; 10MS TIME CONSTANT
00018          ;
```

```
00020 ;*****
00021 ;
00022 ; MACRO INSTRUCTIONS
00023 ;
00024 ;*****
00025 ;
00026 MACRO CHMODE ; CHECK IF MODE IS VAR1
00027 LDA A MODE ; IF YES JUMP TO VAR2
00028 CMP A #'1' ; ELSE CONTINUE
00029 BEQ '2'
00030 ENDM
00031 ;
00032 MACRO BLANK ; LOADING BLANK TIMERS
00033 LDX '1' ; LOAD FROM BLANK BUFFER
00034 STX '2' ; STORE IT IN TIMER
00035 ENDM
00036 ;
00037 MACRO LETI ; INSERT THE NUMBER OF SAMPLES
00038 LDA A #'1' ; ACCORDING TO VAR1
00039 CMP A ACIAB+1
00040 BNE G1'2'
00041 LDX #'2'
00042 STX TRCNT ; LOAD TRACK COUNTER
00043 G1'2' NOP
00044 ENDM
00045 ;
00046 MACRO MD
00047 LDA A #'1' ; CHANGE MODE TO VAR1
00048 STA A MODE
00049 ENDM
00050 ;
```

```

00052      ;*****
00053      ;
00054      ;      MAIN PROGRAM
00055      ;
00056      ;*****
00057      ;
00058      SECTION D1,ABSOLUTE
00059      FC00      ORG      0FC00H      ; TRACK MODULE STARTS HERE
00060      TRACKM CHMODE 20H,MODE20      ; CHECK FOR MODE & JUMP.
00061      CHMODE 21H,MODE21
00062      CHMODE 22H,MOD22A
00063      CHMODE 23H,MOD22A
00064      CHMODE 24H,MOD24A
00065      FC23 3B      IRQEND RTI
00066      FC24 7EFC3      MOD22A JMP      MODE22      ; DUMMY JUMP (RANGE PROBLEM)
00067      FC27 7EFC6      MOD24A JMP      MODE24      ; DUMMY JUMP (RANGE PROBLEM)
00068      ;
00069      FC2A BDFCFD      MODE20 JSR      SETBLK      ; PREPARE BLANK TIMERS
00070      FC2D BD0000      >      JSR      PTMINT      ; INITIATE COUNTER WITH FFFFH
00071      ;
00072      FC30 84E2      LDA      A      #0E2H      ; CONTROL WORD FOR INIT TIMER
00073      FC32 B70000      >      STA      A      PTM1      ; BIT 7=1 OUTPUT ENABLE
00074      ; BIT 6=1 INTERRUPT ENABLE
00075      ; BITS 543=100 FOR SINGLE SHOT
00076      ; BIT 2=0 16 BIT OPERATION
00077      ; BIT 1=1 USE E CLOCK
00078      ; BIT 0=0 DIVIDE BY 8 DISABLED
00079      FC35 CE2710      LDX      #MS10      ; X IS LOADED WITH 10MS VALUE
00080      FC38 FF0006      >      STX      PTM1+6      ; START INIT PULSE FOR 100MS
00081      ;
00082      LETI      "0",0      ; IF 0 THEN 1 SAMPLE
00083      LETI      "1",9      ; IF 1 THEN 10 SAMPLES
00084      LETI      "4",2      ; IF 4 THEN 3 SAMPLES
00085      LETI      "2",83      ; IF 2 THEN 1 Sec BUFFER SAMPLES
00086      LETI      "3",999      ; IF 3 THEN 12 SEC SAMPLES
00087      ;
00088      FC81 B0FD22      JSR      TIME2      ; PREPARE 2MSEC TIMER
00089      MD      21H      ; MODE IS 21
00090      FC89 7EFC23      JMP      IRQEND      ; WAIT 10MS
00091      ;
00092      FC8C B60001      >      MODE21 LDA      A      PTM1+1      ; CHECK FOR 10MS INTERRUPT
00093      FC8F 8484      AND      A      #84H      ; FROM INIT PTM
00094      FC91 8184      CMP      A      #84H      ; IF NOT GOTO IRQEND
00095      FC93 268E      BNE      IRQEND
00096      ;
00097      FC95 8454      LDA      A      #"T"      ; TRANSMIT T
00098      FC97 B70001      >      STA      A      ACIA+1
00099      FC9A BD0000      >      JSR      LDPTM      ; LOAD TIMERS DATA
00100      ;
00101      FC9D C60D      LDA      B      #13      ; TRANSMIT 13 CHARACTERS
00102      FC9F F70000      >      STA      B      TXMAX
00103      FCA2 7F0000      >      CLR      TXCNT      ; TX COUNTER=0
00104      FCA5 CE0000      >      LDX      #ECHO8      ; LOAD ECHO POINTER

```

Tektronix M6800 ASM V3.3 TRACK MODULE

Page 5

00158 ;
00159 END

