CONF-970970--1

Instrumentation and Controls Division

Controls and Systems Integration Section

# Closed-Loop Step Motor Control Using Absolute Encoders


J. S. Hicks

M. C. Wright

Oak Ridge National Laboratory[1]*

Lockheed Martin Energy Research

P.O. Box 2008

Bldg. 3606, MS 6003

Oak Ridge, Tennessee 37831-6003

Ph. (423) 576-7342

FAX (423) 574-4529





Submitted to the

Powersystems World International Conference & Exhibit

Baltimore, Maryland

September 6-12, 1997




[1]*Managed by Lockheed Martin Energy Research Corp. for the U.S. Department of Energy under contract number DE-AC05-96OR22464.

## DISCLAIMER

# Closed-Loop Step Motor Control Using Absolute Encoders

by

J. Steven Hicks

Michael C. Wright

Oak Ridge National Laboratory*

Instrumentation and Controls Division

Oak Ridge, Tennessee 37831

**Abstract** - A multi-axis, step motor control system was developed to accurately position and control the operation of a triple axis spectrometer at the High Flux Isotope Reactor (HFIR) located at Oak Ridge National Laboratory. Triple axis spectrometers are used in neutron scattering and diffraction experiments and require highly accurate positioning. This motion control system can handle up to 16 axes of motion. Four of these axes are outfitted with 17-bit absolute encoders. These four axes are controlled with a software feedback loop that terminates the move based on real-time position information from the absolute encoders. Because the final position of the actuator is used to stop the motion of the step motors, the moves can be made accurately in spite of the large amount of mechanical backlash from a chain drive between the motors and the spectrometer arms. A modified trapezoidal profile, custom C software, and an industrial PC, were used to achieve a positioning accuracy of 0.00275 degrees of rotation. A form of active position maintenance ensures that the angles are maintained with zero error or drift.

## 1. INTRODUCTION

As with many problems in engineering, there is often more than one solution or approach that can be made to work. User familiarity, backward compatibility, future compatibility, and budget all had an influence on the final system design. Starting with the right components is the key to successful systems integration. Often the most cost effective solution is a complete integrated turnkey system from a distributor or value added re-seller. However, sometimes unique system requirements do not match up with COTS (Commercial Off-the-Shelf) hardware well enough to allow an outright "plug and play" solution. A good systems integrator can adapt COTS hardware with custom software or interfaces to meet the customers' "wants" as well as needs. As the end users' desired capabilities evolve over time, and requirements change, custom solutions also have the added benefit of future flexibility. The downside to developing one's own hardware and software is the added cost of development time. The rest of this paper will describe the need for and then the design of a closed-loop step motor control system that exceeded customer requirements.

* Managed by Lockheed Martin Energy Research Corp. for the U.S. Department of Energy under contract number DE-AC05-96OR22464.

### 1.1 Description of Neutron Spectrometer

A triple-axis neutron spectrometer consists of four functional parts: a source of neutrons of a certain energy or wavelength (in this case a reactor and a monochromater crystal), a means of precisely locating a sample in this beam of neutrons (a sample holder), a way of selecting neutrons of a certain wavelength that have scattered from the sample (analyzer crystal), and a way to count neutrons at different angles as they leave the analyzer crystal (detectors) (Figures 1 and 2). Precise control of the angular relationship between these components is fundamental to the operation of a spectrometer. Neutrons of a desired energy are selected with a monochromater crystal. The first axis, supporting the entire spectrometer, determines the angle at which neutrons leave the monochromater. These neutrons form a beam that is directed into the sample volume, which is at a fixed distance above the sample table. The sample table sits co-axially on top of the second axis, called the PHI axis, and rotates with this arm. The sample can also be rotated about two other mutually orthogonal axes above the sample table, but these axes are not shown in the figures. Near the end of the 41" arm sits an assembly that consists of another arm called the 2-Theta-A axis. The analyzer crystal (not shown in the figures) sits on a rotary table that is mounted co-axially on top of this third arm, which is 36" long. The detector sits at the end of this third arm. The 2-Theta-A axis rotates in the same plane as the first and second axes, which are parallel to the floor of the beam room. When both the second and third arms are aligned, the center of mass of the detector is cantilevered a total of ~60" from the center of rotation of the PHI axis. The detector shielding is quite heavy, on the order of 1000 lbs. Because the PHI axis supports the rest of the spectrometer, including the detector, it has a large moment of inertia.

### 1.2 Background on Project

The beam room facility at HFIR supports three triple axis spectrometers for basic physics and material science research. The spectrometers were aging and mechanically worn. One of the spectrometers was upgraded with new arms, bearings, rotary tables, absolute encoders, and stepper drives. Some software and electronic modifications were made to the FORTRAN code on the Digital VAX computer and Computer Automated Measurement And Control (CAMAC) modules that run the spectrometer and collect data in order to handle the new 17-bit absolute encoders. These encoders provide 0.002747 degrees of angular position resolution for PHI, 2-Theta-A, and the rotary tables that sit on these axes as described above. The first system upgrade did not fully satisfy the user's requirements because of missed moves and less than optimal accuracy. The system was not fully utilizing the resolution of the new encoders. However, the users liked the absolute encoders because they did not require re-initialization after power failures or between experimental setups.

