- Coarse Actuation System -
Control System Description

HYTEC, Inc.
March 3, 1998

This document summarizes the control logic and requirements for the coarse actuation control system used on the LIGO program. All hardware and software issues will be presented and covered in this document. This document applies to both BSC and HAM control systems.

PREPARED BY: _____

Greg Hayman

_____

Erik A. Swensen

PROJECT MANAGER _____

Tim Thompson

HYTEC Inc.
110 Eastgate Dr., Ste 100
Los Alamos, NM 87544
(505) 661-3000 / Fax (505) 662-5179

LIGO PROJECT

Table of Contents

1. Introduction

The coarse actuation system (CAS) control system has progressed a great deal since PDR. A number of "Value Added" changes have improved the system. Most of these changes were presented again in the January Mini-Review. This review covered the details of the CAS as it stood at this time, however, there was again some concern regarding the level of detail presented to implement and protect the control system logic. This report has been written to describe the control system in greater detail. The document is broken down into a systems level description and a component level description of both the hardware and software that will be used to control the CAS. The control system presented in this document will be implemented on both the BSC and HAM platforms.

2. Control System Requirements

See the LIGO technical specification LIGO_TS_02 (Attachment A) for details of the coarse actuation system requirements.

3. Coarse Actuation System Design Selection

HYTEC has identified a coarse actuation system that will meet or exceed the LIGO requirements specified in section 2. An eight axis electro-mechanical system has been proposed and designed. This system uses mechanical actuators to move the SEI platform and position it in six degrees of freedom (DoF's). The system will use four air bearings, one at each pier, each configured with a spherical and linear seat to provide nearly frictionless motion in the horizontal plane and all three rotational DoF's. Two linear translation tables will be used in each of the horizontal directions to provide horizontal translations and four linear actuators will be used to provide vertical translations. The eight actuators will also be capable of providing coupled translations to satisfy the platform rotational requirements of the system.

The system components identified to perform the required motions have not changed since PDR, however there have been changes to the controller. The control system that was presented at PDR was a stand-alone system that required the use of both a DAEDAL COMPUMOTOR 4000 and an EASON 1100 to accomplish what a single bus based controller could accomplish with much greater ease and efficiency. It was decided that an 8-axis GALIL DMC-1780 ISA-bus controller would be used to replace the old hardware. Taking advantage of this technology substantially reduced the overall cost of the system.

This is a very general description of the CAS, however the scope of this document is intended to describe the advancement of the control system, for the CAS, since PDR and the January review. If the reader would like a more detailed description of the CAS hardware components, refer to the PDR document, HYTEC-LIGO-002. The remainder of this document will assume some prior knowledge of the CAS components and focus attention on the control system details.

4. Control System – Systems Level Description

This section has been written to provide a general overview of the control system design from a systems level and not a component level. Sections 5 and 6 will describe the hardware and software, respectively, on a component level.

The discussion presented in this document is based on a State-Transition methodology to identify both the state of the system and the transition required to pass from one state to another. The state of the system can be described as a "snapshot" in time as the control system sits in a known and predefined configuration, awaiting the next control command signal. The transitions are defined as the sequence of events that must take place to move the system between states. The entire control system will be broken down and discussed in these terms.

4.1 System State Diagram

Figure 4.1.1 shows the State-Transition Diagram for the CAS. This diagram illustrates the five states of the CAS and the logical transitions between states. Notice that the CAS can not arbitrarily move from one state to another, a logical sequence must be maintained. These transitions are produced by the logical sequence of events that must occur to move between states. Three different transition lines are used to identify who is ultimately responsible for initiating the transition (i.e. the operator or controller or either). The following discussions will reflect, in detail, on this diagram.

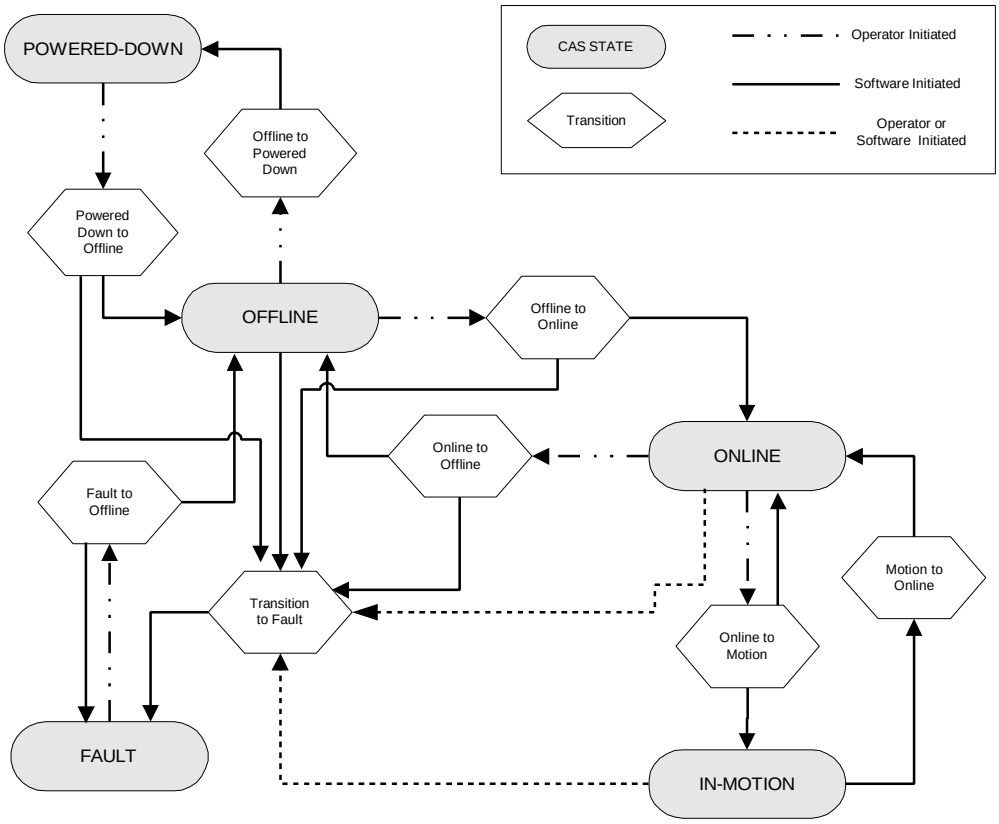


Figure 1. CAS State-Transition Diagram

4.2 System States

This section describes the control system as it exists in each predefined state. The term "state" implies that the control system has already executed a number of steps to reach the defined state. This section describes the condition of the system as it exists in each state.

4.2.1 POWERED-DOWN State

While POWERED-DOWN, all of the CAS-related electronics are powered-down and the pneumatic solenoid valve is closed, thereby maintaining the air bearings in a de-energized* condition.