```

00105 FCAB FF0000 > STX TXPONT
00106 FCAB 860D LDA A #0DH ; LOAD CR IN BUFFER
00107 FCAD B7000C > STA A ECHOB+12
00108 FCB0 FE0000 > LDX TRCNT ; CHECK FOR LAST SAMPLE
00109 FCB3 2611 BNE MOD21A ; JUMP IF NOT LAST
00110 ;
00111 MD 22H ; MODE IS 22
00112 FCBA CE0000 LDX #0 ; RESET INIT & LOAD TIMERS
00113 FCBD FF0006 > STX PTM1+6 ; TO PREVENT OSCILATIONS
00114 FCC0 FF0006 > STX PTM4+6
00115 FCC3 7EFC23 JMP IRQEND
00116 MOD21A MD 23H ; MODE IS 23
00117 FCCB 09 DEX
00118 FCCC FF0000 > STX TRCNT ; TRACK COUNTER IS DECREMENTED
00119 FCCF 86F2 LDA A #0F2H ; CONTROL WORD FOR INIT TIMER
00120 FCD1 B70000 > STA A PTM1
00121 FCD4 CE2710 LDX #MS10 ; X IS LOADED WITH 10MS VALUE
00122 FCD7 FF0006 > STX PTM1+6 ; START INIT PULSE FOR 10MS
00123 FCDA BDFCFD JSR SETBLK ; PREPARE BLANK TIMERS
00124 FCDD BD0000 > JSR PTMINT ; INITIATE COUNTER WITH FFFFH
00125 FCE0 7EFC23 JMP IRQEND
00126 ;
00127 FCE3 7E0000 > MODE22 JMP TXON ; TRANSMIT ECHO BUFFER
00128 ;
00129 FCE6 B60001 > MODE24 LDA A PTM4+1 ; CHECK FOR 2MS INTERRUPT
00130 FCE9 8484 AND A #84H ; FROM INIT PTM
00131 FCEB 8184 CMP A #84H ; IF NOT GOTO IRQEND
00132 FCED 2703 BEQ MOD24B ; IF OK START ANOTHER TRACK
00133 FCEF 7EFC23 JMP IRQEND ; WAIT FOR IT
00134 FCF2 BDFD22 MOD24B JSR TIME2 ; PREPARE 2MSEC TIMER
00135 MD 21H ; MODE IS 21
00136 FCFA 7EFC23 JMP IRQEND ; WAIT 10MS
00137 ;
00138 ;
00139 ;*****
00140 ;
00141 ; SUBROUTINES PROGRAM
00142 ;
00143 ;*****
00144 ;
00145 SETBLK BLANK BLANKB,PTM1+4 ; PREPARE P101
00146 BLANK BLANKB+2,PTM2+4 ; PREPARE P102
00147 BLANK BLANKB+4,PTM3+6 ; PREPARE P103
00148 BLANK BLANKB+6,PTM3+4 ; PREPARE P104
00149 BLANK BLANKB+8,TIMER+4 ; PREPARE P105
00150 BLANK BLANKB+10,PTM4+4 ; PREPARE P106
00151 FD21 39 RTS
00152 ;
00153 FD22 86F2 TIME2 LDA A #0F2H ; PREPARE LOAD TIMER
00154 FD24 B70000 > STA A PTM4 ; FOR 2 MSEC
00155 FD27 CE07D0 LDX #MS2
00156 FD2A FF0006 > STX PTM4+6
00157 FD2D 39 RTS

```

Strings and Macros

BLANK -- 0048 M	CHMODE - 004E M	LETI --- 0076 M
MD ----- 0035 M		

Scalars

MS10 --- 2710	MS2 ---- 07D0
---------------	---------------

D1 Section Absolute (FD2E)

G16000 - FC48	G17000 - FC56	G18000 - FC64
G19000 - FC72	G1A000 - FC80	
IRQEND - FC23	MOD21A - FCC6	MOD22A - FC24
MOD24A - FC27	MOD24B - FCF2	
MODE20 - FC2A	MODE21 - FC8C	MODE22 - FCE3
MODE24 - FCE6	SETBLK - FCFD	
TIME2 -- FD22	TRACKM - FC00 G	

ACIA Unbound Global

ACIAB Unbound Global

BLANKB Unbound Global

ECHOB Unbound Global

FLT Unbound Global

LDPTM Unbound Global

MODE Unbound Global

PTM1 Unbound Global

PTM2 Unbound Global

PTM3 Unbound Global

PTM4 Unbound Global

PTMINT Unbound Global

TIMER Unbound Global

TRCNT Unbound Global

TXCNT Unbound Global

TXMAX Unbound Global

TXON Unbound Global

TXPONT Unbound Global

APPENDIX C

MAIN COMPUTER LISTING

CLEAR