### 1.3 Problems with the First System

An investigation revealed the following problems. PHI, which is the most demanding axis, was being operated with a maximum speed that was very close to the primary resonance speed of the step motor. A chain drive is used to connect the step motor to the worm gear. Even with a pre-tensioned chain, enough windup exists to cause significant backlash (~ 0.1 degrees). This backlash even varies slightly over the 230 degree range of motion. The PHI arm and gear do not rotate in a perfect plane, causing the worm and gear to bind at certain points in their travel. It is essential that the bearings be properly supported because they are carrying significant loads caused by the heavy overhung loads on both arms. The suspected cause of this run-out was faulty bearing pre-load and support. The second upgrade addressed this problem through a different bearing support design. The motor was also being micro-stepped at 7,200 pulses per revolution. Tests were conducted with dial indicators placed on the sprockets and PHI arm. Data from the indicators and from the encoder both revealed "lost resolution" or missing steps. The PHI motor is a NEMA 42 stepper with 2000 oz-in of holding torque. The motor does not generate a full 2000 oz-in of torque for each microstep. Approximately 20 microsteps were needed before the motor developed enough torque to rotate. The code on the VAX had a minimum move distance that was approximately equal to the measured lost resolution; therefore, this problem was masked and not

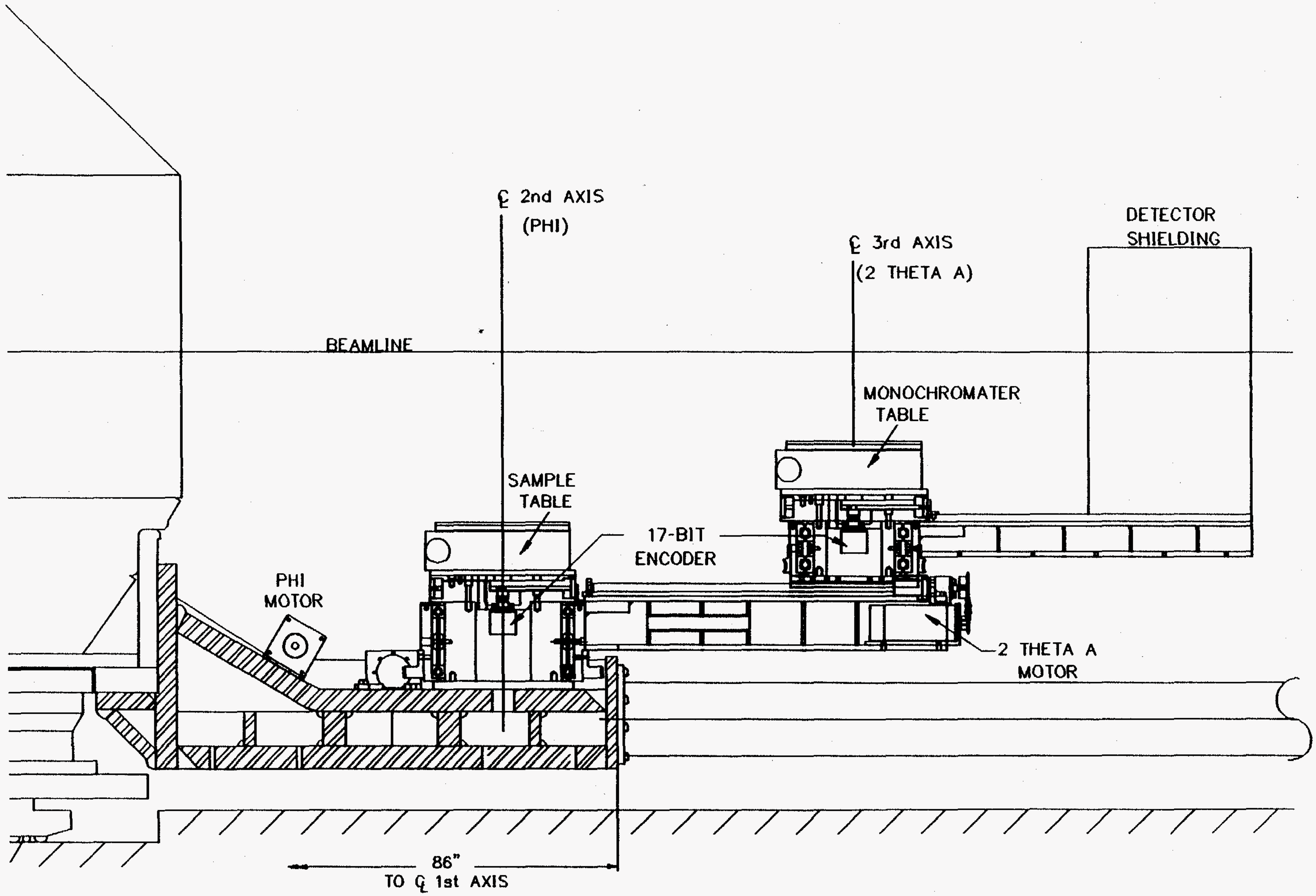


Fig. 1 CUT AWAY SIDE VIEW OF TRIPLE-AXIS SPECTROMETER

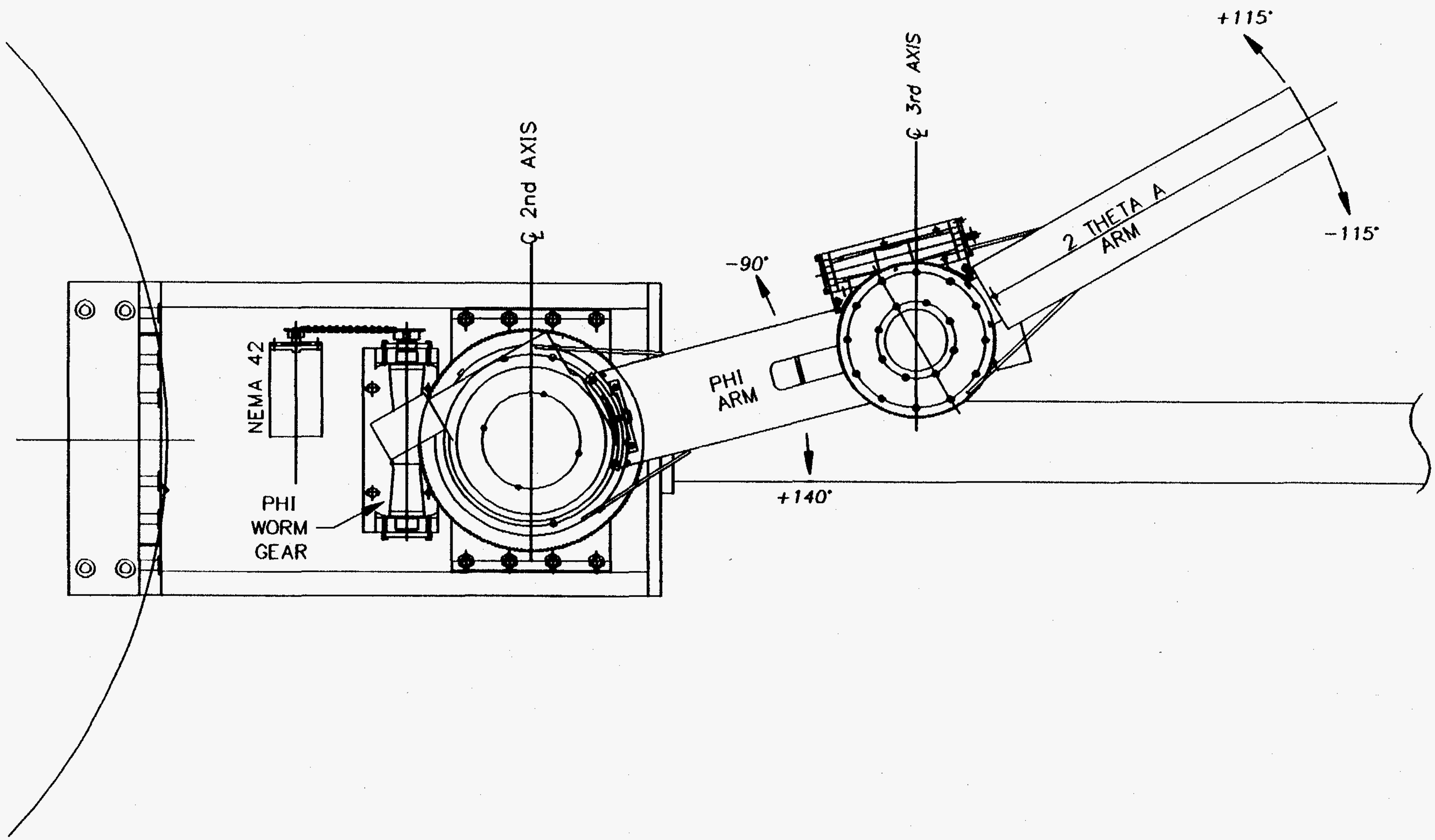


FIG. 2 TOP VIEW OF TRIPLE-AXIS SPECTROMETER

noticed before. A torque of 2000 oz-in proved adequate for handling the inertia, but was marginal at times in overcoming the frictional losses in the bearing supports and in the double enveloping worm gear drive mechanism. Direction changes often required several retries before the axis was within a tolerance of ±1 encoder count of the targeted move position. Because overshooting a move was costly, i.e. more time spent correcting the move instead of taking data, all moves were deliberately shortened by a very small amount. This prevented most overshoots, but occasionally required an additional move to get within the move tolerance. Clearly, something better was needed.

### 1.4 Lessons Learned From Investigation

The first upgraded system had three problems that needed correction: 1) the motor did not have enough torque to always overcome the frictional losses in the binding drive train, 2) the motor was operating near the primary resonance speed, and 3) the system had excessive backlash. A cost-effective way to mask some of the mechanical problems was to add a 3:1 planetary gearbox to allow higher motor speeds and a better torque margin. This would help with problems one and two, but not with the backlash. It would have been difficult to remount the PHI motor for direct drive of the worm. The new motion control system would have to accommodate this nonlinearity.

Another shortcoming of the first system was the slow rate at which the new 17-bit encoder data was read into the system. The data first entered the system through a counter in the CAMAC crate and then into the controlling Digital VAX computer over a SCSI bus. The FORTRAN code in the VAX did not have real time (fast enough) access to this data, preventing any form of closed-loop motor control. Thus, only post move corrections could be made to moves that did not reach the intended position.