> Note: *It has been assumed that the host computer will also be powered-down in this state. We need feedback from CALTEC to determine if this is the desired status of the host computer, or is it more desirable to have the host computer on during all facility operations?*

> **Note: The term "de-energized" (or similar) will be used throughout this document to refer to the state of the air bearings when the airflow is off and an air film does not exist (pressure = 0 psi).*

4.2.2 OFFLINE State

While OFFLINE, the host computer, controller and all sensors are receiving power and able to perform operator-requested commands. The control software is monitoring the health and status of the system. The operator will receive a status message that indicates the state of the system and will be able to query the health and status information received by the sensors. The air bearings will continue to be de-energized and the actuators will be held in a stationary position. The actuators will not be capable of executing a motion command from the controller.

The system can be changed from OFFLINE to POWERED-DOWN, ONLINE, or FAULT. The system will return to POWERED-DOWN if initiated by the operator. The system can proceed to ONLINE if initiated by the operator. The system will be sent to FAULT if the control software detects a faulted condition while OFFLINE or during the transition to ONLINE. These transitions will be discussed in greater detail later.

4.2.3 ONLINE State

While ONLINE, the host computer, controller and sensors will continue to function as they did in the OFFLINE state. The motor amplifiers will be enabled and the actuators will be placed in an idle state awaiting a motion command signal from the controller. The actuators will be held in their original position until a motion signal is sent. The pneumatic solenoid valve will be in the open position, which will allow all four air bearings to be energized*. The operator will receive a status message that will indicate that the system is online and awaiting a command.

The system can be changed from ONLINE to IN-MOTION, OFFLINE, or FAULT. The system will return to OFFLINE if commanded by the operator. The system will be sent to IN-MOTION through an operator commanded motion. The system will be sent to FAULT if the control software or operator detect a faulted condition while ONLINE or during the transition to IN-MOTION or OFFLINE.

> *Note: The term "energized" (or similar) will be used throughout this document to refer to the state of the air bearings when the airflow is on and an air film exists. It will also imply that the air pressure is between the upper and lower pressure limit requirements.*

4.2.4 IN-MOTION State

While IN-MOTION, ONLINE will continue to be satisfied with the exception that one or more actuators will physically be in motion. The operator will receive a status message that tells them that the system is making a commanded motion. The operator will also be able to verify the motion execution by viewing the position feedback.

The system can be changed from IN-MOTION to ONLINE or FAULT. The system will return to ONLINE automatically through the control software after the commanded motion has successfully been achieved. The system will be sent to FAULT if the control software detects a faulted condition while operating IN-MOTION.

4.2.5 FAULT State

While in FAULT, the system will remain at rest in a known and predefined state awaiting a system reset by the operator. The faulted state will leave CAS with the air bearing de-energized and the actuators incapable of providing motion, but will hold the actuators in their current position. To accomplish this the control software will close the pneumatic solenoid to de-energize the air bearings after the motion has been settled. The power will be disconnected from the actuators. The system will now be in a known state until such time as the operator resets the fault condition. The operator will receive a status message that will tell the operator that a FAULT state is present and notify the operator of the type of fault and possible errors that may exist. The operator will need to clear the error, which will return the system to the OFFLINE state. Depending on the fault condition, this could be a simple reset by the operator on the control software, or this may require a technician to physically correct a mechanical fault.

The state of the system can only be changed from the FAULT state to the OFFLINE state assuming that the faulted condition is no longer detected by the control software. The operator may not be able to reset the fault if the faulted condition has not been corrected and the control software will return the system to a FAULT state.

4.3 System Transitions

The system transitions describe the logical sequence of events that must take place in order for the control system to transition to another state. This information will become the functional logic required to satisfy the control system needs. Mechanical interlocks

and operational procedures have been derived from these transition descriptions and the control logic, both hardware and software, will evolve from this information as well.

4.3.1 POWERED-DOWN to OFFLINE

Here, the power supplies will be turned on to enable the host computer, controller and sensors. The controller will begin monitoring the health and status of the system. Sensor readings will be taken and compared to previous data to check for drift. A large drift value (TBD) will send the system to FAULT.

4.3.2 OFFLINE to POWERED-DOWN

Sensor readings will be taken and stored for the next startup sequence. The host computer will need to be properly shutdown. The power supplies used to power the control system will be shutdown.

4.3.3 OFFLINE to FAULT

If the controller detects one or more fault conditions it will perform the required steps to place the system in a FAULT. This will include de-energizing the air bearings and disabling the actuator motors.

4.3.4 OFFLINE to ONLINE

Here, the controller will enable the motor drive that will hold the actuators in their current positions. The air bearings will be energized and the system will verify that the pressure is within the required limits and the emergency stop buttons have not been activated. With these conditions satisfied, the system will mechanically close a circuit to allow the drive to perform a motion command on the actuators. If any of these conditions are not satisfied, the system will go to FAULT.

4.3.5 ONLINE to OFFLINE

The transition to OFFLINE will easily be achieved by reversing the process described in section 4.3.4. The only precaution is to ensure that the system has come to a complete stop before the air bearings are de-energized and the surfaces are seated. Improper care could result in damage to the air bearing surfaces. If any conditions exist that are not expected, the system will go to FAULT.

4.3.6 ONLINE to FAULT

A number of mechanical or software based fault conditions exist that could send the system to a FAULT state. The hardware or software can send the system into a FAULT state, however the software will perform the appropriate functions to properly transition. This will include de-energizing the air bearings and powering down the motors. The operator will be notified of the fault and possible errors.

4.3.7 ONLINE to IN-MOTION

The controller will initiate the commanded motion and pass the electrical information to the motor drives. The drives will supply sufficient power to the stepper motors and begin moving the actuators. The controller will monitor the health and status of the system and notify the operator of any changes.

4.3.8 IN-MOTION to ONLINE

The controller will discontinue the commanded motion when the system has reached the commanded position. The motors will return to a rest condition and hold the actuators in their current position. The controller will return to an idle position awaiting the next command. The controller will continue to monitor the health and status of the system.

4.3.9 IN-MOTION to FAULT

The CAS will be taken directly to the FAULT state if a fault condition has been detected. Again, the air bearings will be de-energized and the motors will be disconnected and held in their current position. The controller will send a status message to the operator.

4.3.10 FAULT to OFFLINE

The FAULT to OFFLINE transition can only take place when the fault has been cleared. This can only be done by physically correcting the problem and resetting the control system. The operator will be able to clear the fault condition, however if the fault has not been corrected, the system will return to the FAULT state.

5. Hardware – Component Level Description

The control system has undergone a number of design changes since PDR. These changes have resulted in both a more robust system and a lower cost to the system. The control system that was developed through all of this work will be illustrated and the hardware logic will be described as an expansion to section 4. This section is intended to provide an overview of the hardware side of the control logic and will be expanded to encompass all of the electro-mechanical and electro-pneumatic components that are currently being used in the control system.

5.1 Control Logic Overview

HYTEC is working to develop a robust control system to perform the CAS operations and meet the CAS requirements. The design accomplishes this goal while minimizing both cost and downtime to the system, in the unlikely event of a catastrophic failure to one or more components in the CAS. In addition, the cost of the system has been reduced from the originally quotes presented in the cost books.