```
DIM D(6),rx(2,6),w(3),array(3,2),dist(6),c22(9,2,5),b(2),b1(2)
DIM a(2,5),a1(2,2)
DIM c(3),c1(9),er(6),ermax(6)
DIM ci(6),bi(6)
ci(1)=.341:ci(2)=.3423:ci(3)=.341:ci(4)=.3437:ci(5)=.339:ci(6)=.341
bi(1)=-.002:bi(2)=-.0075:bi(3)=-.002:bi(4)=-.0113:bi(5)=.004:bi(6)=-.002:
```

.

.

STO Module

.

Main Menu

STO:

```
trflag=0
WINDOW 2,"3D Tracking System",(0,40)-(505,330),1
WHILE DIALOG(0)=1:WEND
PRINT "Select by clicking button"
BUTTON 1,1,"Receivers set up",(5,25)-(200,40),2
BUTTON 2,1,"Distance Test",(5,55)-(200,70),2
BUTTON 3,1,"Auto Track",(5,85)-(140,100),2
BUTTON 5,1,"Manual Track",(150,85)-(290,100),2
BUTTON 4,1,"Stability",(5,105)-(200,135),2
CLOSE
WHILE DIALOG(0) <> 1:WEND
'Get window number
btt=DIALOG (1)
IF btt=1 THEN CALL Setup
IF btt=1 THEN CALL Setrx
IF btt=2 THEN GOTO Test
IF btt=3 THEN GOSUB LOCA
IF btt=5 THEN GOSUB Locam
IF btt=4 THEN GOTO Stab
GOTO STO
bi(1)=-.002:bi(2)=-.0075:bi(3)=-.002:bi(4)=-.0113:bi(5)=.004:bi(6)=-.002:
```

.

.

Stab Module

.

Stability Testing

.

Stab:

CLS

WINDOW 2,"Transducer Stability Measurement",(0,40)-(505,330),1

stflag=1

GOTO test1