### 1.5 Motivation for the Continued Use of Step Motors

The motion control needed for neutron science experiments often consists of the following; move, stop, count neutrons, and then repeat. This sequence is called a scan. Although several axes are sometimes moved simultaneously, it is still a simple repetitive motion task. Step motors fit this pattern quite well. It is also a very simple pattern; move a prescribed amount, and then remain motionless until the next move, which may not be for several minutes while data is being taken at this position. A servo motor was considered for the PHI axis because dual loop systems can be used to handle backlash. An inner loop is used to control motor velocity based on feedback from an incremental encoder or resolver on the motor, and the outer loop is closed on the position of the load. This was rejected for two reasons. Reason one, is the inability of standard servo hardware to read absolute encoder data. Reason two, is that only the final position of a move is of concern. Because of this, the motivation for using a servo system on just one axis was not strong enough to outweigh the added cost, complexity, or proliferation of motor technologies and the extra interfaces needed to implement a servo axis.

### 1.6 The Need for Custom Software

The inability to handle the 17-bit absolute encoder data proved to be the most difficult aspect of buying a complete turnkey servo or stepper motion control system. The question of whether absolute encoders are worth the effort necessary in order to handle their information must be answered for each unique application. In this case, the users wanted to keep the absolute encoders. The following system is based on a simple scheme that provided the advantages of the simplicity of open-loop stepper operation with the positioning speed and certainty of a closed-loop system.

## 2. NEW MOTION CONTROL SYSTEM DESIGN

The goal of this project was to design a better motion control system for a second spectrometer upgrade. After the second system was operational, the new control system would be retrofit to the first spectrometer, which already had new mechanical parts and encoders, to improve its performance.

### 2.1 Design Constraints

Several constraints existed from the start of the project. The only mechanical problem to be fixed was the binding of the worm gear set by providing an improved axis bearing pre-load design for the two arms. The proper operation of these bearings is critical to keeping the arc swept by these arms in a plane. The worm gear set has a very tight tolerance on alignment and run-out. Other than this change, the mechanical system design is almost identical to the first upgrade. The same step motors, drives, and absolute encoders were used on both systems (those had already been purchased when the first system was built). In addition to moving motors, data collection, reduction and graphical display, the FORTRAN code that runs the experiments is quite complex in all the tasks that it must handle. The VAX would act as a supervisory controller to the new system and it was desirable to integrate the new motion control system with as few changes to the present software as possible.

