(1)

A Final Report Prepared For

U.S. Army Missile Command

under

Contract DAAH01-87-D-0182

ANALYSIS AND SOFTWARE FOR THE DEVELOPMENT OF AN INERTIALLY AIDED ROBOT MANIPULATOR

James C Hung
Consultant
Purchase Order 701329C01
Computer Science Corporation
Huntsville, AL 35805

31 January 1989

ACKNOWLEDGMENT

This task was supported by U. S. Army Missile Command (MICOM), Redstone Arsenal, Alabama, under U. A. Army Contract DAAH01-87-D-0182 which was contracted to the Computer Science Corporation (CSC), Huntsville, Alabama. The task period was from August 16 to December 31, 1988. The task was performed in conjunction with the work performed by MICOM personnel. Technical assistance from Troy Hester and Chris Lofts of MICOM and Jim Gray and Phil Jayco are gratefully acknowledged.

TABLE OF CONTENTS

Statement A per telecon Martin Harris
Army Missile Command
ATTN: AMSMI-RD-SE-MT
Redstone Arsenal, AL 35898
NWW 3/10/92

TECHNICAL SUPPORT FOR THE DEVELOPMENT OF AN INERTIALLY AIDED ROBOT MANIPULATOR CONTROL

I. INTRODUCTION

This report documents the result of a technical support task performed for the U. S. Army Missile Command (MICOM).

The Task

U. S. Army Missile Command (MICOM) is interested in the use of inertial sensors for the guidance and control of robot manipulators. A project has been launched to conduct a preliminary investigation of a simplified inertially aided robot manipulator control system. This task was to provide technical support for the project, including dynamic and error analysis, modeling, data reduction, control scheme establishment, algorithm development, and performance evaluation.

Background

Accurate control of end-effector position and orientation is important for many robotic manipulator applications. Since the end-effector is connected to the base of the manipulator through its arms, its control is accomplished by controlling the rotation of each arm joint. In current manipulator control systems, each manipulator joint is controlled by a local joint servo. Angular position sensors are installed at joints to measeare joint angles. For a desired end-effector position and orientation, inverse kinematics is used to generate command signals in joint coordinates. These signals become inputs to local joint servos. [1-4] Such control concept may be called joint sensor based manipulator control.

Although manipulators are already widely used for various industrial applications, active researches in this area are still being conducted at government, industrial, and academic institutions. These researches are aiming at extending manipulators' capabilities, improving their motion accuracy, increasing their robustness, enhancing their learning and tracking ability, reducing their sizes and weights, and making them more cost effective and easier to maintain. Two of the most important problems currently under intensive research in several institutions are 1) how to cope with the problem of compliance effects of manipulators, and 2) how to improve the robustness of manipulator control.

The compliance is caused by the physical structure of a manipulator and by the softness of its joint servos. One effect of the compliance is the bending of manipulator arms caused by loading and by their own weights. This affects the accuracy of positioning the end-effector. Another serious effect of the compliance is the existence of bending modes in the manipulator dynamical characteristics which make an accurate and steady control of the end-effector difficult. Current method to cope with the problems of compliance is to adopt large sizes for arm cross-sections, resulting in a bulky manipulator.

The lack of robustness in present manipulators is due to their control scheme. For the joint sensor based manipulator control, although local feedback loops are used at joints, the control of the end-effector (or hand) is open-loop as far as the actual hand position is concerned. There is no mechanism to take care of loading effect and the effect of changes in arm parameters. In fact, the actual state (including position and orientation) of the end-effector is not sensed.

Arrangement has been used to make the control closed-loop by monitoring the position of the end-effector using cameras. Image processing techniques are used to process the three dimensional camera pictures and derive from them the manipulator command signals. This arrangement can solve the open-loop problem only partially. First of all, it requires a good deal of computation effort, thus reduces the bandwidth of the measured data. As a result, the data are not useful for bending mode control. Secondly, the determination of the position and orientation of the end-effector from camera pictures can not always be done to a desired accuracy, resulting in poor end-effector control. Thirdly, there are situations where uses of camera are not feasible. Thus, closing the end-effector loop by camera does not seem to be an effective approach for the improvement of system robustness.

A New Robot Manipulator Control Concept [5]

A new concept has been conceived for effectively closing the end-effector loop by the use of an inertial measurement system (IMS). This concept will be called inertial measurement (IM) based manipulator control. An IMS consists of two parts, namely, an inertial measurement unit (IMU) and a microcomputer. The IMU is a sensing unit which senses both the linear and angular motions of it body. By installing an IMU at the end-effector of a manipulator, the position and orientation of the end-effector can be determined. The microcomputer may either be one dedicated to the IMS or one shared with other subsystems of the manipulator. The information from the output of IMS is used to control the joints of the manipuulator. In this approach, each joint sensor becomes a joint actuator subsystem. The precision requirement of each joint sensor can greatly be reduced since the error of the state of end-effector is sensed by IMS and can be made independent of the errors of joint sensors.

The IM based concept differs markedly from the joint sensor based concept in that the former employs a total system feedback while the latter employs local feedback. The new concept has three distinct advantages:

1. Capable of coping with active and passive compliance effects of robot manipulator.

2. Capable of increasing the robustness of robot manipulator.

3. Easy to be reinitialized.

In terms of applications, these advantages offer many attractive features not availbale from the joint sensor based control. These features include:

1. Capability of dealing with both static and dynamic elasticities:

   - Improved robustness with respect to manipulator loading.

   - Capability of supporting the control of bending mode.

   - Simpler robust implementation of learning and repeating procedures.

2. Stabilization of the end-effector of a manipulator on a moving platform.

   - A side benefit: capable of supporting overall navigation of a mobile robot due to the nature of an IMU.

3. Relaxing the need for highly complex analytical model of the manipulator and be capable of coping with the problem of actuator saturation.

   - Enabling the use of simpler and faster algorithms for precision end-effector control.

   - Enabling stiffer control of the end-effector.

4. Reduction of the accuracy requirements of joint sensors and the performance requirements of joint actuators.

5. Capability of supporting frequent automatic reinitialization of the IMU.

6. Enabling the use of lighter arms thus reduces the bulk and weight of the manipulator.

It is anticipated that a successful development of the proposed scheme will have significant impact on robotic technology.

Inertial Sensor Consideration

Recent progresses in small size inertial sensors have made the IM based control concept potentialy realizable. There are two types of IMU's, one is the gimbaled type and the other is the strapdown type. A strapdown IMU is in general smaller in size and lighter in weight as compared to a gimbaled one. For manipulator control, the physical size of the IMU's must be small, therefore strapdown IMU's are better candidates for such application. [6,7] However, currently, there is no on-the-shelf IMS suitable for manipulator control, but the technology for the desired IMS is available.

Inertial sensors have their imperfections; mainly, gyros suffer from drifts and accelerometers suffer from biases. Each of these imperfections consists of two parts, namely, the known imperfection and the uncertainty. The known imperfection is obtained by calibration while the uncertainty is the result of calibration error and the time-varying shift of the parameter value. The known sensor imperfections can be compensated by software, and, therefore, will not affect IMU accuracy. Uncertain sensor imperfections contribute to IMU errors. These uncertainties are often slowly time-varying quantities which may become excessive over a long period of time. The effect of these uncertainties can be minimized by periodic re-initialization of the IMU.

The Present Project

As an initial phase of the development, MICOM chose to investigate a single axis inertially aided robot manipulator position control rather than a three axis control. This greatly simplified the complexity of the inertial sensor required. Instead of a three axis IMU, only a single accelerometer was required. The system specification consisted of:

Maximum acceleration mangitude:	+&- 2g
Position accuracy desired:	100 micron or .004 in
Acceleration resolution:	1 micro-g (μg)
Control sampling rate	100 Hz

Task Activities

The task activities included the following:

- Analysis of the effect of accelerometer errors on the inertially aided robot control.

- Analysis of accelerometer characteristics, development of a least-square software for recalibrating scale factors and biases.

- Analysis of ADC characteristics and development of an algorithm for compensating the error due to non-zero clearing time of ADC's integrator.

- Develpment of the overall system scheme and the needed system software.

- Analysis and data reduction of experiment results.

II. EFFECT OF ACCELEROMETER ERRORS

An inertially aided single-axis manipulator control requires that the accelerometer, used as motion sensor, be very accurate and stable. Bias and scale factor uncertainties of the accelerometer can hinder a successful development of the desired control.

Bias Effect

Let B be the bias uncertainty of the accelerometer. The resulting position error is

$$X = \frac{1}{2} B t^2 \qquad (1)$$

For B expressed in number of micro-g's (µg), where g = 9.8 meters per second is the gravitational acceleration, the position error in microns (u) is given by

$$\Delta X = \frac{1}{2} \times B \times 10^{-6} \times 9.8 \times 10^{6} \times t^2 = 4.9 B t^2 \quad (\mu) \qquad (2)$$

where t is in seconds. The following table shows values of position error in microns for different values of B and t.

Position error		t in sec				
		1	10	60	120	600
B in µg	1	4.9 µ	490 µ	17.6 mm	7.056 cm	1.764 m
	10	49 µ	4.9 mm	17.6 cm	70.56 cm	17.64 m
	30	147 µ	1.47 cm	52.8 cm	2.1168 m	52.8 m
	100	490 µ	4.9 cm	1.76 m	7.056 m	176.4 m

One sees that, for a high quality accelerometer having a bias specified at 10 micro-g's, the position error is about 4,900 microns (4.9 mm) in 10 seconds and 176,400 microns (176.4 mm) in one minute. These errors are too high for precision manipulator applications.

The specification of an accelerometer bias is usually for day-to-day stability. For the present application, only the stability within several minutes (very short term) is needed. One therefore hopes that, in additional to knowing the bias accurately, the accelerometer has good very-short term bias stability.