```

*****
.
.       Test Module
.       Distance Testing
.
*****

```

```

Test:
CLS
WINDOW 2,"3D Tracking System",(0,40)-(505,330),1
stflag=0

```

```

test1:
WHILE DIALOG(0) = 1:WEND
PRINT "Select by clicking button"
BUTTON 1,1,"Error Mode",(5,25)-(200,40),2
BUTTON 2,1,"Search Mode",(5,55)-(200,70),2
BUTTON 3,1,"Track 1 Mode",(5,85)-(200,100),2
BUTTON 4,1,"Track 3 Mode",(5,115)-(200,130),2
BUTTON 5,1,"Track 10 Mode",(5,135)-(200,160),2
BUTTON 6,1,"EXIT",(5,175)-(200,190),2
st:CLOSE
WHILE DIALOG(0) <> 1:WEND
'Get window number
btt=DIALOG (1)
erflag=0:FOR i=1 TO 6:er(i)=0:ermax(i)=0:NEXT i
ON btt GOSUB Er,Sr,Tr1,Tr3,Tr10,Main
GOTO st
Main:GOTO ST0

```

```

*****
.
.       Er Module
.       Communication Testing
.
*****

```

```

Er: Error mode.
OPEN "I",#1,"COM1:19200,N,8,1",256           :REM Prepare input buffer
OPEN "O",#3,"COM1:19200,N,8,1",256           :REM Prepare output buffer
er1:er$="XYZ":il=6
PRINT #3,er$
GOSUB gdata                                   :REM Get error data.
LOCATE 8,37: PRINT x$
RETURN
GOTO st

```

Sr Module
Testing distances in search Mode

```
Sr:Search Mode
  LET b$="S0"+CHR$(10):il=14          :REM Invoke track 0 mode.
  GOSUB TRWIND
  x$=STRING$(14,49): FOR j=1 TO 6:D(j)=1:NEXT j
  WHILE MOUSE(0)=0
    PRINT *3,b$
    L1=0:GOSUB pr1
    GOSUB gdata
  WEND
  WINDOW CLOSE 3
  WINDOW CLOSE 4
RETURN
```

Tr1 Module
Testing distances in Track 1 Mode

```
Tr1:
  LET b$="T0"+CHR$(10):il=14          :REM Invoke track 0 mode.
  GOSUB TRWIND
  x$=STRING$(14,49): FOR j=1 TO 6:D(j)=1:NEXT j
  WHILE MOUSE(0)=0
    PRINT *3,b$
    L1=0:GOSUB pr1
    GOSUB gdata
  WEND
  WINDOW CLOSE 3
  WINDOW CLOSE 4
RETURN
```

Tr3 Module
Testing distances in Track 3 Mode

```
Tr3:
  LET b$="T4"+CHR$(10):il=42          :REM Invoke track 3 mode.
  GOSUB TRWIND
  x$=STRING$(42,49): FOR j=1 TO 6:D(j)=1:NEXT j
  WHILE MOUSE(0)=0
    PRINT *3,b$
    L1=2:GOSUB pr1
    GOSUB gdata
  WEND
  WINDOW CLOSE 3
  WINDOW CLOSE 4
```

RETURN

Tr10 Module
Testing distances in Track 10 Mode

Tr10:

```
LET b$="T1"+CHR$(10):il=140          :REM Invoke track 10 mode.
GOSUB TRWIND
x$=STRING$(140,49): FOR j=1 TO 6:D(j)=1:NEXT j
WHILE MOUSE(0)=0
  PRINT #3,b$
  L1=9:GOSUB pr1
  GOSUB gdata
WEND
WINDOW CLOSE 3
WINDOW CLOSE 4
RETURN
```

Displaying Results

```
pr1: IF stflag=0 THEN pr11
FOR j=1 TO 6
  CN=0:DIS1=0
  FOR i=0 TO L1          :REM Print results.
    x1$=MID$(x$(i*14)+(2*j),2)
    IF LEN(x1$) > 1 THEN x1=CVI(x1$)
    d1=ci(j)*x1/1000+bi(j)
    IF (d1<2 AND d1>.7) THEN CN=CN+1:DIS1=DIS1+d1
  NEXT i
  IF CN>0 THEN DIS=DIS1/CN:D(j)=0
  IF D(j)<2 THEN LOCATE 2+j,7
  IF CN=0 THEN cn23
  IF erflag<4 THEN GOTO er22
  k11=erflag-3
  er1=ABS(DIS-dist(j))
  ermax(j)=(ermax(j)*(k11-1)+er1)/k11
  IF er1>er(j) THEN Er(j)=er1