It is impossible to talk about the hardware without referring to the software and vice versa. The control system can only function as intended if both elements are successfully performing their respective operations as required. There is, however, a distinction between the function of the software and that of the hardware. This distinction does have some gray areas in which the software and the hardware are performing parallel operations. This control system has been designed with some parallel performance by the hardware and software and the following paragraph will answer the question why.

The CAS has a number of components that are both expensive and require a long lead-time to fabricate. In particular, the air bearings are an expensive item that would require

several months to fabricate. The downtime to the LIGO facility will be costly during this period and the loss to the scientific community could be devastating. The design team has decided that it would be best to add mechanical interlock circuits to protect the high priced items of the control hardware. The mechanical circuits will perform parallel operations to the control software up until the point at which it is no longer protecting the mechanical hardware and simply mirroring the control software actions. The electro-mechanical circuits will perform many of the steps to place the system in the FAULT state, however the control software will be required to complete the transition.

The next concern is the failure of one or more of the electro-mechanical/pneumatic components used to protect the system. It is not conceived that the likely-hood of this is great, however the possibility does exist. Given the above scenario, each electro-mechanical/pneumatic component has been set in the appropriate configuration to protect the system from damage. This means that the electro-mechanical switch will fail such that the circuit is no longer conducting or the electro-pneumatic switch will no longer allow airflow. For example, the air solenoid has been spec'd-out to be normally closed, this means that the airflow will be cutoff if the solenoid no longer functions properly.

5.2 Control System Schematic

The COARSE ACTUATION CONTROL SYSTEM LOGICAL FLOW DIAGRAM can be found in Attachment B.

The COARSE ACTUATION CONTROL SYSTEM LOGICAL FLOW DIAGRAM can be interpreted as a means of illustrating the electro-mechanical and electro-pneumatic logical flow of the control system. All of the hardware used in the control system is represented. It is worth noting that this diagram is not a wiring diagram nor is it meant to replace the wiring diagram. HYTEC will begin work on the wiring diagram and rack layout once we have had favorable comments from CALTEC regarding this document.

The components shown on this diagram have been arranged to represent the physical location near any given chamber. The LIGO/HYTEC interface has been shown to distinguish the logical cutoff of HYTEC responsibilities. The components that will be physically located within the rack are boxed within a dashed line. The actuators and sensors located to the right of the rack will be located on the appropriate piers. The components shown on the bottom of the page are the electro-pneumatic circuitry. The components between the interface and the manifold will be housed in an electrical box located on one of the piers for each respective chamber. The electro-pneumatics to the right of the manifold will be located nearest the air bearing it is required to monitor. The table in the lower right corner of the diagram breaks out and identifies the pin connections of the GALIL Controller and lists how HYTEC is intending to use each pin. Several pin connections have been left open and are currently available for other uses if desired and possible.

5.3 Hardware States

This section will describe the state or condition of the hardware as it exists for the states described in section 4.2.

5.3.1 POWERED-DOWN State

While POWERED-DOWN, all of the CAS-related electronics are completely powered-down. The host computer and local power supplies are turned off. The pneumatic solenoid valve is closed thereby maintaining the air bearings in a de-energized condition.

It has been assumed by HYTEC, that the host computer would be turned off to reduce both power consumption and electrical noise that may exist, especially within the racks that will house the fine actuator control hardware. This may not be the desired state and will need to be clearly noted by CALTEC. If this is the desired state, HYTEC will need to know how CDS or the operator would like to initiate the startup of the host computer. This can be performed with a switch located on the rack but would require the operator to be local, or the operator could send a command signal that will close a relay switch and allow AC power to be supplied to the host computer and local power supplies.

5.3.2 OFFLINE State

While OFFLINE, the local power supplies are powered-up and boots up the host computer. The controller and all of the sensors are now powered and sending or receiving electrical signals. An electrical signal is sent to the motor amplifier (NextStep Drive) through the shutdown lead that disables the drive (a high signal indicates amp disabled). The mechanical actuators are held in a stationary position by the mechanical friction of the system and an electrical brake[*]. The air bearings remain de-energized and are being held by the flexural pivot assembly and the friction between the bearing surfaces. During this state, the actuators will not be capable of receiving and executing a step command from the control software. This is true because a solid state relay holds the step lead in a normally open position (electro-mechanical interlock) in addition to the amplifier being turned off (controlled by the control software).

[*] *<u>The electrical brake</u> is a circuit that has been designed to short the motor windings when the motor amplifier is disabled. The back emf of the motor will produce a holding torque that is order(s) of magnitude higher than the detent torque (that produced by the magnet) of the motor. The total torque supplied by this circuit was measured to be ~5 in-lb for the NEMA 34 motor. This can be compared to the detent torque, which was not measurable. The holding torque of the enabled amplifier was verified to be 25 in-lb with 3.5 amps applied by the amplifier.*

This is a simple circuit to build and is commonly used in similar applications. HYTEC included the electrical brake as a result of the "Value Added" engineering that took place after the CDR and again after PDR. The mechanical brakes were removed (CDR) and the mechanical

actuators were downsized (PDR) as a result of this engineering effort. This has left the system in jeopardy of exceeding the backdrive limit of the actuators (HAM vertical actuators in particular). An electrical brake is required to prevent the actuators from backdriving because the amplifier supplied holding torque will be removed when the NextStep drive is disabled.

This circuit has been added to the control system and is planned because of analytically determined limitations of the system components. HYTEC intends to verify the benefit to the system by adding this circuit and validate its performance in the overall scheme. There are also plans to test the backdrive limits and load carrying capabilities of the actuators. The actuator testing and system performance validation will determine the fate of the electrical brake. If it turns out that the actuators will be able to carry the loads required, the electrical brake will be removed. Until this time, it will be incorporated to ensure its performance.

5.3.3 ONLINE State

While ONLINE, the host computer, controller and sensors will remain powered-on and performing operations as they did OFFLINE. The amplifier shutdown leads will be low which enables the NextStep drive. When the NextStep drive is enabled, the electrical brake is removed and the drive provides a holding torque to the motors, holding the actuators in position. The emergency stop buttons are all closed (no fault), the pneumatic solenoid valve is open, the air pressure is between the proper limits (~80 to 150 psi) and all of the air bearings are energized (floating). Because this is a no-fault state, the electro-mechanical checks indicate a status OK and the amplifier step leads will be closed. The system is ready to receive a motion command from the control software.

5.3.4 IN-MOTION State

While IN-MOTION, the conditions of ONLINE will continue to be satisfied with one exception. One or more linear translation tables and/or linear actuators will physically be in motion. The position sensors will be returning a signal proportional to the motion.

This state can only remain true as long as the air bearings and/or actuators are within the physical limitations of both the mechanical stops and the limit switches on the actuators. The control software can send the system into a FAULT when a limit switch is activated, however the controller will not be aware that a mechanical stop has been reached other than through the position sensors. The control software will need to sample the position to identify a mechanical limit or bind-up in the system.