Scale Factor Uncertainty Effect

The effect of scale factor uncertainty on position error is given by

$$\Delta X = \Delta S \, (a - g_p) \qquad (3)$$

where ΔS is the scale factor uncertainty, a is motional acceleration, and g_p is the component of gravitational acceleration along the input axis of the accelerometer. The maximum value of g_p is g, occurred when the input axis is vertical; and the minimum value is 0, occurred when the input axis is horizontal. Therefore, when the input axis is horizontal, the effect of scale factor uncertainty on position error is

$$\Delta X = \Delta S \times D \qquad (4)$$

where D is the distance traveled. On the other hand, when the input axis is vertical, the effect of scale factor on position error is given by

$$\Delta X = \Delta S \times D + \Delta S \times g \times t^2 \qquad (5)$$

As an example, let ΔS = 100 ppm (parts per million), the input axis be in the vertical orientation, and the accelerometer be traveled one meter distance in 2 seconds. Then the position error in microns due to scale factor uncertainty is

$$\Delta X = 100 \times 10^{-6} \times 10^{6} + \tfrac{1}{2} \times 100 \times 10^{-6} \times 9.8 \times 10^{6} \times 2^{2} = 100 + 1960 \quad (\mu)$$

Note that the component of the position error due to gravitational acceleration is far greater than that due to traveling. To have a combined position error not more than 100 microns, the scale factor uncertainty must be less than 4.85 ppm, a stringent requirement.

The above example showed that, in additional to knowing the scale factor accurately, its stability within a day must be good, say, within a few ppm.

III. ACCELEROMETER CHARACTERISTICS STUDY

Q-Flex Accelerometers

A group of Q-Flex QA2000 accelerometers were chosen for this project. They are of inertial grade, meaning they are classified as good quality accelerometers. Each accelerometer has a buit-in temperature sensor whose output current value is used for compensating the measured acceleration signal for temperature effect. The compensation is done by computing the scale factor, S, and bias, B, of each accelerometer using [8]

$$S = c_0 + c_1 u + c_2 u^2 + c_3 u^3 + c_4 u^4 \qquad (mA/g) \tag{6}$$

$$B = b_0 + b_1 u + b_2 u^2 + b_3 u^3 + b_4 u^4 \qquad (mg) \tag{7}$$

where c's and b's are tempeature coefficients and u, in microamperes (μA), is given by

$$u = I_{TS} - 293 \qquad (\mu A) \tag{8}$$

where I_{TS} is the temperature sensor current in μA. Each accelerometer has its own set of temperature coefficients furnished by the manufacturer. Since values of these coefficients might change with time and since the accelerometers were manufactured ten years ago, it was necessary to recalibrate them.

Figure 1 is a block diagram of Q-Flex QA2000 series accelerometer. In the figure, terminal 1 is the output for sensed acceleration, terminal 6 is the output for sensed temperature, and terminal 8 is the common return. Thus the accelerometer has two output ports for usual operation.

Tests were performed by MICOM personnel to recalibrate the accelerometers and to reveal their noise and time varying characteristics. These tests involved placing accelerometers in a temperature oven and unde- different input conditions ranging from +1 g to -1 g.

Parameter Stability

Table 1 shows a typical set of test data. In the table, Note that at time instants 8:00 and 8:10 the environment temperatures were the same. The current scale factors (SFI) for the three accelerometers differed by as much as 160 ppm (S/N 104) while the biases differed by as much as 90 μg (S/N 104). Note that the two time instants were only 10 minutes apart, indicating that the parameter stability of these accelerometer is probably too poor

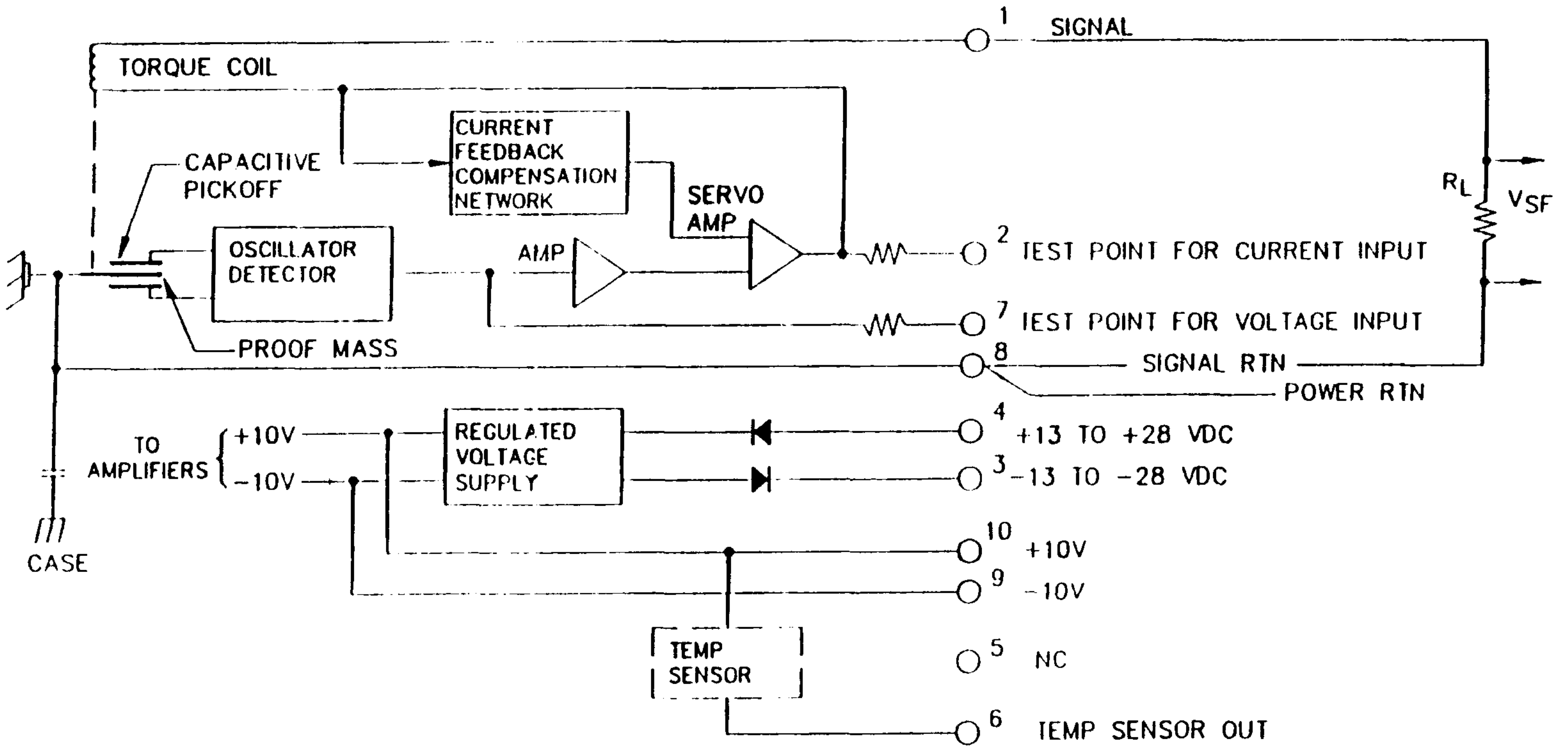


Figure 1. Block diagram of Q-Flex QA2000 accelerometer.

Table 1. Accelerometer Test Data

Three Q-Flex QA2000 accelerometers
Parts number 979-2020-001
Serial numbers 102, 103, & 104

Time	Temp. deg C	S/N	VTS V	SFV V/g	SFI mA/g	Bias mg
7:40	21.5	102	2.9529	4.9144	1.2337	1.1293
		103	2.9533	4.9952	1.2540	.27024
		104	2.9560	4.9511	1.2430	.86849
7:51	21.7	102	2.9546	4.9336	1.2386	-2.72620
		103	2.9565	4.9952	1.2540	.28027
		104	2.9568	4.9513	1.2430	.85837
8:00	21.9	102	2.9531	4.9143	1.2337	1.05810
		103	2.9578	4.9954	1.2541	.20018
		104	2.9579	4.9510	1.2429	.80792
8:10	21.9	102	2.9553	4.9144	1.2338	1.07850
		103	2.9597	4.9956	1.2541	.19017
		104	2.9594	4.9515	1.2431	.82803
8:20	22.1	102	2.9560	4.9146	1.2338	1.09880
		103	2.9599	4.9955	1.2541	.21019
		104	2.9616	4.9517	1.2431	.81791
8:30	22.2	102	2.9577	4.9149	1.2339	1.08850
		103	2.9613	4.9956	1.2541	.24021
		104	2.9625	4.9516	1.2431	.81793

S/N - Serial number
VTS - Temperature sensor output voltage
SFV - Scale factor voltage
SFI - Scale factor current

Data from a test performed by Chris Loft, MICOM, September 1988.

to meet the given control specification.

Determination of Temperature Coefficients

It was found by testing that the original set of temperature coefficients for each accelerometer were not able to make correct compensation for scale factor and bias. Therefore test was done to generate sufficient data for redetermination of these coefficients. A least-square based software was developed to extract temperature coefficients from test data for three accelerometers (Q-Flex P/N 979-2020-001: S/N 102, 103, and 104). Result of the data reduction is shown in Table 2 (a), (b), and (c). In the table are shown the test data, the new and old temperaturte coefficients.

Comparing the new and old values for the three accelerometers, one sees that they have changed appreciably. For examples, the coefficient b_0 of unit S/N 102, which is the dominant term of the bias equation, changed by as much as 190 μg; c_0 of unit S/N 104, the dominant term in the scale factor equation, changed by as much as 1546 ppm; and the coefficient c_4 of S/N 102 changed by 400%.

Software for extracting temperature coefficients is given in Appendix A.

Noise Characteristics

The output of each accelerometer was analyzed for its frequency content using a electronic frequency analyzer. Result of analysis for a typical accelerometer is shown in Figure 2. One sees that the accelerometer output is quite noisy. Figure 2(a) is the frequency spectrum when the input axis of the accelerometer was in level position, and Figure 2 (b) is the spectrum when the input axis was in vertical position. Comparing (a) and (b) reveals that the noise level increased with increasing input. Furthermore, there are strong noise peaks at frequencies around 5, 17, and 25 KHz as shown in the figure. There was also a noise peak at 2 Mhz not shown in the figure. The maximum noise peak is 46 mV corresponding to noise acceleration of 9000 μg.