  IF D(j)=0 THEN PRINT USING "***. * cm.":er(j)*100;:LOCATE 2+j,22: PRINT USING "***. *
cm.":ermax(j)*100;
  er22:dist(j)=DIS
  cn23:IF D(j)=1 THEN PRINT " X.X          X.X";
  IF D(j)<3 THEN D(j)=D(j)+1
NEXT j
erflag=erflag+1
RETURN
```

pr11:

```

FOR j=1 TO 6
  CN=0:DIS1=0
  FOR i=0 TO L1
    x1$=MID$(x$, (i*14)+(2*j),2)
    IF LEN(x1$) > 1 THEN x1=CVI(x1$)
    d1=ci(j)*x1/1000+bi(j)
    IF (d1<2 AND d1>.7) THEN CN=CN+1:DIS1=DIS1+d1
  NEXT i
  IF CN>0 THEN DIS=DIS1/CN:D(j)=0
  IF D(j)<2 THEN LOCATE 1+j,13
  IF D(j)=0 THEN PRINT USING ".**** m.";DIS;
  IF D(j)=1 THEN PRINT "X.XXX";
  IF D(j)<3 THEN D(j)=D(j)+1
NEXT j
RETURN

*****
.
.
.
Definition of the output window
.
*****

TRWIND:
OPEN "I",#1,"COM1:19200,N,8,1",2046 :REM Prepare input buffer
OPEN "O",#3,"COM1:19200,N,8,1",256 :REM Prepare output buffer
IF stflag=1 THEN WINDOW 4,"Stability Data",(150,75)-(445,240),1:GOTO org14
IF btt=2 THEN WINDOW 4,"Searching Data",(250,75)-(445,240),1
IF btt=3 THEN WINDOW 4,"Tracking 1 Data",(250,75)-(445,240),1
IF btt=4 THEN WINDOW 4,"Tracking 3 Data",(250,75)-(445,240),1
IF btt=5 THEN WINDOW 4,"Tracking 10 Data",(250,75)-(445,240),1
IF btt=88 THEN WINDOW 4,"Locating Data",(250,75)-(445,240),1
IF btt=88 THEN org11
LOCATE 2,1
FOR i=1 TO 6
  PRINT "Rx#";i;" Dist.= "
NEXT i
org11:
WINDOW 3,.(335,275)-(380,290),2
PRINT " STOP";
WINDOW OUTPUT 4
RETURN
org14:
LOCATE 1,7
PRINT "Max. devi.      Aver. devi.":PRINT
FOR i=1 TO 6
  PRINT USING "Rx #";i
NEXT i
GOTO org11

```

```

*****
.
.
.
Gdata Module
Receiving data from control processor
.
*****

```

```

gdata:
TIMER ON
ON TIMER (10) GOSUB RET2
x=0

```

```

x$=""
LOOP:
a$=INPUT$(LOC(1),1)
x=x+LEN(a$)
x$=x$+a$
REM PRINT LEN(A$),LEN(x$),x
IF x=1 THEN GOTO RET1

NODATA:
IF LOC(1)>0 THEN LOOP ELSE NODATA
RET2:
WINDOW CLOSE 3
WINDOW CLOSE 4
INPUT WINDOW 2
TIMER OFF
GOTO st
RET1: TIMER OFF
RETURN

```

```

*****
.
.
.      Locam Module
.      Manual Location Module
.
*****

```

```

Locam:
trflag=1
LOCATE 15,6
INPUT "Select Receivers Set (0-9)";setx
IF setx>9 OR setx<0 THEN Locam
wset=INT(setx)
IF wset=0 THEN w(0)=1:w(1)=2:w(2)=3
IF wset=1 THEN w(0)=1:w(1)=2:w(2)=4
IF wset=2 THEN w(0)=1:w(1)=2:w(2)=5
IF wset=3 THEN w(0)=1:w(1)=3:w(2)=4
IF wset=4 THEN w(0)=1:w(1)=3:w(2)=5
IF wset=5 THEN w(0)=1:w(1)=4:w(2)=5
IF wset=6 THEN w(0)=2:w(1)=3:w(2)=4
IF wset=7 THEN w(0)=2:w(1)=3:w(2)=5
IF wset=8 THEN w(0)=2:w(1)=4:w(2)=5
IF wset=9 THEN w(0)=2:w(1)=4:w(2)=5

```