5.3.5 FAULT State

The electro-mechanical system has been designed to place the CAS in a know state that closely matches that of the control software. However, a fault is unable to bring the CAS

to a distinct state. There are two distinct states that the system and controller must account for. The two states are described below:

5.3.5.1 Type I FAULT

This state will occur if an emergency switch has been opened, a system low-pressure or high-pressure condition has been detected.

The command motion to the actuators will be interrupted because the OK lead will be low and the amplifier step lead SSR's will be open. The holding current in the NextStep drive will hold the motors in their current position. The pneumatic solenoid valve will be closed and the air bearings will be de-energized. The controller will put the electro-mechanical system in a state that is identical to the OFFLINE state. The controller, however, will know that the system is in a FAULT state and monitor the condition until it has been corrected.

5.3.5.2 Type II FAULT

This state will occur if a low-pressure condition is detected near any of the four air bearings.

The command motion to the actuators will be interrupted because the OK lead will be low and the amplifier step lead SSR's will be open. The holding current in the NextStep drive will hold the motors in their current position. The pneumatic solenoid valve will remain open and the air bearings will remain energized. The controller will put the electro-mechanical system in a state that is identical to OFFLINE, sending a signal to close the solenoid valve and de-energize the air bearings. The controller, however, will know that the system is in a FAULT and monitor the condition until it has been corrected.

5.4 Hardware Transitions

The transitions represent the steps or sequence of events in both the software and hardware that occur in order to change the system from one state to another. These sequences of events are actually controlled by software commands that are processed by the controller and electro-mechanical hardware. This section describes the sequence of events that will take place to transition the hardware components from one state to another. One should always remember that the software will initiate the events.

5.4.1 POWERED-DOWN to OFFLINE

Here, the power supply will be turned on either manually or via a mechanical relay. The power will turn on the host computer and local power supplies. The controller and sensors will be powered and a signal will be sent to the amplifiers through the shutdown leads. The sensor information could send the system to FAULT.

5.4.2 OFFLINE to POWERED-DOWN

The control software will properly shutdown the computer and the mechanical relay will be switched to the normally open position such that the host computer and CAS electronics.

5.4.3 OFFLINE to FAULT

The software will control the transition from OFFLINE to FAULT. The electro-mechanical components will not process a sequence of events to make the transition, however an electro-mechanical/pneumatic switch may be the initiator of the faulted condition.

5.4.4 OFFLINE to ONLINE

The control software will initiate the transition sequence. The GALIL controller will begin the transition by removing the amplifier shutdown signals (current low). The low signal on the amplifier shutdown lead will enable the NextStep drive. The drive is now capable of providing a holding torque to the motor, however, the amp-to-motor leads will be open until the mechanical relays used to connect/disconnect the electrical brake have been repositioned to allow current to flow from the NextStep drive to the motor. The amplifier holding torque will maintain the current position of the actuators once the relay has been switched.

Although the drive has been enabled, there is a solid-state relay (SSR) on the step lead that is normally open to protect the air bearings from accidentally being forced to move without proper bearing pressure. The sequence of events that will ultimately close the amplifier step relay are described in the following paragraphs.

The controller will send out a signal to the pneumatic circuit to initiate the startup sequence that will energize the air bearings. The signal will be sent on the OK/pneumatic solenoid lead. The OK/pneumatic solenoid lead is used as both a mechanical summation to check the mechanical interlocks and close the amplifier step lead if the air bearings have been properly energized and as a toggle to open or close the pneumatic solenoid valve.

The signal will be sent by the controller and will begin the transition by verifying that the emergency stop buttons are all closed (check #1). The system will transition to FAULT if one button has been activated. Three buttons are being used to ensure that an emergency button will always be accessible to a local technician. The rack will house one emergency button, and the chamber will house two to prevent the possibility of a technician requiring an emergency stop and finding that it is located on the opposite side of the beam tube (oops!).

The command signal will be split at this point to open the pneumatic solenoid valve and continue along the OK/pneumatic solenoid lead. The pneumatic solenoid valve will open after a ~1 second delay. This delay has been included to account for any inertial effects if the system is aborted during a motion sequence. The air bearings will de-energize quickly

and the residual motions may cause two air bearing surfaces to slide across one another, resulting in damage to the air bearing surfaces. This will need to be tested to determine how much inertial energy is stored in the system during a commanded abort and how much time is actually required to damp out this motion. It may turn out that the mechanical delay of the solenoid valve is sufficiently long enough to complete this task.

A low-pressure switch is used in front of the pneumatic solenoid valve to ensure that air is at a minimum required pressure. This pressure switch will be able to close the pneumatic solenoid valve in the event that the LIGO air supply is not at the proper pressure because of improper bottle or compressor output. It will be set slightly higher than the low-pressure requirements of the air bearings to account for losses through the length of the tubing. The low-pressure switch will send a signal to a SSR and close the OK/pneumatic solenoid lead (check #2). It may be best to place the low-pressure switch in-line before the regulator and change the pressure setting to something higher than the required pressure, if a bottle is being used. This will allow the system ample time to detect a low-pressure condition if the bottle were running low, and allow the operator or software to shutdown the system long before damage might occur.

A high-pressure switch has been placed in-line with the system to ensure that the regulator is functioning properly. High-pressure can result in air bearing damage (vendor specified), as well, and should be protected against. The high-pressure switch will activate a mechanical interrupt, again after a short delay. The mechanical interrupt is used because we could inevitably be caught in a mechanical loop, oscillating between a high-pressure fault and a pressure satisfied condition, when the valve is closed for the high-pressure condition and opened for the pressure satisfied condition. The mechanical interrupt will prevent this from occurring. The high-pressure switch will send a signal to a SSR if the pressure is below the max pressure, closing the OK/pneumatic solenoid lead (check #3).

The Logic diagram is somewhat misleading in its representation of the distance between the pneumatic manifold and the individual air bearings. This distance can be as much as 20 feet and anything can happen to the pneumatic plumbing over this distance. Therefore, low-pressure switches have been placed immediately before each air bearing orifice. Each of the four pressure conditions are monitored and must be satisfied in order to close the final SSR on the OK/pneumatic solenoid lead (check #4). This SSR will remain normally open until the pressure in all four air bearings is satisfied. A single low-pressure condition will disallow the sequence and send the system into a FAULT state.

If all of the emergency stops and pressure conditions have been satisfied, their SSR's will be closed, allowing the OK signal to close the SSR located on the amplifier step leads. The emergency stops, system low-pressure, bearing low-pressures (#'s 1-4) and the high-pressure o.k. conditions are sent back to the controller for monitoring. The system is now ONLINE and can receive a commanded motion signal.

5.4.5 ONLINE to OFFLINE

The sequence of events in taking the system to OFFLINE is not nearly as involved as bringing the system ONLINE. It is essential that the system has come to a complete stop (inertial energy is dissipated) before de-energizing the air bearings. The holding torque can be released after the air bearings are de-energized.