Since the sampling rate of digital control is specified to be 100 Hz, the folding frequency is 50 Hz. Therefore noise above 50 Hz must be greatly eliminated before sampling to avoid aliasing effect. An anti-aliasing filter with a cutoff frequency at about 27 Hz was used to suppress the high frequency noise. Figure 3 shows the frequency spectra of the accelerometer output with and without the filter. Although the filter helped a great deal to remove the noise, it introduced an output fluctuation at about 1/2 to 1 Hz with a peak-to-peak value of about 20 μg. Removing the filter eliminated the fluctuation. It appeared that the fluctuation was caused by the filter. Two different filter implementations were tried, both gave the same result. The exact

Table 2. Determination of Accelerometer Temperature Coefficients

(a) Unit S/N 102

Q-Flex P/N 979-2020-001, S/N 102

Raw data from test:

Temp. deg C	SFV v/g	Bias mg	Temp.sensor v
-5.500000	4.904720	1.714670	2.691270
-31.700000	4.896840	2.188150	2.451860
-46.200000	4.892600	2.433270	2.304290
-29.200000	4.897440	2.185840	2.468280
-5.200000	4.905290	1.825580	2.696440
21.100000	4.915890	1.403610	2.950940
46.600000	4.928070	0.976040	3.201360
69.900000	4.941240	0.649630	3.439650
94.100000	4.954060	0.314890	3.655820
67.400000	4.939380	0.623560	3.412830
45.700000	4.927430	1.022580	3.200180
21.800000	4.916000	1.360860	2.954040

New coefficients:

B 0	=	1.412769	C 0	=	1.23357
B 1	=	-1.571002E-02	C 1	=	1.116125E-04
B 2	=	4.109868E-06	C 2	=	3.591995E-07
B 3	=	-4.837625E-08	C 3	=	-1.808189E-10
B 4	=	1.306034E-09	C 4	=	-2.16005E-12

Old coefficients:

B 0	=	1.603	C 0	=	1.233831
B 1	=	-.01538	C 1	=	.0001458
B 2	=	.0000062	C 2	=	3.698E-07
B 3	=	-9.9E-08	C 3	=	-3.35E-10
B 4	=	-1.5E-09	C 4	=	6.9E-13

Table 2. Determination of Accelerometer Temperature Coefficients (Continuation)

(b) Unit S/N 103

Q-Flex P/N 979-2020-001, S/N 103

Raw data from test:

Temp. deg C	SFV v/g	BIAS MG	TEMP.SENSOR V
-5.500000	4.986330	0.795170	2.695410
-31.700000	4.978620	1.101710	2.458600
-46.200000	4.974760	1.277450	2.306320
-29.200000	4.978980	1.120710	2.466560
-5.200000	4.986640	0.860300	2.702240
21.100000	4.996710	0.575380	2.953070
46.600000	5.008810	0.302470	3.207460
69.900000	5.021440	0.023900	3.450060
94.100000	5.033860	-0.229450	3.665530
67.400000	5.019650	0.041840	3.421270
45.700000	5.008160	0.283540	3.199530
21.800000	4.997100	0.539310	2.963810

New coefficients:

B 0	=	.584242	C 0	=	1.253878
B 1	=	-1.080819E-02	C 1	=	1.080253E-04
B 2	=	4.375878E-06	C 2	=	3.280584E-07
B 3	=	-6.731949E-08	C 3	=	-6.360779E-10
B 4	=	-5.861747E-10	C 4	=	3.467449E-12

Old coefficients:

B 0	=	.686	C 0	=	1.255471
B 1	=	-.01123	C 1	=	1.0766E-04
B 2	=	-.0000093	C 2	=	3.634E-07
B 3	=	-4.4E-08	C 3	=	-4.12E-10
B 4	=	1.77E-09	C 4	=	-7.2E-12

Table 2. Determination of Accelerometer Temperature Coefficients (Continuation)

(c) Unit S/N 104

Q-Flex P/N 979-2020-001, S/N 104

Raw data from test:

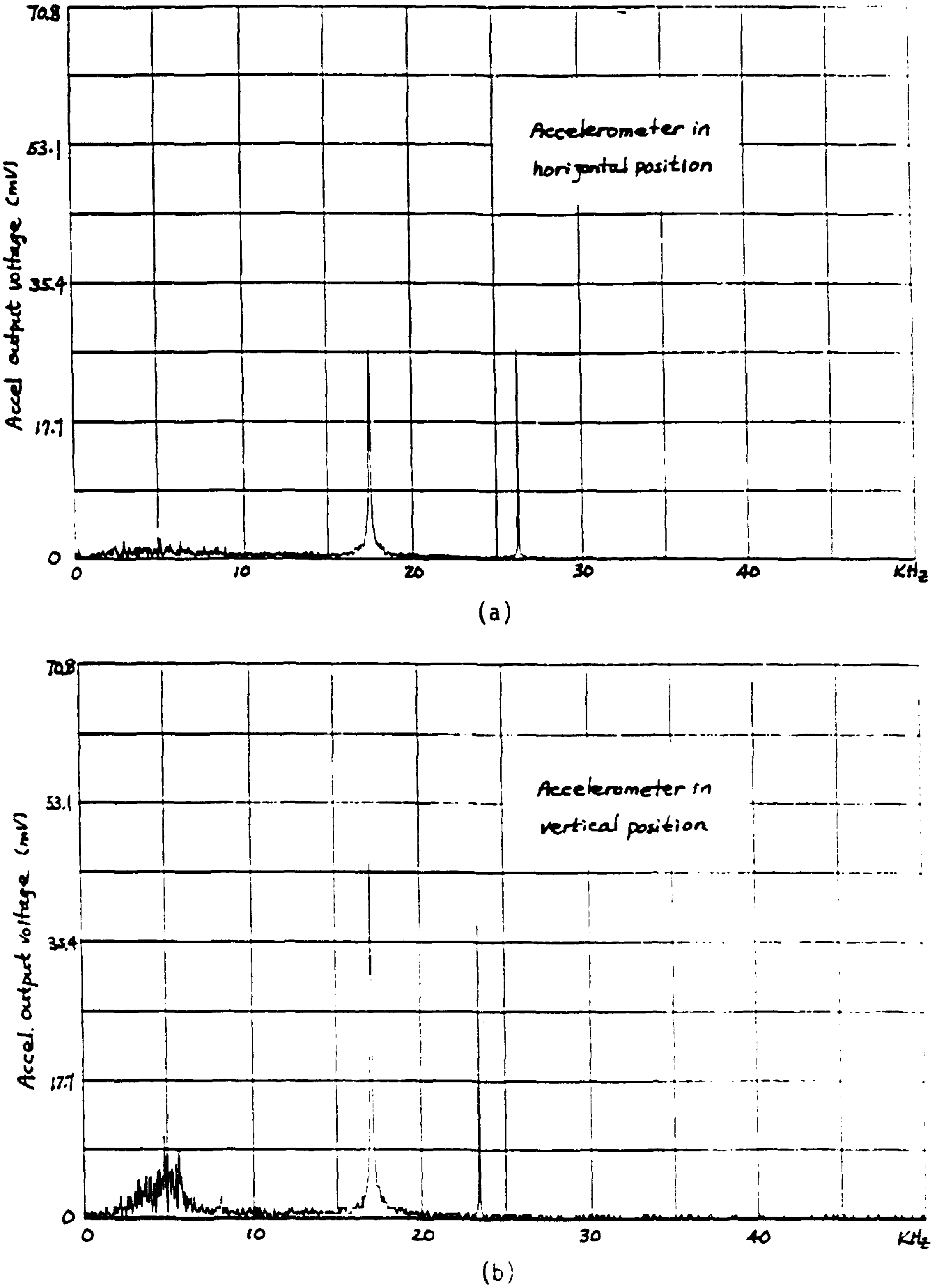
Temp. deg C	SFV v/g	Bias mg	Temp.sensor v
-5.500000	4.942160	1.621760	2.699260
-31.700000	4.934950	1.981780	2.459080
-46.200000	4.931400	2.246830	2.315340
-29.200000	4.935190	1.972570	2.474880
-5.200000	4.942170	1.560040	2.700790
21.100000	4.952270	1.117670	2.956220
46.600000	4.964360	0.732220	3.208400
69.900000	4.977030	0.264210	3.441820
94.100000	4.989680	-0.128260	3.659710
67.400000	4.975760	0.315530	3.415230
45.700000	4.964270	0.714100	3.200570
21.800000	4.953230	1.139660	2.964340

New coefficients:

B 0	= 1.193782	C 0	= 1.242741
B 1	= -1.723073E-02	C 1	= 1.088916E-04
B 2	= -1.388611E-05	C 2	= 4.002359E-07
B 3	= -9.966629E-08	C 3	= -9.820269E-10
B 4	= 1.580361E-09	C 4	= -2.366107E-12

Old coefficients:

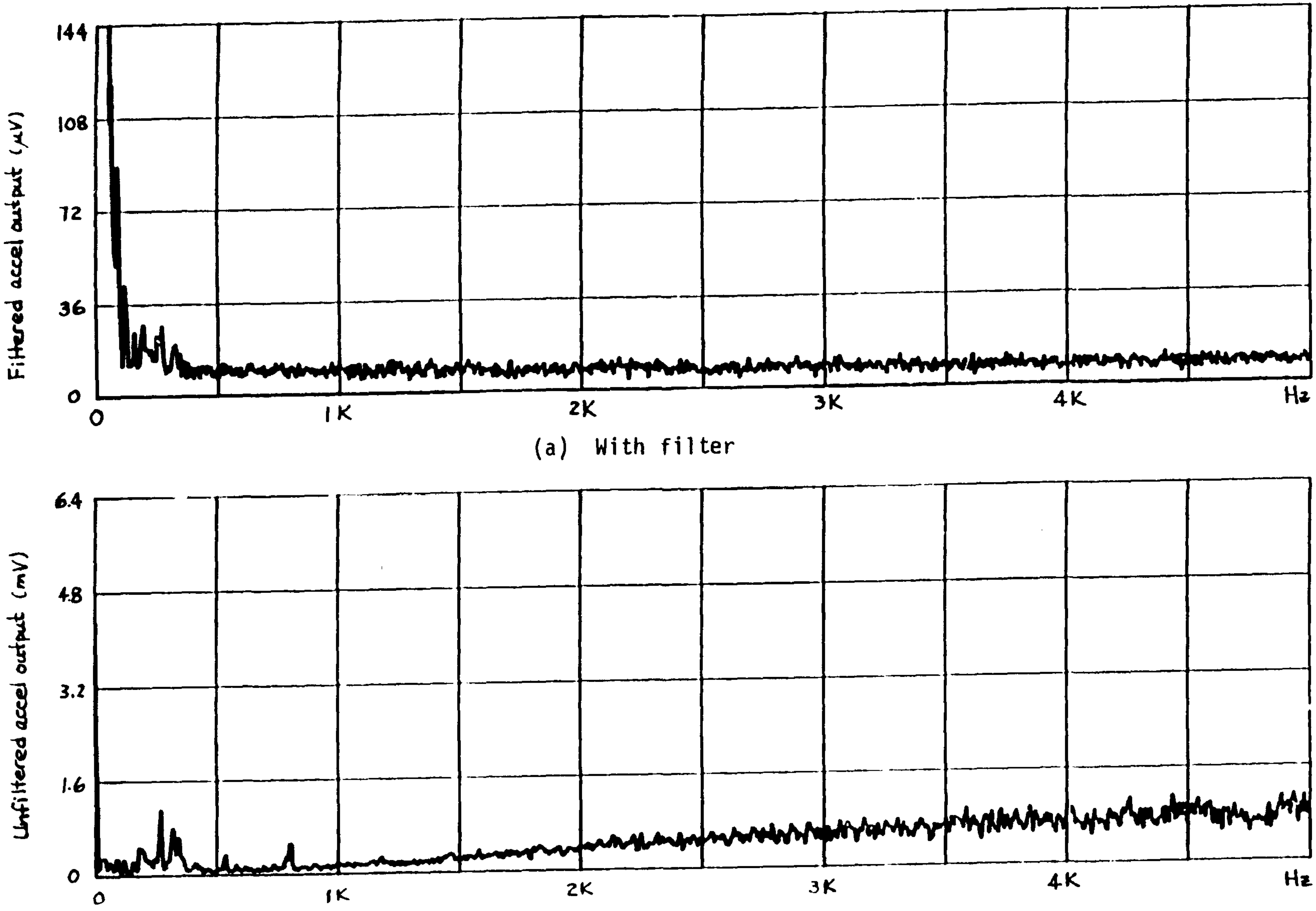
B 0	= 1.247	C 0	= 1.244662
B 1	= -.01725	C 1	= .000103
B 2	= -.0000027	C 2	= 3.961E-07
B 3	= -1.64E-07	C 3	= -4.96E-10
B 4	= -2.9E-10	C 4	= 1.32E-12



(a)

(b)

Figure 2. Frequency spectra of accelerometer output.



(a) With filter

(b) Without filter

Figure 3. Frequency spectra of accelerometer output with and without filter.

source of fluctuation has not been identified. Additional work is needed to resolve this problem.

IV. ANALOG TO DIGITAL CONVERTERS

To interface with a microcomputer, both outputs of the accelerometer needs to be digitized. The analog-to-digital converters (ADCs) used should be of sufficient accuracy to acheive a resolution equivalent to 1 μg. There are basically two types of ADCs as far as output format is concerned. One type employ parallel output lines, the number of lines is equal to the number of bits in the digital word. An ADC of finer resolution requires longer wordlength, thus more output lines. The other type of ADC operate on the principle of voltage-to-frequency conversion and its digital output is a series of pulses passing through a single line.

Parallel Line ADC

For an acceleration range within +&- 2g and a resolution of 1 ug, it requires that the digital wordlength be long enough to represent any decimal value from -2,000,000 to +2,000,000 ug. Thus the required word-length for a parallel line type of ADC would be 22 bits including the sign bit. In addition, the required integral nonlinearity at full scale range of the ADC must be less than .5 ppm. An ADC of such accuracy was not commercially available. Design and fabrication of such an ADC would be itself a big task.

Voltage-to-Frequency Converter

The use of voltage-to-frequency converter (VFC) as ADC avoids the word-length requirement. A VFC was being designed by MICOM staff based on the principle shown in Figure 4. The unit generates a string of pulses which are counted by a pulse counter. The number of pulses generated depends on the acceleration signal received. For the device to have an accuracy better than 1 μg, each block in the figure must have an accuracy much better than 1 ug. An analysis was made to generate design considerations.

Pulse Weight With a required resolution of 1 ug, the pulse weight must be less than 1 ug/pulse.

Pulse Rate With a maximum acceleration magnitude of 2 g, a resolution of 1 μg, and a sampling rate of 100 Hz, the maximum pulse rate of the VFC is

$$\text{Max rate} = 2\text{x}10^6\text{x}100 \text{ Hz} = 200 \text{ MHz}$$

which is a very high full range rate for a VFC.

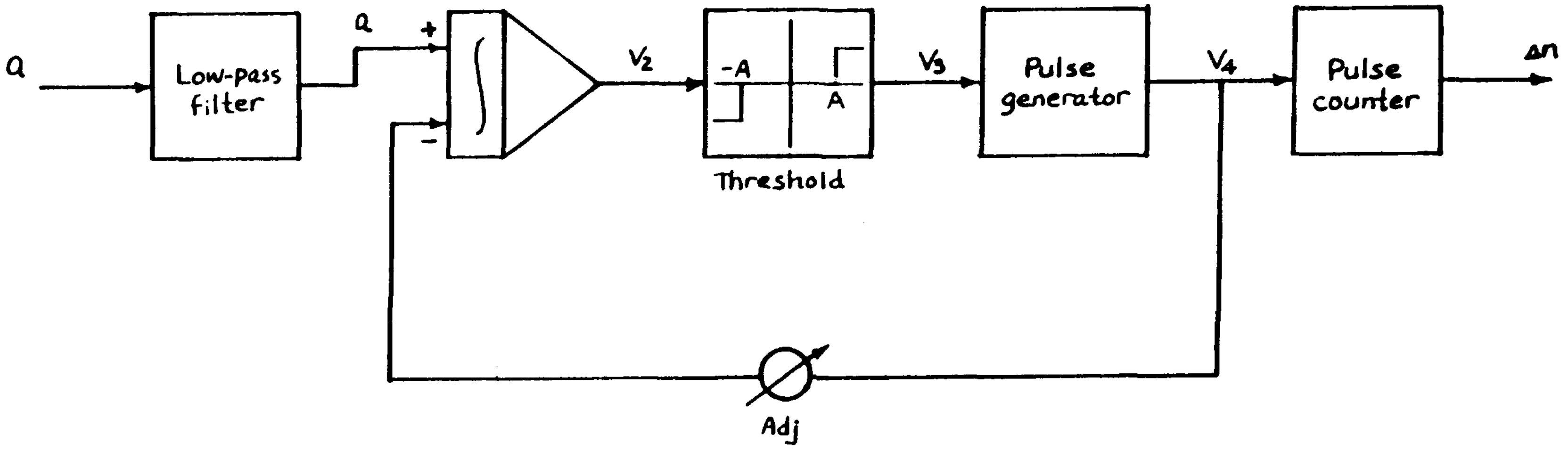


Figure 4. Block diagram of a voltage-to-frequency converter.

Pulse Width The pulse width must be less than the reciprocal of the pulse rate. Therefore,

$$\text{Pulse width} < \frac{1}{2 \times 10^8} \text{ sec} = 5 \text{ nanosec}$$

which is a very narrow pulse.

Anti-aliasing To avoid aliasing error, the bandwidth of analog acceleration signal, which contains heavy high frequency noise, should be limited by a low-pass filter. Knowledge and stability of the gain of this filter should both be better than a fraction of one ppm.

Integrator Clearing As shown in Figure 4, the generated pulses also serve to clear the integrator (setting integrator output to zero) by a negative feedback arrangement. Very accurate clearing is required. Ideally, the clearing by each pulse should be accomplished in zero time. Such an instantaneous response is not physically possible, and this results in convertion error. However, this error can be accurately compensated by software in the microcomputer. The development of an algorithm for this purpose is given here.

Figure 5 (a) shows a sample acceleration signal appearing at the input of the integrator shown in Figure 4 over one sampling period T. The corresponding signal at the output of the integrator is shown in Figure 5 (b). Notice that there are six completed saw teeth in time period T, and they generate six pulses at the output of the pulse generator. The seventh saw tooth has not reached the threshold level and therefore has not generated an output pulse in this sampling period. Since there is no fractional pulse count, the result is a pulse count error for this sampling period. However, the missed pulse count will contribute to the pulse count of the next sampling period. Therefore mis-counts in individual sampling periods are not additive. The maximum mis-count at any time is one pulse. Thus, in analyzing the non-zero clearing time problem of the integrator, the mis-count in each sampling period will be neglected.