```

*****
.
.
.      LOAC Module
.      Automatic Location Module
.
*****

```

```

LOCA:
WINDOW 2,"3D Tracking System",(0,40)-(505,330),1
OPEN "Rx_data" FOR INPUT AS#4
FOR i=0 TO 2:FOR j=1 TO 5
INPUT #4,rx(i,j)
NEXT j:NEXT i
CLOSE #4
CALL Setrx : 'PREPARE LOCATION MODULE
CLS
btt=88

```

```

LET b$="S0"+CHR$(10):il=14                :REM Invoke track 0 mode.
stflag=0
GOSUB TRWIND
LOCATE 1,6:PRINT "Set="
LOCATE 3,2:PRINT "X=";LOCATE 4,2:PRINT "Y=";LOCATE 5,2:PRINT "Z=";
PRINT *3,b$: GOSUB gdata                    :REM GET DATA Module

```

```

LET b$="T4"+CHR$(10):il=42                :REM Invoke track 3 mode.
WHILE MOUSE(0)=0

```

STABILITY CHECK Module

```

PRINT *3,b$
L1=2
FOR j=1 TO 5
CN=0:DIS1=0
FOR i=0 TO L1                            :REM Print results.
x1$=MID$(x$,i*14)+(2*j),2)
IF LEN(x1$) > 1 THEN x1=CVI(x1$)
d1=ci(j)*x1/1000+bi(j)
IF (d1<2 AND d1>.7) THEN CN=CN+1:DIS1=DIS1+d1
NEXT i
dk=1:D(j)=1
IF CN>0 THEN dis2=DIS1/CN:dk=0
IF (dk=0 AND ABS(dist(j)-dis2)<.0008) THEN D(j)=0 :REM stability for tracking
IF dk=0 THEN dist(j)=dis2
dist(j)=dis2
NEXT j

```

ARRAY SELECTION Module