To summarize the transition, the pneumatic solenoid valve signal will be disconnected thereby closing the valve. This will de-energize the air bearings. The OK/pneumatic solenoid lead will go low, which opens the step SSR. The amplifier shutdown lead signal is sent high, which disables the NextStep drive. The high signal in the amplifier shutdown lead also will discharge the mechanical relay on the motor winding leads (electrical brake). The motor winding leads are now shorted and provide additional holding torque.

5.4.6 ONLINE to FAULT

The system will be sent to FAULT if any of the mechanical interlocks described in the OFFLINE to ONLINE transition are no longer satisfied after the system has been brought to ONLINE. The system will be mechanically sent to one of the FAULT states described above through the use of the pneumatic circuitry described previously.

An emergency stop, system low-pressure or high-pressure condition will initiate a Type I FAULT. The OK/pneumatic solenoid lead will go low and open the step SSR. The motors/actuators will be held with the holding torque supplied by the NextStep drives. The air bearings will be de-energized after some delay to ensure that the system inertia has been damped out and protect the bearing surfaces from rubbing damage. The controller will place the system in a know/preset state described in the system state descriptions.

A low-pressure condition that is detected near the air bearings will initiate a Type II FAULT state. The air bearings will remain energized. The OK/pneumatic solenoid lead will go low and open the step SSR. The motor/actuators will be held with the amp supplied current. The air bearings can only be discharged at this time by the controller. Again, this action will be delayed until the system has come to rest.

5.4.7 ONLINE to IN-MOTION

This transition is easy from a hardware standpoint. The controller will initiate the commanded step and direction and pass the electrical information to the NextStep drives. The appropriate drives will supply sufficient power to the stepper motors and begin moving the actuators. The state is now IN-MOTION.

5.4.8 IN-MOTION to ONLINE

Again, the transition is easy from a hardware standpoint. The electrical signal that has been sent be the controller to provide step and direction information will be discontinued. This will force the motors to discontinue actuator motions and cause the NextStep drive to hold the motors in their current positions.

5.4.9 IN-MOTION to FAULT

The electro-mechanical/pneumatic system will not recognize any additional faulted conditions IN-MOTION that are found ONLINE. Therefore, the transitions between IN-MOTION to FAULT will match that described for the ONLINE to FAULT transition described in section 5.4.6.

5.4.10 FAULT to OFFLINE

The FAULT to OFFLINE transition can only take place when the fault is cleared. The controller will clear the fault and return the system to OFFLINE. However, if the mechanical fault still exists, it will have to be corrected by a technician before the system can be transitioned to OFFLINE.

6. Software

The role of the software is to establish a systematic approach to controlling the CAS. The software must monitor the CAS hardware and be capable of accepting input from an operator. The software design is based upon the state and state transition logic of sections 4.2, 4.3, 5.3, and 5.4. Parallel descriptions of these states and transitions are discussed in terms of the control software and its duties.

6.1 External Interface

The software interacts with the external environment through two forms of external interfaces. The first is a set of hardware inputs and outputs from the CAS which are passed through a controller card and into the control software. The second is a set of operator inputs. These inputs enable communication between the operator and the control software.

6.1.1 Hardware Inputs and Outputs

The CAS control software interacts with the CAS hardware through a DMC-1780 ISA-bus controller card produced by Galil Motion Control, Inc. The software monitors and/or controls the following signals using this controller card.

6.1.1.1 Inputs--TTL

Emergency stop
System low pressure switch
High pressure switch
Air Bearing #1, air bearing #2, air bearing #3, and air bearing #4 low pressure switches
U1, V1, U2, V2, W1, W2, W3, and W4 amplifier faults
U1, V1, U2, V2, W1, W2, W3, and W4 axis forward limit switches
U1, V1, U2, V2, W1, W2, W3, and W4 axis reverse limit switches
U1, V1, U2, V2, W1, W2, W3, and W4 axis home switches

6.1.1.2 Inputs—16 bit Analog

U1, V1, U2, V2, W1, W2, W3, and W4 position sensors

6.1.1.3 Outputs--TTL

OK/pneumatic solenoid signal
U1, V1, U2, V2, W1, W2, W3, and W4 step
U1, V1, U2, V2, W1, W2, W3, and W4 direction
U1, V1, U2, V2, W1, W2, W3, and W4 amplifier shutdown

6.1.2 Operator Inputs

Operator inputs are commands sent to the control software either through the software's Graphical User Interface (GUI), directly at the local host, or through a CDS command signal sent across an ethernet connection. The current state of the CAS determines which inputs are available. Please refer to section 6.2 for a complete description of the CAS software states. Operator inputs are used to transition the CAS into a new state. For example, the MOVE CAS input transitions the CAS from the ONLINE state to the IN-MOTION state. Please refer to section 6.3 for a complete description of the CAS software state transitions.

6.1.2.1 BRING CAS ONLINE

This input is only available while the CAS is OFFLINE and is used to transition the CAS to ONLINE. ONLINE indicates that the CAS is ready and capable of performing a motion command.

6.1.2.2 TAKE CAS OFFLINE

This input is only available while the CAS is ONLINE and is used to transition the CAS to OFFLINE.

6.1.2.3 MOVE CAS

This input is only available while the CAS is ONLINE and is used to transition the CAS to IN-MOTION. The MOVE CAS input specifies the required motion in terms of roll, pitch, yaw, or translation in the U, V or W directions.

6.1.2.4 EMERGENCY STOP

This input is available while the CAS is either ONLINE or IN-MOTION and is used to initiate a transition to FAULT. Note that pressing one of the hardware emergency stops, is not, by definition, considered an operator input but the effect on the software is the same.

6.1.2.5 STOP MOTION

This input is only available while the CAS is IN-MOTION and is used to transition the CAS to ONLINE. All motion is immediately halted, but a fault condition is not generated since the stop was operator requested.

6.1.2.6 POWER DOWN CAS

This input is only available while the CAS is either OFFLINE or ONLINE and is used to transition the CAS to POWERED-DOWN.

> Note: If the POWER DOWN CAS command is given while the system is ONLINE, the control software will first transition the system to OFFLINE and then transition to POWERED-DOWN.

6.1.2.7 CLEAR FAULT

This input is only available while the CAS is in a FAULT state and is used to transition the CAS to OFFLINE. Clearing a fault is not always possible because the hardware may still exhibit the faulted condition. In this case, the system is returned to a FAULT state unless health monitoring has been toggled off using the TOGGLE HEALTH MONITORING input, described below.

6.1.2.8 CAS STATUS

This input is available from any state. It is used to send health and status information to the operator.

6.1.2.9 TOGGLE HEALTH MONITORING

This operator input toggles health monitoring on or off. This is to be used with extreme caution. Without health monitoring the control software will be unable to prevent state-to-state transitions, and would be unable to transition to FAULT when needed.

> Note: this option is included for CAS testing and "debugging" purposes only and should never be used when actually operating the CAS under normal conditions.