### 2.2 New Control Strategy

After studying the operation of the first upgraded spectrometer, it seemed possible that a PC based motion control system could be designed to improve the positioning accuracy without requiring major mechanical design changes or different motors or drives. One way to minimize the effects of backlash on the system would be to reduce the number of direction changes that the system makes. A way to eliminate some of these is to never allow the system to overshoot a move. If the motor is moving slowly enough, the arm can be stopped without overshooting to the next encoder count by stopping the pulse train to the motor when the encoder reading is at the correct position. The upper limit on this slow speed is governed by two separate issues. First, the position data must be sampled faster than the position is changing. Obviously, if this is not true, overshoots will be common. This rate is limited by hardware and software. The other rate-limiting factor is the friction in the arm itself. The friction drag must produce a deceleration larger than that of the motor, or the inertia of the arm will carry it into the uncertain region of mechanical backlash, potentially overshooting the move. This is true on PHI because it has backlash in the worm drive as well as the chain drive. Software implementation details are found in Sections 4.3 and 4.4.

### 2.3 Verification of the New Technique

This theory was tested experimentally on the first system, which was already equipped with the absolute encoders. Test programs were written, and two different indexer motor control cards were tested. Both cards could be programmed in such a way as to stop the move based on encoder feedback. Experience showed that almost any of the controller cards on the market could be made to work. Some would require more complex software than others. One of these cards allowed easy access to control registers that can be used to modify the classic trapezoidal move profile in such a way to ensure success without overshoot. Since all of these adjustments to the move can be made prior to starting the move, less code overhead is needed while motors are moving. The only time critical task remaining is to stop the move at the correct encoder count.

## 3. DESCRIPTION OF CONTROL SYSTEM HARDWARE

The control system is based on a single 19" rack mount industrial PC with a passive backplane. A single board 133 MHz Pentium computer occupies one slot. A Synchronous Serial Interface (SSI) encoder interface card occupies another ISA slot. This card makes reading the absolute encoders simple because a few lines of C code and a call to this function returns data for up to eight *17-bit* encoders. Two more ISA slots are used by the motion controller cards. Each card can handle eight axes of motion for a total of 16 axes. One more slot is occupied by a GPIB interface card. All communications between the VAX and PC take place over this bus.

### 3.1 VAX and PC Communication - GPIB Hardware Interface

The VAX 4000 computer that runs the spectrometer communicates to the CAMAC crate and NIM hardware through a SCSI connection. The CAMAC crate has a GPIB card that talks to various other hardware such as a cryostat temperature controller. Since this bus was already being used, the easiest way to establish a communications link between the PC motion controller and the VAX was to add a GPIB interface to the PC. This also made the new motor controller more independent from the VAX since any computer that can communicate using this standard interface can control the motors.

### 3.2 VAX and PC Communication - Software Interface Design

The present FORTRAN code runs the way it did before the new system was developed except that motion related commands are now forwarded from the CAMAC crate to the PC over the GPIB connection. A subset of a standard motion control language was implemented in software. Using such a motion control protocol allows the system to be used in other applications with similar requirements. The VAX interrogates the status of the command to determine if the command was understood and accepted without errors. The PC interprets the command and calls the appropriate function to carry out one of the following: 1) move the specified axis, 2) return the current position for the specified axis, or 3) if the specified axis is moving, the code returns a busy status.

The GPIB code and the main motor control code communicate through a form of shared memory implemented as a data structure. Data flow is unidirectional to prevent synchronization errors or read/write conflicts. If a change in state occurs from that which existed since the last check of the GPIB device, a status flag is set for the specified axis. This signals the PC code to take action on the new command. The command data is present in the same data structure as the status flags.

## 4. SYSTEM OPERATION

### 4.1 VAX Software

The VAX keeps track of all positions at all times. It calculates a target position based upon what type of scan it is running and the current position of the axis to be moved. The VAX then calculates the number of motor pulses needed to make the move and sends this number to the PC via the GPIB bus.

### 4.2 PC Software

The code on the PC is set to run at boot time in the autoexec.bat file. If a power failure occurs, the motor control system will be back on-line in several seconds. The code on the PC first initializes all of the hardware boards in its backplane. Setup parameters and degree conversion offsets for the motion control boards are read from disk files. The main loop is entered after all of the setup and initialization tasks have been completed successfully. As long as no keys are pressed on the keyboard, the infinite loop will continue to execute. During normal operation, the keyboard is not attached to the system so there is no chance for the code to stop unintentionally.

The first task in the loop is a call to the GPIB routine that handles the communication between computers by updating the shared data structures. The next thing that happens is a comparison of current and target positions for all axes with absolute encoders that are currently moving. If a comparison is true, that axis is stopped. Next, the positions of all axes with absolute encoders that are not moving are checked against their target positions. An automatic retry is started for any axis that has a position error unless the "try again" limit has been reached. The try again limit is described below in Section 4.4. The PC executes this loop every 0.166 ms or approximately 6000 times a second while motors are moving or stopped. Next, the registers that count incremental encoder pulses or pulses generated by the card for the axes that operate open loop are loaded into the shared data structure to make it available to the VAX. If the last call to the GPIB function returned an axis number, the status change flags are checked for this axis to determine what the VAX has requested. Valid requests or commands include an emergency stop, a degree conversion offset change, load new move distance, or start the new move.

The type of feedback available for an axis determines how the new move distance data is handled. If the command is for an axis without encoder feedback, the motor is moved that number of steps and the move is finished. If the axis has an incremental encoder the same thing is done, but the final position can be checked by reading the register that is counting encoder pulses. This is how step motors are normally operated.