```

LOCATE 1,1
w=10
IF trflag=1 AND D(w(0))=0 AND D(w(1))=0 AND D(w(2))=0 THEN w=wsel
IF trflag=1 THEN send
' DO NOT USE IF w=10 AND d(1)=0 AND d(2)=0 AND d(3)=0 THEN w(0)=1:w(1)=2:w(2)=3:w=0 '123
IF w=10 AND D(1)=0 AND D(2)=0 AND D(4)=0 THEN w(0)=1:w(1)=2:w(2)=4:w=1 '124
IF w<10 THEN CALL org:w=10
IF w=10 AND D(1)=0 AND D(2)=0 AND D(5)=0 THEN w(0)=1:w(1)=2:w(2)=5:w=2 '125
IF w=10 AND D(1)=0 AND D(3)=0 AND D(4)=0 THEN w(0)=1:w(1)=3:w(2)=4:w=3 '134
IF w=10 AND D(1)=0 AND D(3)=0 AND D(5)=0 THEN w(0)=1:w(1)=3:w(2)=5:w=4 '135
' DO NOT USE IF w=10 AND d(1)=0 AND d(4)=0 AND d(5)=0 THEN w(0)=1:w(1)=4:w(2)=5:w=5 '145
IF w=10 AND D(2)=0 AND D(3)=0 AND D(4)=0 THEN w(0)=2:w(1)=3:w(2)=4:w=6 '234
IF w=10 AND D(2)=0 AND D(3)=0 AND D(5)=0 THEN w(0)=2:w(1)=3:w(2)=5:w=7 '235
IF w=10 AND D(2)=0 AND D(4)=0 AND D(5)=0 THEN w(0)=2:w(1)=4:w(2)=5:w=8 '245
IF w=10 AND D(3)=0 AND D(4)=0 AND D(5)=0 THEN w(0)=3:w(1)=4:w(2)=5:w=9 '345
send:
IF w<10 THEN CALL Org
IF w=10 THEN PRINT "no ",0
GOSUB gdata :REM GET DATA Module
WEND
WINDOW CLOSE 3
WINDOW CLOSE 4

```

RETURN
END

Setup Module
Updates the receivers position

SUB Setup STATIC

Defines x,y,z locations of receivers

rx(0,1)=0: rx(1,1)=0: rx(2,1)=.12

GOTO Ww

SHARED rx(),c22()

OPEN "Rx_data" FOR INPUT AS #4

FOR i=0 TO 2:FOR j=1 TO 5

INPUT #4,rx(i,j)

NEXT j:NEXT i

CLOSE #4

CLS

Ww Module
Waiting window for the receivers
position entry.

Ww:

WINDOW 2,"3D Receivers Location",(0,40)-(505,330),1

WHILE DIALOG(0) =1:WEND

PRINT "Select by clicking button"

FOR i=1 TO 5

x\$=CHR\$(48+i)

FOR j=1 TO 3

y\$=CHR\$(67+j)

z\$="Rx"+x\$+" "+y\$+" ="

BUTTON i+(j-1)*5,1,z\$(5+(j-1)*151,32+(i-1)*32)-(70+(j-1)*151,47+(i-1)*32),3

NEXT j

NEXT i

FOR i=1 TO 5

FOR j=0 TO 2

LOCATE 1+2*i,10+19*j

PRINT USING ".*** m";rx(j,i);

NEXT j

NEXT i

BUTTON 16,1,"EXIT",(350,210)-(395,240),1

STup:CLOSE

WHILE DIALOG(0) <>1:WEND

'Get window number

btt=DIALOG (1)

IF btt<16 THEN GOSUB Up1

IF btt=16 THEN GOTO Up2

GOTO STup

Up1 Module
Updates the new location of
the receiver position

Up1:

LOCATE 15,8
y\$="X"
IF btl>5 THEN y\$="Y"
IF btl>10 THEN y\$="Z"
z=btl MOD 5:IF z=0 THEN z=5
PRINT USING "Enter new "+y\$+" position of Rx*:";z
LOCATE 16,8

INPUT newd
IF (newd<-1 OR newd>1) THEN GOTO Up1
FOR i=0 TO 1: LOCATE 15+i,8:PRINT SPACE\$(37);NEXT i
LOCATE 1+2*z,INT(btl/5.001)*19+10
PRINT USING "### m";newd;
rx(INT(btl/5.001),z)=newd
RETURN

Up2 Module
Stores the location of the
receivers on a disk

Up2:

OPEN "Rx_data" FOR OUTPUT AS#4
FOR i=0 TO 2:FOR j=1 TO 5
WRITE #4, rx(i,j)
NEXT j:NEXT i
CLOSE #4
WINDOW CLOSE 2

END SUB

Setrx Module
 PREPARE LOCATION Module
 Calculates the transformation constants

SUB Setrx STATIC
 SHARED rx(),c22(),a(),a1(),c(),c1()
 REM define new coordinates
 pi=3.141593

DATA 123,124,125,134,135,145,234,235,245,345
 RESTORE
 FOR c=0 TO 9
 READ c11
 c(0)=INT(c11/100):c(1)=INT(c11/10) MOD 10:c(2)=c11 MOD 10

REM Move origin to c(0)
 FOR x=0 TO 2:k=rx(x,c(0)):a(x,4)=k:FOR y=0 TO 2
 a(y,x)=rx(x,c(y))-k
 NEXT y:NEXT x

REM Rotate x,y plane that Rx2 axis will be on x,0,z
 IF a(1,0)<>0 THEN teta=ATN(a(1,1)/a(1,0)) ELSE teta=pi/2
 a(0,3)=SIN(-teta):a(0,5)=COS(-teta)
 FOR i=1 TO 2:FOR j=0 TO 1:a1(i,j)=a(i,j):NEXT j:NEXT i
 a(1,0)=a1(1,0)*COS(teta)+a1(1,1)*SIN(teta)
 a(2,0)=a1(2,0)*COS(teta)+a1(2,1)*SIN(teta)
 a(1,1)=a1(1,1)*COS(teta)-a1(1,0)*SIN(teta)
 a(2,1)=a1(2,1)*COS(teta)-a1(2,0)*SIN(teta)

REM Rotate x,z plane that Rx2 axis will be on x,0,0
 IF a(1,0)<>0 THEN teta=ATN(a(1,2)/a(1,0)) ELSE teta=pi/2
 a(1,3)=SIN(-teta):a(1,5)=COS(-teta)
 FOR i=1 TO 2:FOR j=0 TO 2:a1(i,j)=a(i,j):NEXT j:NEXT i
 a(1,0)=a1(1,0)*COS(teta)+a1(1,2)*SIN(teta)
 a(2,0)=a1(2,0)*COS(teta)+a1(2,2)*SIN(teta)
 a(1,2)=a1(1,2)*COS(teta)-a1(1,0)*SIN(teta)
 a(2,2)=a1(2,2)*COS(teta)-a1(2,0)*SIN(teta)

REM Rotate around x axis that Rx3 will be on x,y,0
 IF a(2,1)<>0 THEN teta=ATN(a(2,2)/a(2,1)) ELSE teta=pi/2
 a(2,3)=SIN(-teta):a(2,5)=COS(-teta)
 FOR i=2 TO 2:FOR j=1 TO 2:a1(i,j)=a(i,j):NEXT j:NEXT i
 a(2,1)=a1(2,1)*COS(teta)+a1(2,2)*SIN(teta)
 a(2,2)=a1(2,2)*COS(teta)-a1(2,1)*SIN(teta)

FOR i=0 TO 2:FOR j=0 TO 5:c22(c,i,j)=a(i,j):NEXT j: NEXT i

NEXT c

END SUB