6.2 Software States

From the software's point-of-view, it is important to know whether or not the conditions of a current state are satisfied. If they are not, the software transitions to a new state that satisfies the current conditions. The software continually monitors the status of the CAS to see if a state transition is necessary. This section summarizes when and why a transition will occur. For specific information about the transition from state-to-state see section 6.3, below.

6.2.1 POWERED-DOWN State

The software has no control over this state.

6.2.2 OFFLINE State

While OFFLINE the software monitors the hardware inputs against the conditions of section 5.3.2. If any of the hardware inputs are inconsistent with these conditions, the software transitions to FAULT. Additionally, the operator may request a state transition using either the BRING CAS ONLINE or POWER DOWN CAS operator inputs which trigger transitions to ONLINE or POWERED-DOWN, respectively.

6.2.3 ONLINE State

To stay ONLINE the software monitors the hardware inputs against the conditions of section 5.3.3. If any of the hardware inputs are inconsistent with these conditions, the software transitions to FAULT. Additionally, the operator may request a state transition using the MOVE CAS, EMERGENCY STOP, or TAKE CAS OFFLINE operator inputs which will force a transition to IN-MOTION, FAULT, or OFFLINE, respectively.

6.2.4 IN-MOTION State

While IN-MOTION the software monitors the hardware inputs against the condition of section 5.3.4. If any of the hardware inputs are inconsistent with these conditions, the software transitions to FAULT. Additionally, the software actively controls the stepper motors based on the MOVE CAS operator input. The MOVE CAS flow diagram of section 6.4.4.5 details the logic used to complete a move. If a position error or a timeout occurs during the move, the system is transitioned to FAULT. The operator may also

request a state transition using either the STOP MOTION or EMERGENCY STOP operator inputs which will force a transition to ONLINE or FAULT, respectively.

6.2.5 FAULT State

Once the software has been placed in the FAULT state only the CLEAR FAULT request can transition the software to OFFLINE. The system does not monitor health and status, but health and status can be queried with a CAS STATUS request.

6.3 Software State Transitions

A transition can be the result of an operator request, a software initiated action, or both. For example, while the system is OFFLINE the software cannot, of its own accord, move the system to ONLINE; the software requires external input from the operator. This section details how the control software deals with each of the transitions shown in figure 4.1.1.

6.3.1 POWERED-DOWN to OFFLINE

The CAS electronics must either be powered-up manually or powered-up by a CDS-controlled relay. After the electronics are powered-up, the control software is started. Once the control software is running, it initializes flag variables, sends signals to disable the amplifiers, and checks for the current position of all eight axes. These positions are compared with recorded positions that correspond to the position of the CAS when it was last powered-down. If the differences in these positions are within a given tolerance the system is placed OFFLINE. If not, the system is transitioned to FAULT and error-indicating variables are set to the appropriate values.

6.3.2 OFFLINE to POWERED-DOWN

The software, upon receiving the POWER DOWN CAS command, saves to disk the current axes' positions and then shuts itself and the host computer down. The CAS electronics must be powered-down either manually or by a CDS-controlled relay.

6.3.3 OFFLINE to FAULT

This transition occurs when the software's health monitoring routines perceive a fault, as discussed in section 6.2.2. The system is placed in the FAULT state and error-indicating variables are set to the appropriate values.

6.3.4 OFFLINE to ONLINE

If the operator gives the BRING CAS ONLINE input, then the software attempts a transition to ONLINE. First, the software enables the amplifiers. Then an output signal is sent to open the pneumatic solenoid, allowing the air bearings to energize. The software looks for sufficient bearing pressure by examining the system's low and high pressure switches. If these correspond to the conditions in section 5.3.3, the system is placed ONLINE. If however, adequate pressure is not achieved or a high-pressure situation results, then the system is transitioned to FAULT and error-indicating variables are set to the appropriate values.

6.3.5 ONLINE to OFFLINE

An operator input of TAKE CAS OFFLINE attempts to transition the system to the OFFLINE state. The signal to the pneumatic solenoid is removed, cutting off air-supply to

the bearings. The software checks that the pressure has dropped and sends output signals to disable the amplifiers. The system is then placed OFFLINE. If the pressure does not drop, error-indicating variables are set to the appropriate values and the system is transitioned to FAULT.

6.3.6 ONLINE to FAULT

This transition occurs when the software's health monitoring routines perceive a fault, as discussed in section 6.2.3, or the operator issues an EMERGENCY STOP request. Error-indicating variables are set to the appropriate values, and the system is placed in the FAULT state.

6.3.7 ONLINE to IN-MOTION

An operator input of MOVE CAS initiates the transition to IN-MOTION. First, the motion command is transformed into local coordinates and then the necessary axis translations are determined. If the translations would move an axis out-of-range, an error message is issued and the system is left in the ONLINE state. If the motions are acceptable (do not exceed software limits) the system is placed IN-MOTION.

6.3.8 IN-MOTION to ONLINE

The software transitions the system to ONLINE if the required motion is achieved.

6.3.9 IN-MOTION to FAULT

If the required move is not achieved within a specified time limit, a limit switch is triggered, or a large motion error is sensed, the system is placed in the FAULT state and the software sets appropriate error variables.

6.3.10 FAULT to OFFLINE

The operator can clear system faults using CLEAR FAULT and attempt to transition the system into the OFFLINE state. If clearing the error only results in triggering the error again, the system will move back into FAULT. However, if the operator wishes, the error(s) can be ignored using the administrative input, TOGGLE HEALTH MONITORING. This would allow the operator to successfully transition to OFFLINE because health monitoring is turned off!

> Note: this option is included for CAS testing and "debugging" purposes only and should never be used when actually operating the CAS under normal conditions.

6.4 Implementation

This section describes HYTEC's plans for implementing the high-level software logic defined in the previous section. Areas covered are: the development environment, user interfaces, heath and status monitoring, and the software logic for operator inputs.

6.4.1 Development Environment

The control software will be developed using Visual Basic 5.0. HYTEC chose this product for its robust development environment, its ability to rapidly generate graphical user interfaces, and the availability of pre-written tools and controls. Chief among these is a set of custom controls developed by GALIL Motion Control which provides an interface to the DMC-1780 controller card.

6.4.2 User Interfaces

Two interfaces shall be provided: a graphical user interface (GUI) for local control and a networked based interface for remote control. A sample GUI is shown in figure 6.4.2.1. The GUI provides access to the Operator Inputs in a controlled manner. For example, the system depicted in figure 6.4.2.1 is currently in the OFFLINE state, therefore the user only has access to the BRING CAS ONLINE, POWER DOWN CAS, and Administration operator inputs. Note: Administration is actually a collection of inputs containing CAS STATUS, TOGGLE HEALTH MONITORING, and CLEAR FAULT.
The second interface is currently undefined. CDS needs to work with HYTEC in defining a network-based interface. Currently, an EPICS compatible interface is being considered. An EPICS ActiveX control has been developed at LANL which may satisfy the requirements for a network interface.