### 4.3 The Modified Move Profile With Auto-Stop

If the move is for an axis that has an absolute encoder, the number of pulses is interpreted differently than previously described. First, this number is converted to the exact encoder value that represents the new desired position. Then a small number of extra pulses is added to the move distance. One hundred motor pulses are added for the PHI axis, which translates into 2 encoder counts. This adjustment guarantees that the target position will be reached or slightly exceeded in all cases except for a move that is in the opposite direction from the last move. However, this adjustment is much smaller than what is needed to compensate for backlash, which is handled differently. Function calls are then made to routines supplied by the manufacturer of the controller board to calculate the points that define the trapezoidal move profile. These points are calculated based on the values of parameters for the starting velocity (minimum), maximum velocity, acceleration, and deceleration which are pre-defined for each axis. Next, the ramp-down point, or the point at which the deceleration segment of the profile starts, is adjusted to occur sooner. It is adjusted by an amount equal to twice the number of extra pulses that was first added to the move (Fig. 3). This ensures that the motor is moving at the minimum velocity before the target position is reached. The control registers are then loaded with the adjusted move distance (number of steps) and the early deceleration point. Then the move is started. The motor moves and decelerates to the low velocity just before the original target position is reached. With no software intervention, the motor would continue slowly past the target position before stopping. The time needed to send out the remaining pulses at this velocity is less than 0.25 seconds. However, the code is in a continuous loop that compares the current position to the original target position. When the encoder reads the same value as the target position, the motor is stopped. There is no need for a move tolerance, since all moves are made to exact encoder positions.

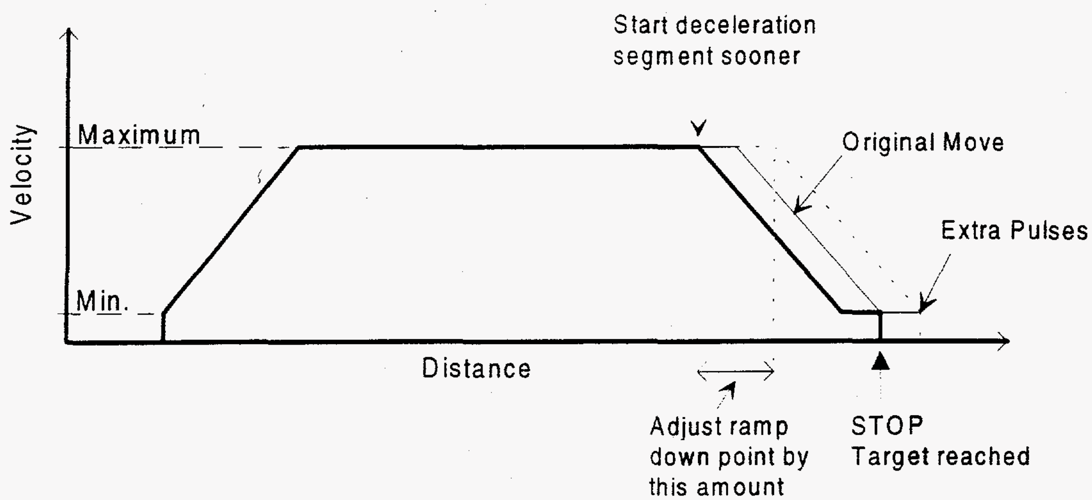


Fig. 3. Modified move profile

**4.4 Dealing With Errors and Backlash**

If the target position is not reached, which usually only happens when the direction is changed, another function is called that both loads another move based on the distance yet to be traveled and starts the motor again. Only four "retries" are allowed for each new move. If the target position is not reached after four tries, the move is terminated. The retries are automatic, and only a fraction of a second is needed to exhaust the limit of four tries. If the VAX checks the position of a moving motor, it will receive the busy status indicating that the motor is still moving. When all axes are at the correct positions, neutron data is collected.

For this control scheme to work well, the position data or loop cycle time needs to be maintained at some minimum rate. The slow speed at the end of the move is 500 motor pulses/second for the PHI axis. This causes the PHI encoder to change counts once every 0.1 seconds. The actions that take the most time in the software loop are the ones that involve direct communication to hardware, either the GPIB routines or the motor controller board. As mentioned above, the time needed to read the encoders is less than 0.0002 seconds. If the state of the GPIB device has changed, the main GPIB function adds time to this loop and slows it down by approximately a factor of four. The start of a new move takes even longer, about 2 ms. So this slows down the execution even more to about 500 Hz. However this is still much faster than the encoder data is changing (~10 Hz). The 500 motor pulses/seconds was determined experimentally as a value that allows the auto-stop feature to work without overshoots. As mentioned in Section 2.2 this limit is determined by the friction in the axis, momentum of the axis, and the deceleration rate of the stop command.

The code was run under DOS to maximize the bandwidth of the system. When the software is not interacting with hardware, Windows 3.1 reduces the speed of execution by a factor of two. This loss in performance is not enough to render the auto-stop feature unreliable, but Windows is not as deterministic as DOS. Therefore, the application was targeted for DOS.

## 5. SOFTWARE DEVELOPMENT

The software is written in C and is compiled using a popular C/C++ compiler. Development was done in the MS Windows 3.1 environment but was targeted as a 16-bit DOS application. Approximately 1500 lines of new code were written for this project. Additional code was borrowed and adapted from other projects to handle routine tasks such as disk file access. Setup files are used for as many parameters as possible. As a result, the code does not need to be recompiled to change any of the motor related parameters such as acceleration or maximum velocity.

## 6. PERFORMANCE

Of the four axes with absolute encoders, PHI and 2-Theta-A benefit the most from the real-time automatic stop feature. The difference between the arms and the turntables is strictly a mechanical one since they all contain identical encoders. The turntables are precision units with spring loaded anti-backlash drive mechanisms. Over 95% of the time, the tables rotate to the correct encoder count without having to adjust the move distance. When the motor has stopped, the data is checked and, if necessary, an additional move is made. Some manufacturers call this move at the end a "position maintenance" mode of operation. The turntables are operated in this mode because performance does not suffer appreciably with this low error rate. If an error exists, one small corrective move is able to eliminate it. This is how a good stepper system should operate. If the mechanical connection between the motor and load is accurate throughout its travel and backlash is minimized or can be accounted for, one can assume the load is positioned correctly without feedback. Because of high gear ratios, the inaccuracies of the step motor itself can be neglected. Positioning accuracy is therefore dominated by how accurately the mechanical system translates the rotary motion of the motor into a movement of the load. Since PHI and 2-Theta-A had loose mechanical coupling between the motor and the load, they benefited the most from the closed-loop, auto-stop, and auto-retry techniques described above.