```
*****
.
.      Getback Module
.      Calculates the reverse transformations
.
*****
```

SUB Getback STATIC

SHARED c22(),b(),config,b1()
REM GOING BACKWARD

PRINT USING "z=*.***";b(2)

FOR i=0 TO 2:b1(i)=b(i):NEXT i
b(1)=b1(1)*c22(config,2,5)+b1(2)*c22(config,2,3)
b(2)=b1(2)*c22(config,2,5)-b1(1)*c22(config,2,3)

FOR i=0 TO 2:b1(i)=b(i):NEXT i
b(0)=b1(0)*c22(config,1,5)+b1(2)*c22(config,1,3)
b(2)=b1(2)*c22(config,1,5)-b1(0)*c22(config,1,3)

FOR i=0 TO 2:b1(i)=b(i):NEXT i
b(0)=b1(0)*c22(config,0,5)+b1(1)*c22(config,0,3)
b(1)=b1(1)*c22(config,0,5)-b1(0)*c22(config,0,3)

FOR i=0 TO 2
b(i)=b(i)+c22(config,i,4)

NEXT i

END SUB

```
*****
.
.      Org Module
.      Calculates the position of the target
.
*****
```

SUB org STATIC

SHARED rx(),dist(),w,b(),c22(),w(),b(),b1(),config
config=w

FOR i=0 TO 2

b(i)=dist(w(i))

NEXT i

d12=b(0)²:d22=b(1)²:d32=b(2)²

x2=c22(w,1,0):x3=c22(w,2,0):y3=c22(w,2,1)

xs=(d12+(x2²-d22)/(2*x2)

ys=(d12+(x3²)+(y3²-d32-(2*xs*x3))/(2*y3)

zs=SQR(ABS(d12-(xs²)-(ys²)))

b(0)=xs:b(1)=ys:b(2)=zs

LOCATE 1,9:PRINT w;

CALL Getback

FOR i=0 TO 2

LOCATE 3+i,4

PRINT USING "*.*** m";b(i);

NEXT i

END SUB