Let the completed n saw teeth be spanned over the sampling period T and be approximated by same shape as shown in Figure 5 (c). This assumption is reasonable since the variation in acceleration is considered to be very slow for the sampling rate of 100 Hz. Each saw tooth spans a time

$$t_t = t_1 + t_2 \tag{9}$$

Each t_t consists of two parts. The first part t_1 is the time for the integrator to integrate from zero to the threshold,

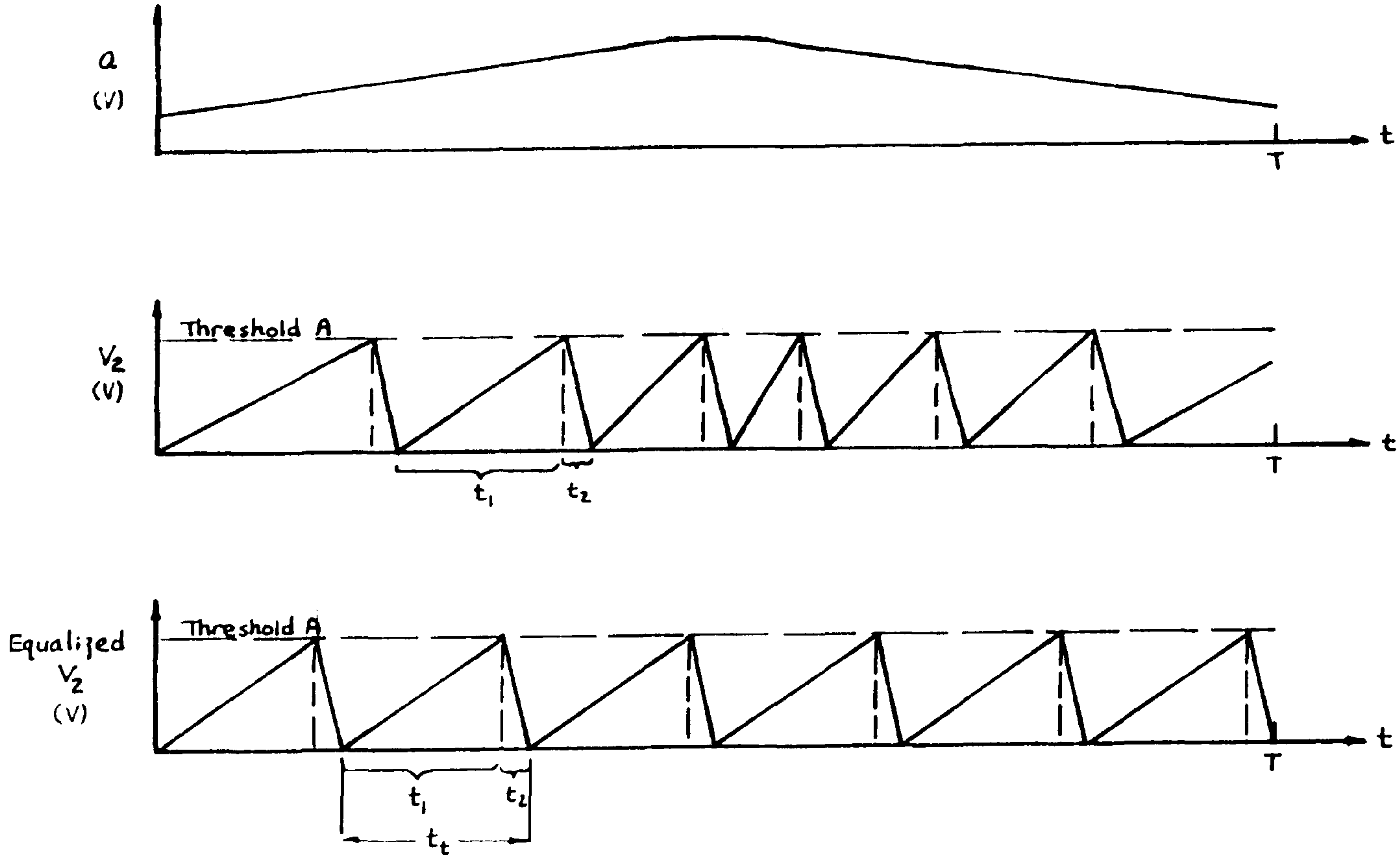


Figure 5. Acceleration, integrator output, and equalized integrator output signals.

and the second part t_2 is the clearing time. During the clearing time, the integrator integrate the sum of two inputs, one is the acceleration signal from accelerometer and the other is the negative feedback pulse from the pulse generator. For rapid clearing, it is neccesary that the fedback pulse be of very large in amplitude and be very narrow in width as compare to the width of each saw tooth. Since the maximum width of the saw tooth is limited to 5 nanosecond wide as discussed previously, the width of the clearing pulse needs to be much narrower, say, one tenth of the saw tooth width. In numerical value,

Width of clearing pulse = 5/10 = .5 nanosec

After recording n pulses in a sampling period T, the average pulse width t_t can be computed using (9). In the duration of each saw tooth generation, the time t_1 contributes to the analog to digital conversion operation, but the time t_2 does not. The present analysis is aiming at estimating the total number of t_2's in each sampling period and determining the needed pulse count compensa-tion for the period. There are three unknown quantities in each saw tooth, namely, t_1, t_2, and the ramp up slope a which is the input acceleration. The ramp down slope is a-b. While a is not known, b is known because it is the magnitude of the feedback clearing pulse designed. The various quantities are related by the following three equations.

$$a t_1 = A \tag{10}$$

$$(a-b)t_2 = -A \tag{11}$$

$$t_1 + t_2 = t_t \tag{12}$$

where A is the value of threshold. Solving the three equations gives

$$a = \frac{1}{2} b\left[1 - \sqrt{1 - \frac{4A}{b t_t}}\right] \tag{13}$$

$$t_1 = \frac{A}{a} \tag{14}$$

$$t_2 = t_t - t_1 \tag{15}$$

The needed compensation for mis-count due to finite clearing time is given by

$$\Delta n = n \frac{t_2}{t_1} \qquad (16)$$

and the compensated pulse count for each sampling period is

$$n_c = n + \Delta n \qquad (17)$$

As an example, consider T= 24, n=6, A=2, and b=3. Using (9), and (13) to (17), gives

$$t_T = \frac{24}{6} = 4$$

$$a = \frac{1}{2} \times 3 \times \left[1 - \sqrt{1 - \frac{4 \times 2}{3 \times 4}} \right] = .6339746$$

$$t_1 = \frac{2}{a} = 3.15470$$

$$t_2 = 4 - t_1 = .8453$$

$$\Delta n = 6 \times \frac{t_2}{t_1} = 1.608$$

and, *finally the compensated* pulse count

$$n_c = n + \Delta n = 6 + 1.608 = 7.608$$

which is the correct value.

<u>Integrator Gain</u> The given system performance specification requires that the gain of the integrator be accurate to a fraction of one ppm. An analysis of the integrator is given with the help of Figure 6. Figure 6 (a) shows the schematic diagram of the integrator where

E_1 and E_2 represent the ideal acceleration signal and feedback clearing signal, respectively,

R_1 and R_2 represent the source resistances,

C is the feedback capacitance, and

K_a is the amplifier gain.

The equivalent circuit of the amplifier is shown in Figure 6 (b)

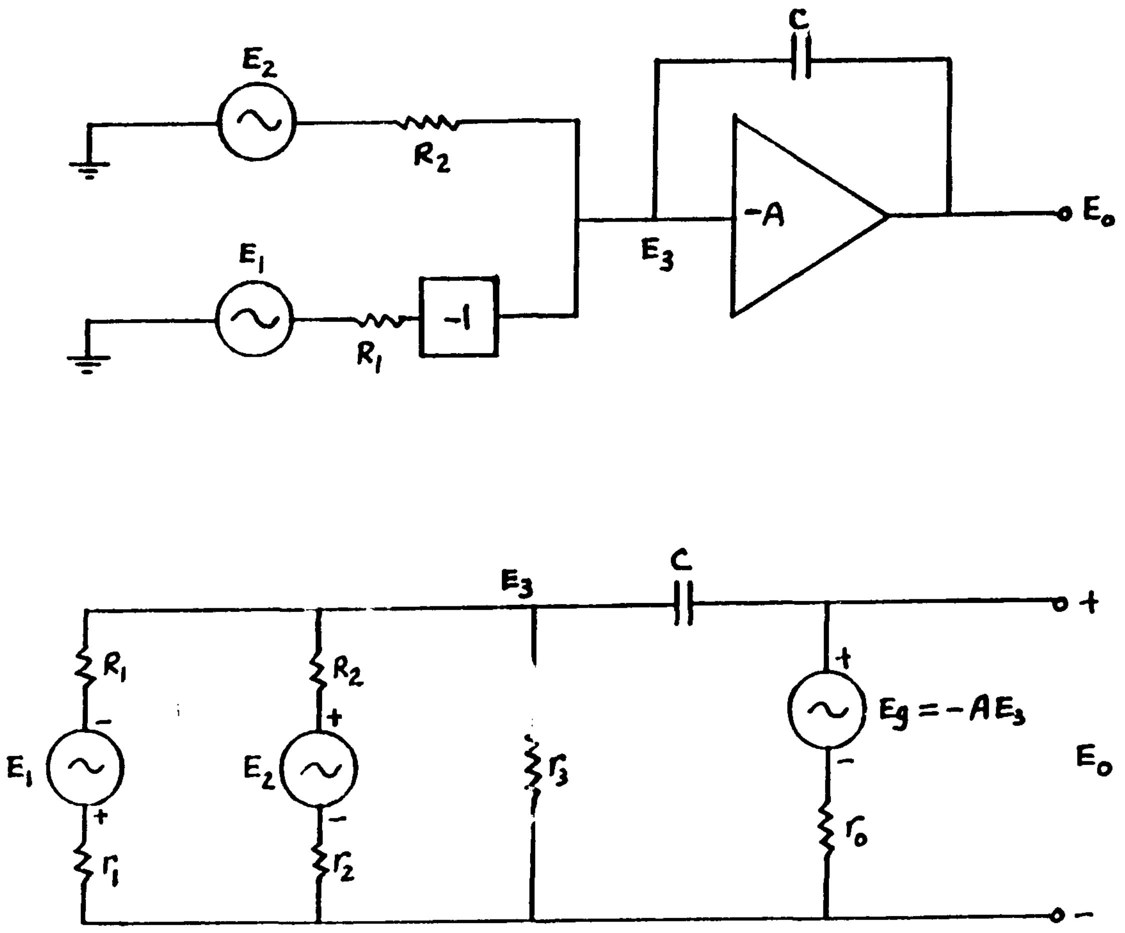


Figure 6. Circuit diagram of ADC's summing integrator.

where r_3 represent the input resistance of the amplifier, r_0, the output resistance, and E_g, the dependent voltage generator. By the usual circuit analysis technique, the equation for output voltage is obtained as

$$E_0 = \frac{1}{CS}\left[\frac{E_1}{R_1+r_1} - \frac{E_2}{R_2+r_2}\right]\frac{1-\frac{1}{K_a}(r_0CS)}{1+\frac{1}{K_a}\left(1+\frac{r_0CS+1}{RCS}\right)} \qquad (18)$$

where R is the parallel combination of $r_1 + R_1$, $r_2 + R_2$, and r_3.