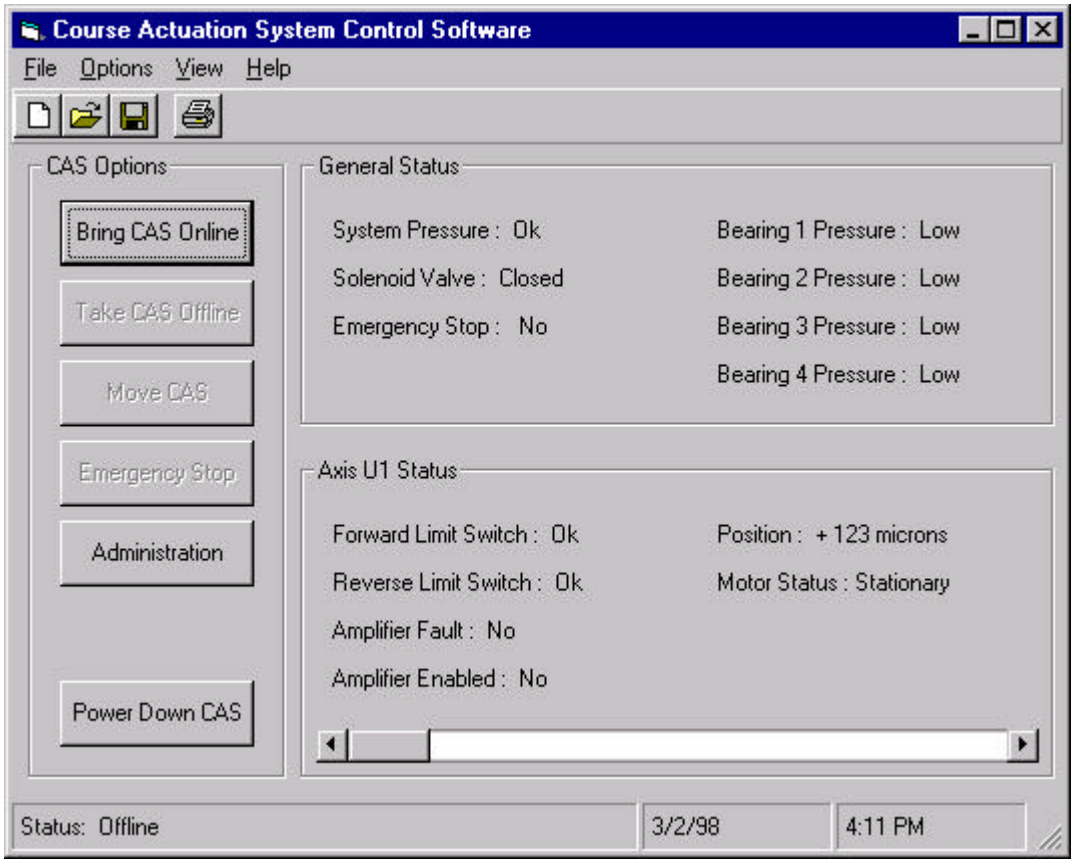


Figure 6.4.2.1. Sample GUI for the CAS Control Software.

6.4.3 Health and Status Monitoring

Health and status monitoring refers to the software's ability to query the hardware inputs to establish the health and status of the CAS. The software implements two forms of health and status monitoring. The first uses periodic polling, and the second uses PC interrupts.

Some of the GALIL OCX controls can be used to periodically poll the status of the hardware inputs and outputs. If a change in any of these occurs the control raises an event.

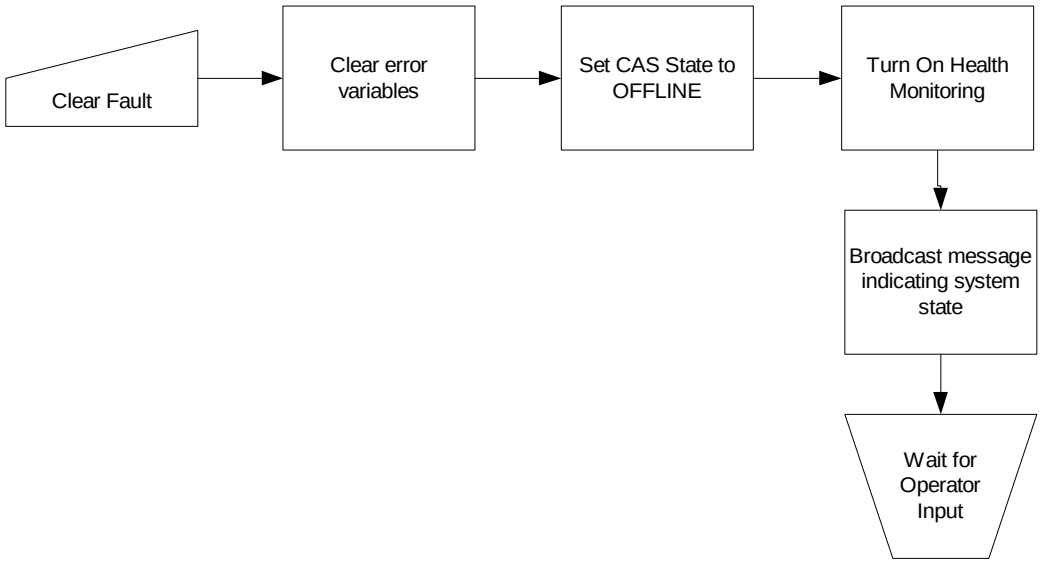
The DMC-1780 board can also be configured to issue a PC interrupt upon the triggering of a limit switch or the toggling of a TTL input. The GALIL OCX traps the interrupt and then raises an event.

This monitoring is used to establish health and status while waiting for an operator input (the system is expected to be sitting in a specific state). When a health-monitoring event is raised, the software checks the state of the hardware inputs and compares them to the conditions required to maintain the current state. If they do not match, the software transitions to FAULT. These forms of monitoring can be enabled or disabled through the TOGGLE HEALTH MONITORING operator input.