### 6.1 Benefits of the New System

Moves are made without error, and the motor only starts and stops once for any move that is in the same direction as the last move. When a reverse in direction occurs, one additional corrective move is made. At a future maintenance opportunity, the software may be updated to handle direction changes by adding a sufficient number of motor steps to complete the move without the momentary stop before the current corrective move is started. A continuous display of status, target, and current positions is available on the PC monitor as another benefit of the new system. Previously a command had to be typed into the VAX to see the positions of stationary axes.

### 6.2 Comparison of Before and After Results

This section will describe the graphs of position versus time that highlight the performance of the old system and the new system. The first data set in Fig. 4 is a step scan from 45.0 to 45.6 degrees in 0.1 degree steps that was performed under VAX control. Several interesting observations can be made about the old system when looking at this data. None of the moves were correct initially. The moves to 45.4 and 45.6 were off by two and one counts respectively when the motor stopped, but drifted one count position before the next move was started. An error of one count exists at 45.1 and 45.3 degrees. An error of two counts exists at 45.2 degrees. At 45.5 degrees, the move tolerance of 0.0055 degrees (~two encoder counts) was exceeded and the axis was repositioned eliminating the error. These types of errors are not found in the data from the new PC system (Fig. 5). The next interesting observation occurs with the first move from 45.0 to 45.1 degrees, and is evident in both data sets. The phenomenon is related to the backlash in the drive train. Both moves were started at 5 seconds and the arm does not move until 6 seconds is reached. Most of the move is taken up by backlash. At approximately 7.7 seconds, the VAX starts a corrective move (Fig. 4). At approximately 6.4 seconds, at the end of the first move, the PC executes an auto retry (Fig. 5). Figures 6 and 7 show similar data in expanded detail. For some unknown reason, the VAX needed three separate moves to travel from 30.0 to 30.1 degrees. One can see that the backlash is approximately 0.1 degrees at the arm, approximately 1828 motor pulses, or five encoder counts. Figure 8 shows the amount of backlash more clearly.

Figure 9 displays the ability of the PC system to hold position after the move has completely stopped. After the move stopped at 4.4 seconds, the arm was manually pushed in the opposite direction from which it was traveling. When the force was removed, the arm overshot the target position, and the PC had to reverse the motor, take up the slack, and make a tiny move to eliminate the error. Now for the next move to be made correctly, the PC must be able to contend with the backlash in a robust way. The backlash is now somewhere in between 0.0 and 0.1 degrees. This is because a normal smooth auto-stop was not the last move made before this data was take. A similar manual disturbance was made in the move prior to the move shown in Fig. 9. This can be seen in the correction that was needed to originally reach 42.0 degrees.

## 7. SUMMARY

A limited form of closed-loop operation of step motors was successfully demonstrated in this project. A closed-loop system would normally refer to a system that generates a path and then uses feedback to continuously vary the torque or speed of the motor to move the load in such a way as to track this path with as small a deviation as possible. Step motors and the indexers that drive them do not except error feedback signals of this sort. A servo motor would be a good choice for a system with such demanding requirements. Because this application did not require precise control of the total path, but only the ending destination, a fully closed-loop system in the traditional sense was not needed. The problem of how to deal with the variable backlash of the system required a robust design that could handle moves for which the exact number of pulses needed is unknown before the move is started. The modified trapezoidal profile allowed a simple way for the code in the PC to control the final destination of the move without undue computational overhead that full closed-loop operation normally requires.

## 8. ACKNOWLEDGMENTS

The authors would like to thank S. King and D. McMillan for software and design input, and J. Hawk for system wiring details. We would also like to acknowledge the fine support received from the Instrumentation & Controls, Solid State, and Plant & Equipment Divisions for fabrication and installation of the new systems.