By examining (18), two remarks can be made. First, For the integrator to be sufficiently accurate, the gain K_a should be greater than 10 million. Next, the integrator gain for the two inputs are different, namely,

$$G_1 = \frac{\frac{1}{CS}}{R_1\left(1+\frac{r_1}{R_1}\right)} \qquad (19)$$

$$G_2 = \frac{\frac{1}{CS}}{R_2\left(1+\frac{r_2}{R_2}\right)} \qquad (20)$$

To know these gain values better than one ppm requires the knowledge of all parameters in (19) and (20) to better than one ppm, which is not easy. However, this problem can probably be coped with by determining the gain values through testing. Then maintaining the stability of these gains will be the remaining requirement.

Threshold The requirement for the threshold is that it be sufficiently stable with respect to temperature variation.

Pulse Generator The precision of the pulse generator is important. Area of each pulse generated should be equal to the value of threshold. For a pulse of .5 nanosec width, meeting this condition requires a carefully designed pulse generator.

Counter The counter is the only part in the VFC which does not present difficulty. It is easy to build and is usually very reliable.

V. SYSTEM SCHEME AND SOFTWARE

The proposed system scheme for the single axis inertially aided position control is shown in Figure 7. A vertical dash line divides the system into two parts. The left-hand side of the diagram represents the hardware part of the system while the right-hand side represents the software part. A number in parentheses indicates either the unit of a quantity or the gain of a block. Each block in the figure will be described separately in the following.

Blocks 1, 2, and 3 Details of the accelerometer and ADC have been discussed in previous sections. To be added here are that Block 1, the accelerometer, has a gain in volts/g, and Block 2 and 3, representing ADC's, have gain in counts/volt-sec.

Block 4 This block performs the computation which includes scaling of the input signal m_{TS} coming from the temperature sensor ADC, obtaining temperature sensor current I_{TS} , and computing the value of a current quantity u. Scaling is needed because the gain of temperature sensor ADC may not be unity and the need of a voltage conversion from volts to micro-volts. Conversion to current is needed since the input m_{TS} represents a voltage. u is linearly related to I_{TS} through a function furnished by the accelerometer manufacturer. The computation algorithm of this block consists of the folowing.

$$V_{TS} = \frac{m_{TS}}{K_4} \qquad (\mu A) \qquad (21)$$

$$I_{TS} = \frac{V_{TS}}{R_{TS}} \qquad (\mu A) \qquad (22)$$

$$u = I_{TS} - 293 \qquad (\mu A) \qquad (23)$$

where $K_4 = K_3 \times 10^{-6}$, K_3 is the gain of Block 3, and R_{TS} is the load resistance in the temperature sensor output circuit. The output of Block 4 is u in µA.

Block 5 Using the data from Block 4, this block computes the bias and scale factor of the accelerometer. The computation algorithm is given by (6) and (7) which are rewritten here.

$$S = c_0 + c_1 u + c_2 u^2 + c_3 u^3 + c_4 u^4 \qquad (mA/g) \qquad (24)$$

$$B = b_0 + b_1 u + b_2 u^2 + b_3 u^3 + b_4 u^4 \qquad (mg) \qquad (25)$$

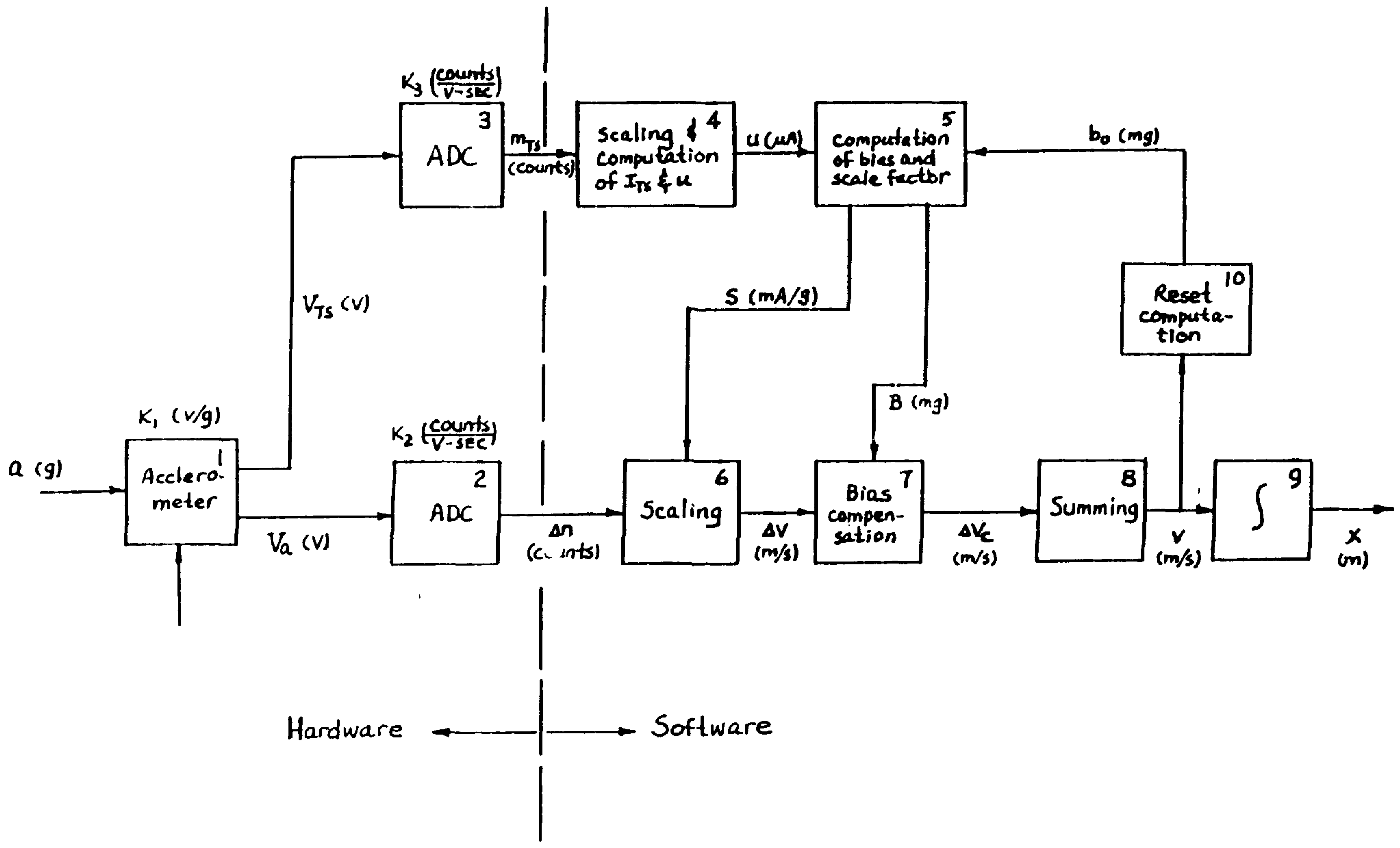


Figure 7. Single axis inertially aided position determination block diagram.

Block 6 This block does two things. First, it converts the scale factor S in mA/g to a scale factor S_p in counts/meter/sec. Then it computes the incremental velocity Δv from the incremental number of counts Δn. The algorithm includes

$$S_p = S \times R_L \times T \times 10^{-3} \times 9.8 \qquad \text{(Counts/m/sec)} \qquad (24)$$

$$\Delta v = \frac{\Delta n}{S_p} \qquad \text{(m/sec)} \qquad (25)$$

where R_L is the load resistance in ohms of the accelerometer as shown in Figure 1.

Block 7 This block does bias compensation for each incremental velocity. It computes the incremental velocity compensation Δv_B and the compensated incremental velocity Δv_c. The associated algorithm includes

$$\Delta V_B = B \times 9.8 \times 10^{-3} \times T \qquad \text{(m/sec)} \qquad (28)$$

$$\Delta V_c = \Delta V + \Delta V_B \qquad \text{(m/sec)} \qquad (29)$$

Block 8 and 9 Block 8 sums up all incremental velocities to give the velocity v using the equation

$$V = \sum \Delta V_c \qquad \text{(m/sec)} \qquad (30)$$

Block 9 integrates the velocity to give the position. The integration uses the Simpson rule algorithm having the following computation equation.

$$X_n = X_{n-2} + \frac{T}{3}\left(V_n + 4V_{n-1} + V_{n-2}\right) \qquad (m) \qquad (31)$$

Block 10 This block has a software switch which is normally off. The switch is closed during reset operation when the end-effector of the manipulator is returned to its reset station. At the station, the velocity of the end-effector should be zero. Any non-zero velocity at the output of Block 8 indicate error due to bias. This information is used to compute an average bias $\bar{B}$ which is then used to correct the b_0 coefficient of the bias equation in Block 5. Then, the initial position of the end-effector is reset to the reference value x_r . The associated algorithm consists of

$$B_n = \frac{V_n - V_{n-1}}{T} \qquad n = 1 \text{ to } N_B \tag{32}$$

$$\bar{B} = \frac{1}{N_B} \sum_{n=1}^{N_B} B_n \tag{33}$$

$$b_o = \bar{B} - \sum_{k=1}^{4} b_k u^k \tag{34}$$

$$x = x_r \tag{35}$$

where N_B is the number of velocity data used to compute $\bar{B}$ and x_r is the position of the reset station.

The above equations, including (21) to (35), can be coded into computer software format. The overall software flow chart is shown in Figure 8. The control sampling rate is specified to be 100 Hz, therefore the computation for position will be done at the same rate. Since temperature variation would be slow, updating the scale factor and bias using the data from temperature sensor is chosen to be once per second. The command for reset comes into the system as an interrupt which is effected only after the position computation is completed in a computation cycle.

Using double precision, all numbers will have 16 decimal digit accuracy which is more the accuracy rquired. Therefore, errors due to software can easily be made under control.

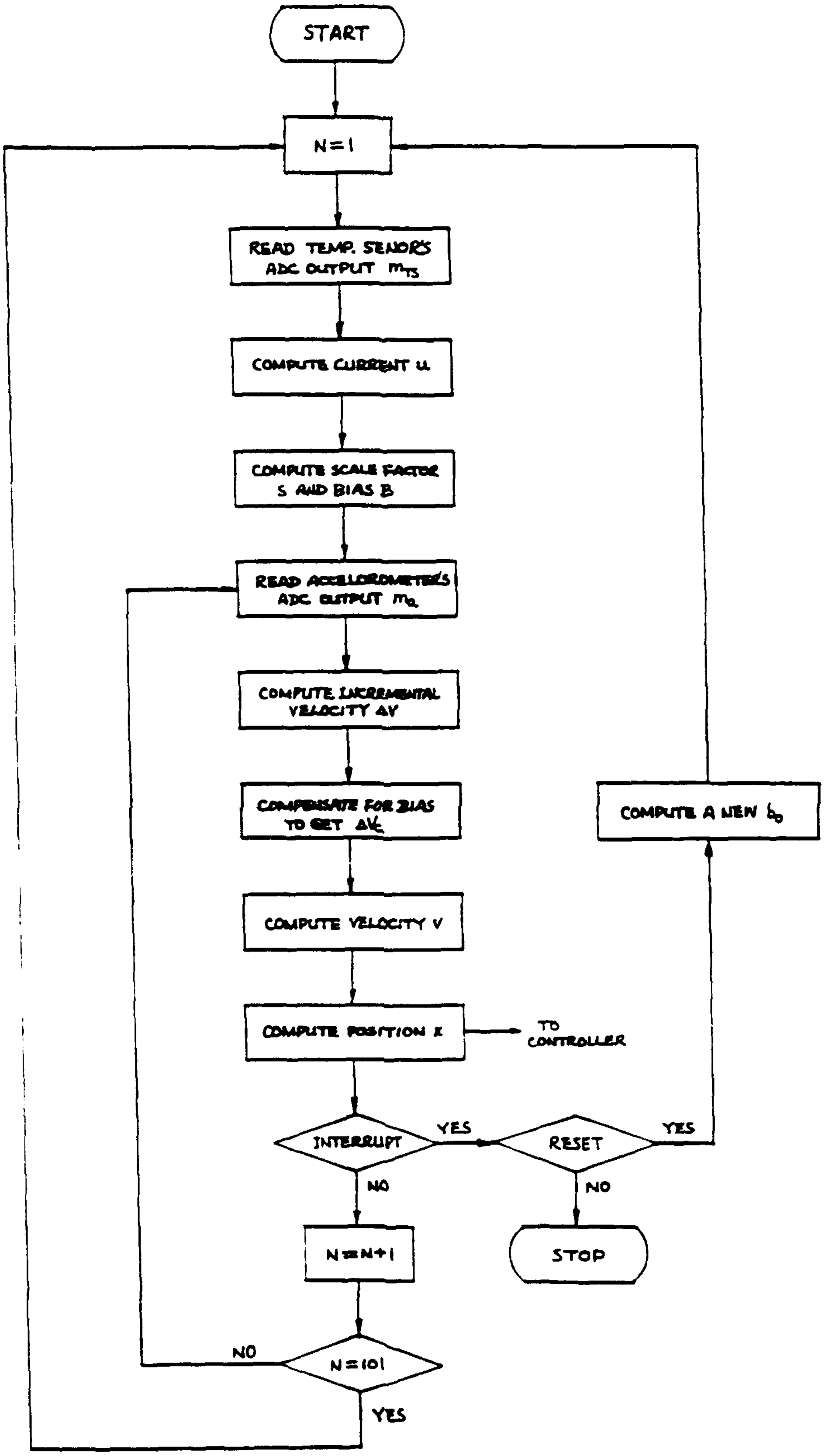

START
N=1
READ TEMP. SENOR'S ADC OUTPUT m_{TS}
COMPUTE CURRENT u
COMPUTE SCALE FACTOR S AND BIAS B
READ ACCELLOROMETER'S ADC OUTPUT m_a
COMPUTE INCREMENTAL VELOCITY ΔV
COMPENSATE FOR BIAS TO GET $ΔV_c$
COMPUTE VELOCITY V
COMPUTE POSITION X
TO CONTROLLER
INTERRUPT
YES
NO
RESET
YES
NO
COMPUTE A NEW b_0
STOP
N=N+1
N=101
NO
YES

Figure 8. System software flow chart.

VI. EXPERIMENT AND RESULT

Because of time limitation in this project phase, the originally planned experminent of the single-axis inertially aided mani-pulator control was simplified. The simplified experiment was performed to test the short term stability of the accelerometer. Instead of mounting the accelerometer on the end-effector of a manipulator, it was mounted on the work peice table of a milling machine. The table was moved along the input axis of the accelerometer to produce an input for the accelerometer. The position of the table was sensed by an Anilan linear encoder having a resolution of .0005 inch or 12.7 micron which was accurate enough to be considered true position. The output of the accelerometer was digitized and processed by a microcomputer to give the inertially generated position. The generated position was then compared to the true position. The expermental setup is conceptually shown in Figure 9.

The data processing involved determining and removing the accelerometer bias and performing integrations for position. The software used is given in Appendix B.

Two kinds of tests were made, namely, the single motion test and the multiple motion test. The result of single position test showed that the accelerometer data generated position was off from the true positon by -.071515 inch in a total travel distance of 1.2112 inches and in 5.8 seconds of time, a 6% error. The position error is far bigger than the specification value of .004 in. By futher refining the data processing software, some improvement in accuracy probably can be obtained, but not to the accuracy of .004 in. The data collected from the multiple position test seemed to be in error, since the computed position was too far off from the true position. The source of the error has not been identified. It may be from the accelerometer, the data reduction software, or from errorneous data recording.

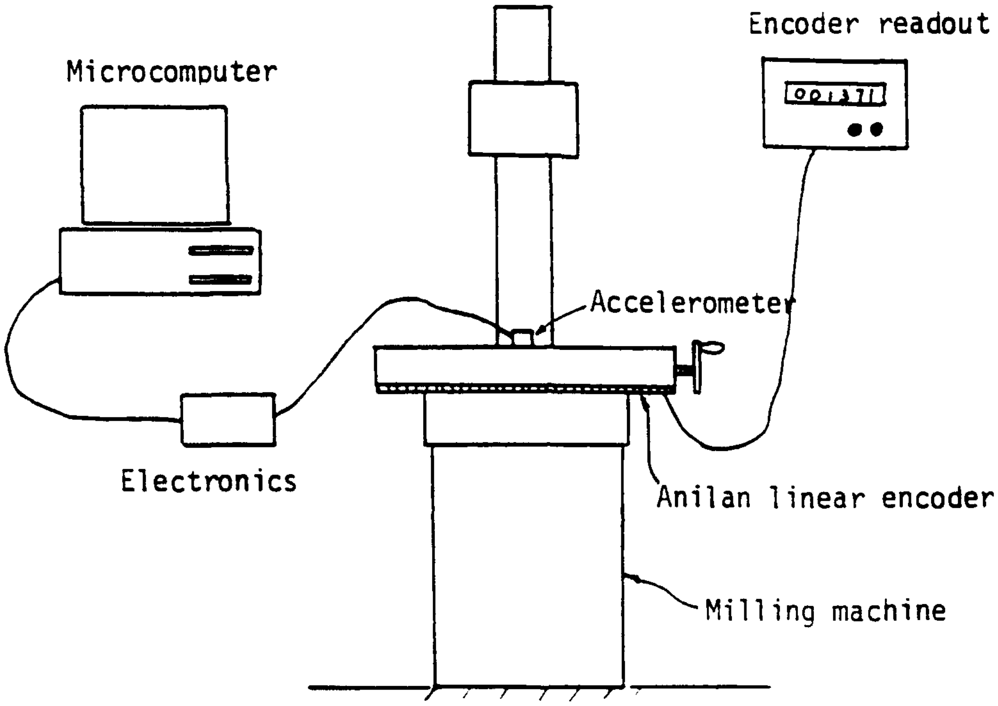


Figure 9. Simplified experiment setup.

VII. CONCLUSIONS AND RECOMMENDATIONS

Conclusion

From the results presented in this report, the following conclusions can be made concerning the development of an inertially aided robot manipulator control system in general:

1. The concept is theoretically very sound.

2. Needed software for the system can be developed. Software development may take time, but does not present difficulty.

3. The computation speed of a micro-computer is fast enough to handle the required computation for the system.

4. The hardware part of the system requires a good deal of research and development effort. First of all, to achieve the accuacy specification of this project, the quality of the accelerometer needs to be much better than that of Q-Flex QA2000. Next, The electonics, including amplifiers, ADCs, and filters, need to be specially designed for very high precision and very low sensitivity. Hardware stability better than a fraction of one ppm is needed.

5. A successful development of the system will be a long term project, say, 2 to 4 years. The long period of time is needed for studying and contemplating.

6. It should be pointed out that the a successful development of the system, taking the cost as well as cost performances into consideration, may be hard to quarantee.

7. It should also be pointed out that there are practical applications where the required performance specifications are much less stringent than that of the present project. For those cases, inertially aided control would be much easier to achieve. In fact, the results of this project would be useful for those applications.