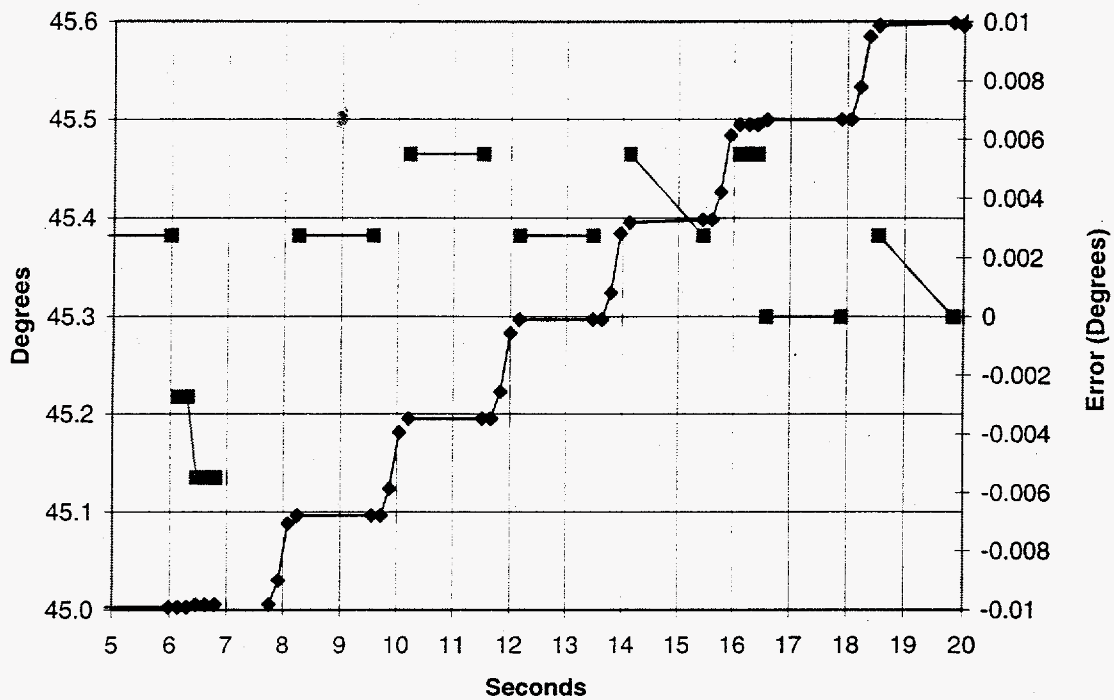


**Fig. 4. VAX Step scan with direction change on first move**

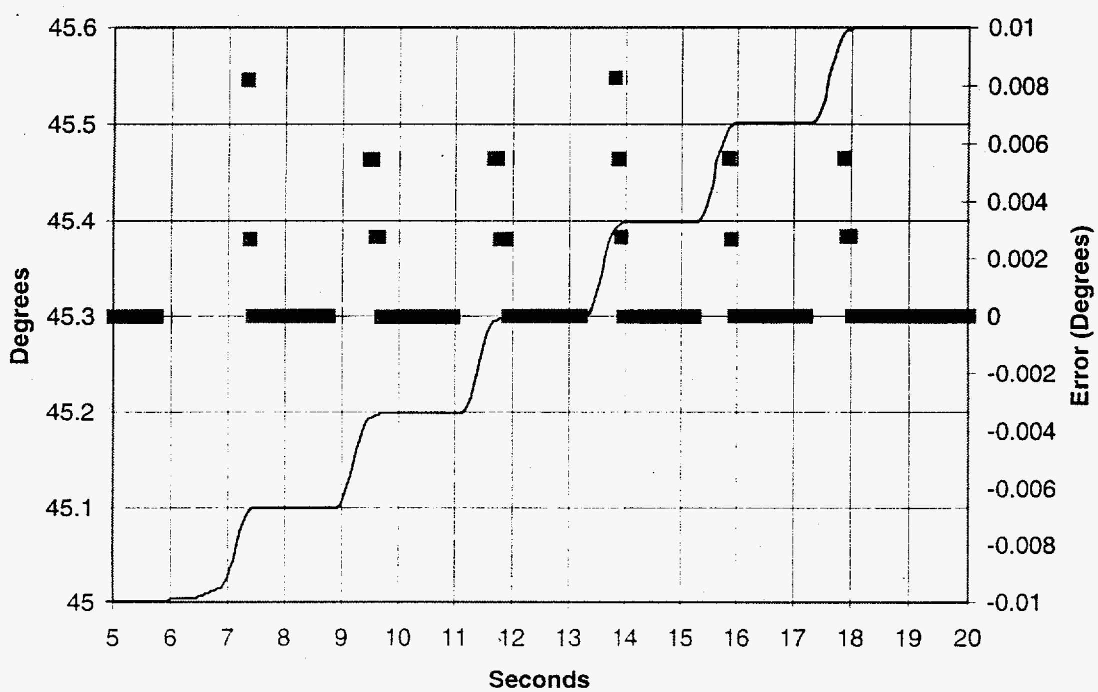


**Fig. 5. PC Step scan with direction change on first move**

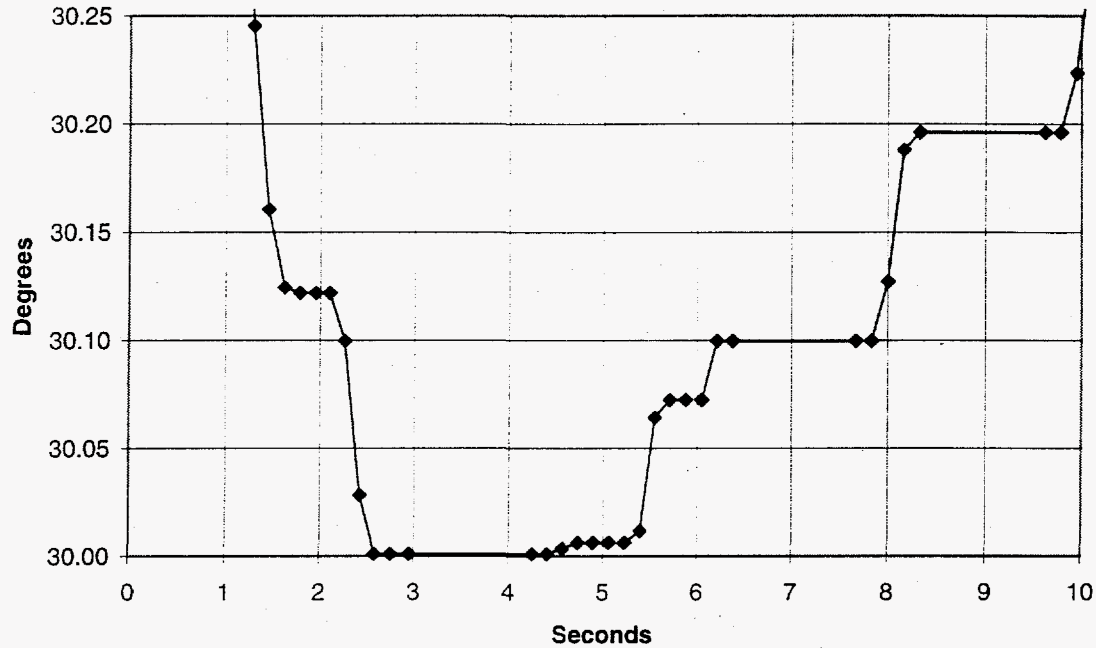


**Fig. 6. VAX Beginning of scan with two direction changes**

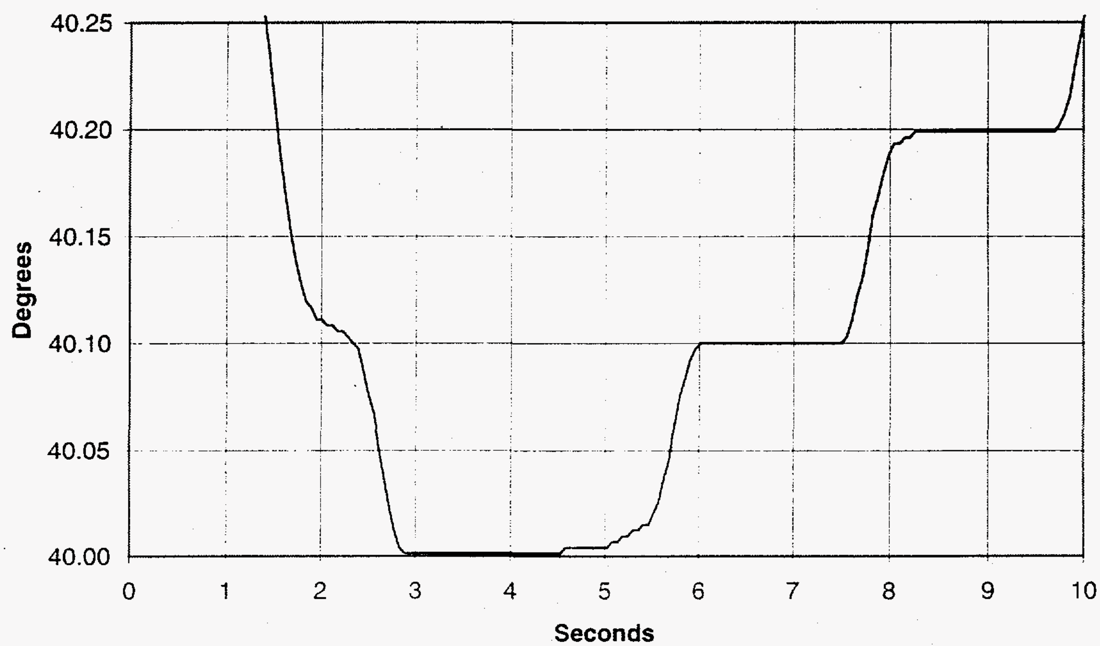