Recommendations

1. It is reasonable to believe that inertially aided control will have important use in some areas of robotics that Army will be interested.. Therefore a continued research and development effort in this

area is warranted. However, the R&D work should be of long term nature, say, 2 to 4 years, and at low funding level.

2. Further development may not be fruitful if only a short term effort is planned.

3. As far as the present project is concerned, the incomplete and non-conclusive data reduction of the preliminary experiment is a disappointment. This work should be completed to discover the source of error regardless of funding situation.

REFERENCES

1. R. P. Paul, "Robot Manipulator," MIT Press, 1981.

2. W. E. Snyder, "Industrial Robots: Computer Interface and Control," Prentice-Hall, 1985.

3. J. J. Craig, "Introduction to Robotics," Addison-Wesley, 1986.

4. H. Asada and J. J. E. Slotine, "Robot Analysis and Control," Wiley-Intercience, 1986.

5. J. C. Hung, "Inertial Measurement Based Robot Manipulator Control", (To appear)

6. G. R. Pitman, Jr., "Inertial Guidance," John Wiley & Sons, 1962.

7. J. C. Hung, "A Study of Strapdown Platform Technology," NASA CR-123710, Clearinghouse for Federal Scientific and Technical Information, Springfield, Virginia, July, 1972.

8. G. A. Blake, a letter, Reference No. 028-GAB-88, Sundstrand Data Control, Inc., 28 April 1988.

APPENDIX A

SOFTWARE FOR EXTRACTING TEMPERATURE COEFFICIENTS

OF Q-FLEX QA2000 ACELEROMETERS

```
10   REM ********** Q-Flex Temperature Coefficients Determination
20   REM                          881208/JCH
30   A$="Q-Flex P/N 979-2020-001, S/N 102 "
40   LPRINT A$:     LPRINT
50   OPTION BASE 1
60   REM ********** There are M measurements and N unknowns
70   M=12:     N=5
80   DIM  TEMP(M), SFV(M), BIAS(M), VTS(M), OLDC(N), OLDB(N)
90   DIM  U(M), SFI(M), C(N), B(N)
100   DIM  P(M,N), PT(N,M), PTP(N,N), PTPI(N,N), GINV(N,M)
110   REM
120   REM ********** Read in data and compute U and SFI vectors
130   REM
140   FOR J=1 TO M:     READ TEMP(J):     NEXT J
150      DATA -5.5, -31.7, -46.2, -29.2, -5.2, 21.1
160      DATA 46.6, 69.9, 94.1, 67.4, 45.7, 21.8
170   FOR J=1 TO M:     READ SFV(J):     NEXT J
180      DATA 4.90472, 4.89684, 4.89260, 4.89744, 4.90529, 4.91589
190      DATA 4.92807, 4.94124, 4.95406, 4.93938, 4.92743, 4.91600
200   FOR J=1 TO M:     READ BIAS(J):     NEXT J
210      DATA 1.71467, 2.18815, 2.43327, 2.18584, 1.82558, 1.40361
220      DATA 0.97604, 0.64963, 0.31489, 0.62356, 1.02258, 1.36086
230   FOR J=1 TO M:     READ VTS(J):     NEXT J
240      DATA 2.69127, 2.45186, 2.30429, 2.46828, 2.69644, 2.95094
250      DATA 3.20136, 3.43965, 3.65582, 3.41283, 3.20018, 2.95404
260   FOR J=1 TO N:     READ OLDC(J):     NEXT J
270      DATA 1.233831, 1.458E-04, 3.698E-07, -3.35E-10, .69E-12
280   FOR J=1 TO N:     READ OLDB(J):     NEXT J
290      DATA 1.603, -1.538E-02, 6.2E-06, -9.9E-08, -1.5E-09
300   FOR J=1 TO M:     U(J)=VTS(J)*100-293:     NEXT J
310   FOR J=1 TO M:     SFI(J)=(SFV(J)/3984.21)*1000:     NEXT J
320   REM ********** PRINT OUT RAW DATA
330   LPRINT "Raw data from test:":  LPRINT
340   LPRINT " Temp. deg C     SFV v/g       Bias mg       Temp.sensor v"
350   FOR J= 1 TO 12
360      LPRINT USING "####.######  "; TEMP(J), SFV(J), BIAS(J), VTS(J)
370      NEXT J
380   REM
390   REM ********** Set up regression matrix P
400   FOR I=1 TO M:     FOR J=1 TO N
410      P(I,J)=U(I)^(J-1)
420      NEXT J:     NEXT I
430   REM
440   REM Compute generalized inverse matrix
450   FOR I=1 TO N:   FOR J=1 TO M
460      PT(I,J)=P(J,I):     NEXT J:     NEXT I
470   FOR I=1 TO N:     FOR J=1 TO N:     PTP(I,J)=0
480      FOR K=1 TO M:     PTP(I,J)=PTP(I,J)+PT(I,K)*P(K,J)
```

```
490     NEXT K:    NEXT J:    NEXT I
500  REM
510  REM ********** MATRIX INVERSION SUBROUTINE **********
520  REM
530  REM  N is the dimension of the matrix.
540  REM  PTP is the input matrix, dimension NxN.
550  REM  PTPI the output matrix, dimension NxN.
560  REM
570  DIM  HX(N,N), T(N,N)
580  FOR I=1 TO N
590  FOR J=1 TO N
600  HX(I,J)=PTP(I,J)
610  NEXT J
620  NEXT I
630  FOR I=1 TO N
640  FOR J=1 TO N
650  T(I,J)=0
660  IF I=J THEN T(I,J)=1
670  NEXT J
680  NEXT I
690  DET=1
700  FOR I=1 TO N
710  MD=I
720  FOR J=I TO N
730  IF ABS(HX(J,I)) > ABS(HX(MD,I)) THEN MD=J
740  NEXT J
750  IF MD=I THEN GOTO 870
760  FOR J=I TO N
770  S=HX(I,J)
780  HX(I,J)=HX(MD,J)
790  HX(MD,J)=S
800  NEXT J
810  FOR J=1 TO N
820  S=T(I,J)
830  T(I,J)=T(MD,J)
840  T(MD,J)=S
850  NEXT J
860  DET=-1 * DET
870  DET=HX(I,I) * DET
880  S=1!/HX(I,I)
890  FOR J=1 TO N
900  IF J>I THEN HX(I,J)=S * HX(I,J)
910  T(I,J)=S * T(I,J)
920  NEXT J
930  FOR J=1 TO N
940  IF J=I THEN GOTO 1000
950  S=HX(J,I)
960  FOR K=1 TO N
970  IF K>I THEN HX(J,K)=HX(J,K) - S * HX(I,K)
980  T(J,K)=T(J,K) - S * T(I,K)
990  NEXT K
1000  NEXT J
1010  NEXT I
1020  FOR I=1 TO N:    FOR J=1 TO N
```

```
1030  PTPI(I,J)=T(I,J):     NEXT J:     NEXT I
1040  REM ********** Matrix inversion done **********
1050  REM
1060  FOR I=1 TO N:      FOR J=1 TO M:      GINV(I,J)=0
1070     FOR K=1 TO N:      GINV(I,J)=GINV(I,J)+PTPI(I,K)*PT(K,J)
1080     NEXT K:     NEXT J:     NEXT I
1090  FOR I=1 TO N:    C(I)=0:     B(I)=0:     FOR K=1 TO M
1100     C(I)=C(I)+GINV(I,K)*SFI(K):     B(I)=B(I)+GINV(I,K)*BIAS(K)
1110     NEXT K:     NEXT I
1120  LPRINT:     LPRINT "New coefficients:":      LPRINT
1130  FOR I=1 TO N
1140     LPRINT "     B"I-1" = "B(I), "C"I-1" = "C(I)
1150     NEXT I
1160  LPRINT:     LPRINT "Old coefficients:":      LPRINT
1170  FOR I=1 TO N
1180     LPRINT "     B"I-1" = "OLDB(I), "C"I-1" = "OLDC(I)
1190     NEXT I
1200  END
```

APPENDIX B

SOFTWARE FOR PROCESSING ACCELEROMETER OUTPUT DATA

```
10  REM ********** DISTANCE COMPUTATION ********** (GETV&D.BAS)
20  DIM A(4100), V(4100), S(4100)
30  REM ********** READ IN RAW DATA IN volts
40  OPEN "R",1,"IARMMT03.RAW",8
50  FIELD 1, 4 AS X$, 4 AS Y$
60  FOR K=20 TO 4019
70    GET 1, K:  A(K-19)=CVS(X$)/4096          'A in volts.
80    NEXT K
90  REM ********** CONVERT TO in/sec-square
100 SFA=3.864                                   'in/sec-sq per volt
110 FOR J=1 TO 4000
120   A(J)=A(J)*SFA
130   NEXT J
140 REM ********** COMPUTE AND REMOVE BIAS
150 ASUM=0
160 FOR K=1 TO 200
170   ASUM=ASUM+A(K)
180   NEXT K
190 AAVE=ASUM/200
200 FOR J=1 TO 4000
210   A(J)=A(J)-AAVE
220   NEXT J
230 REM ********** INTEGRATE TO GET VEL AND DIST BY SIMPSON'S ALGORITHM
240 DT=.01
250 C=DT/3
260 V(0)=0
270 FOR K=2 TO 4000 STEP 2
280   V(K)=V(K-2)+(A(K)+4*A(K-1)+A(K-2))*C   'Integrate to get velocity
290   NEXT K
300 FOR N=1 TO 3999 STEP 2
310   V(N)=(V(N+1)+V(N-1))/2                  'Get mid-point velocity
320   NEXT N
330 S(0)=0
340 FOR K=2 TO 4000 STEP 2
350   S(K)=S(K-2)+(V(K)+4*V(K-1)+V(K-2))*C   'Integrate to get velocity
360   NEXT K
370 REM ********** PRINT OUT RESULT
380 FOR J=2 TO 2000 STEP 2
390   M=J+2000
400   LPRINT J, S(J), M, S(M)
410   NEXT J
420 END
```