**Fig. 7. PC Beginning of scan with two direction changes**

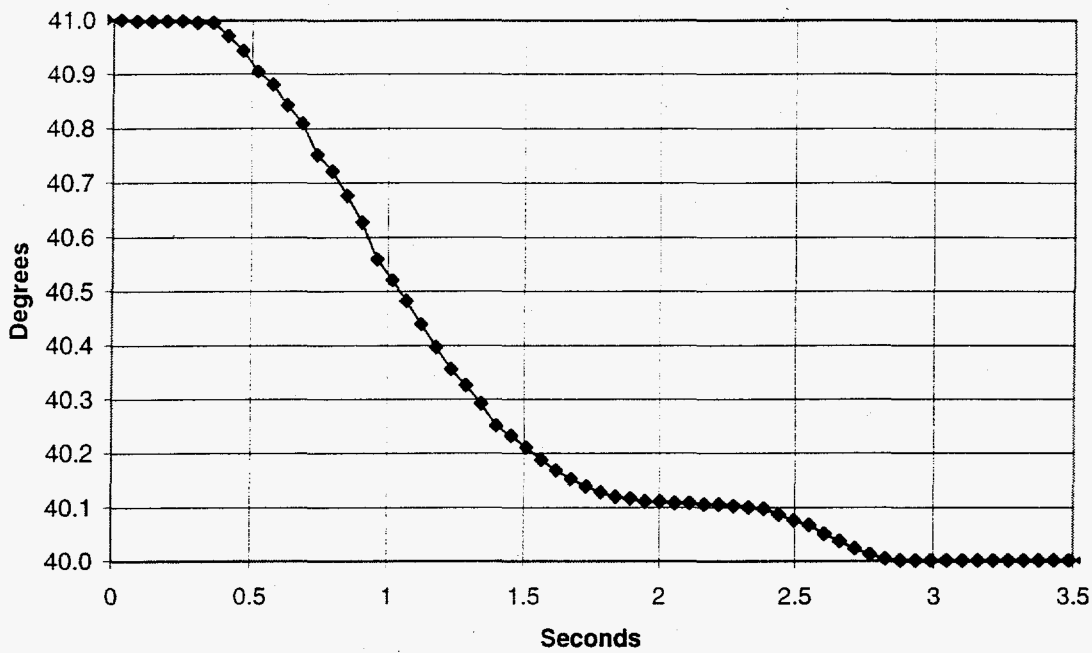


**Fig. 8. PC 1 Deg. move with direction change illustrating 0.1 deg. backlash**

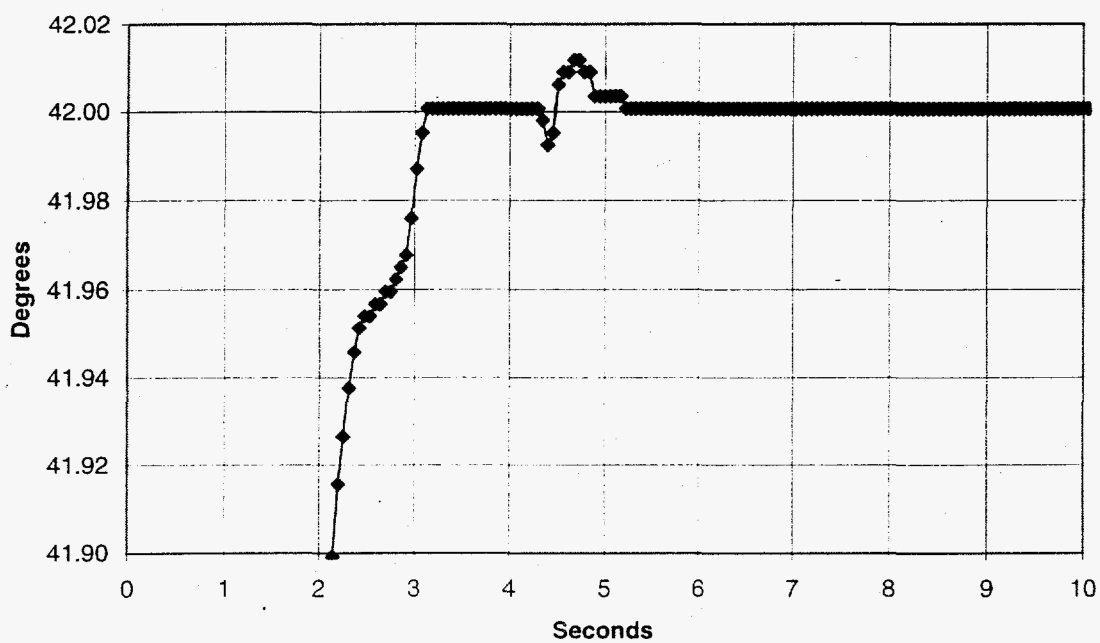


**Fig. 9. "Position Maintenance" after disturbance at 4.4 sec**