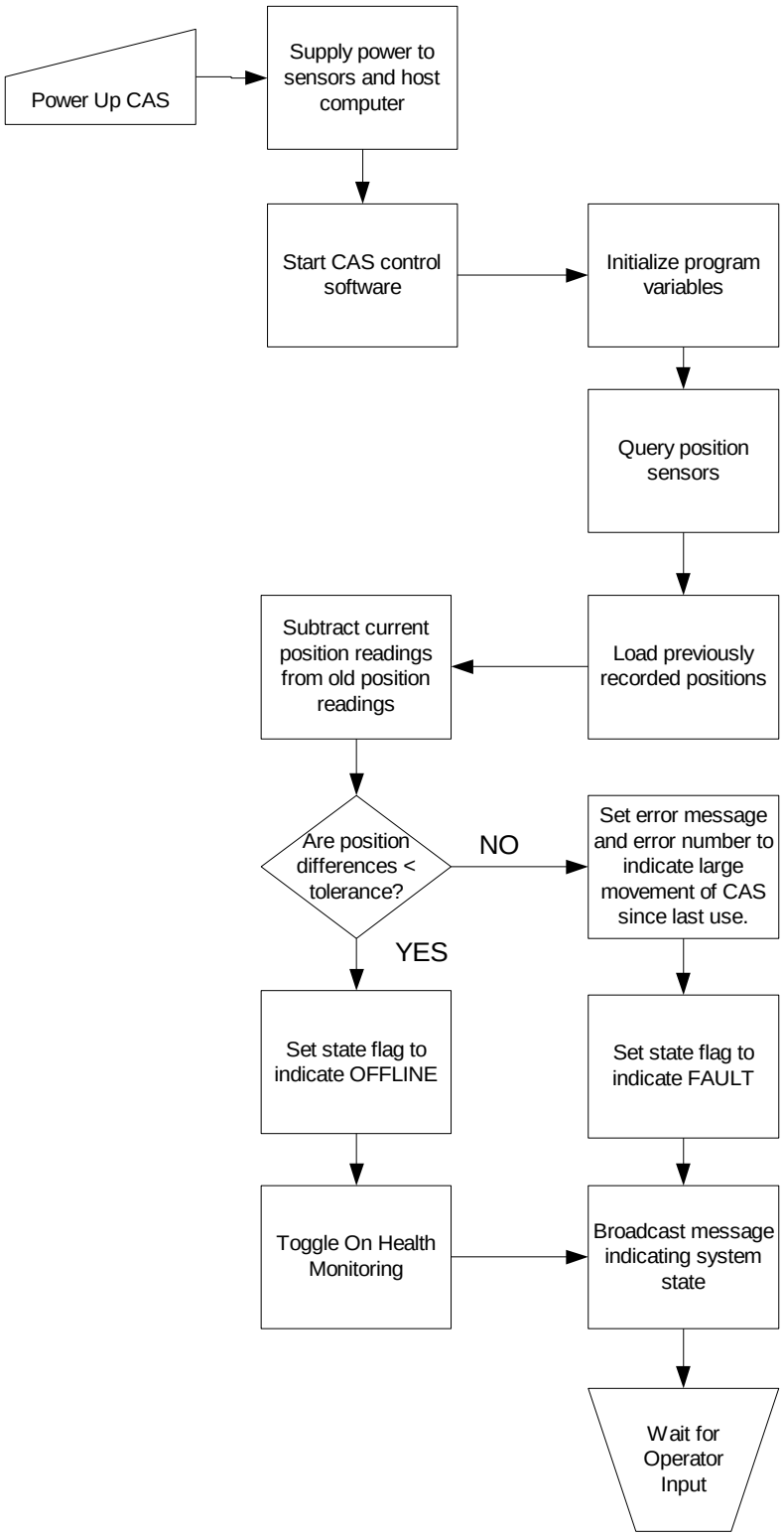
6.4.4 Software Logic for Operator Inputs

The following sections provide flowchart descriptions of the software logic associated with an operator input.

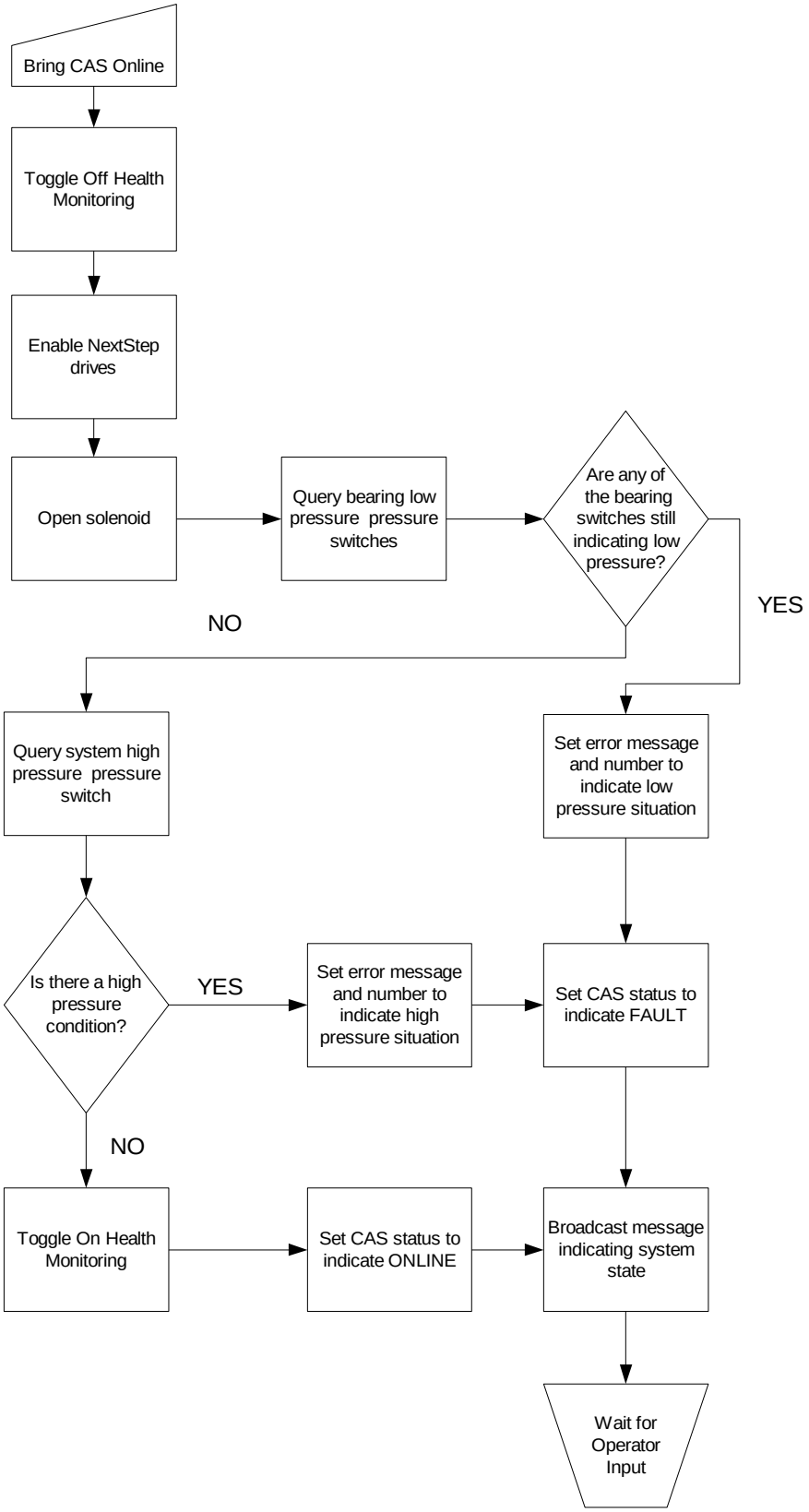
6.4.4.1 CLEAR FAULT



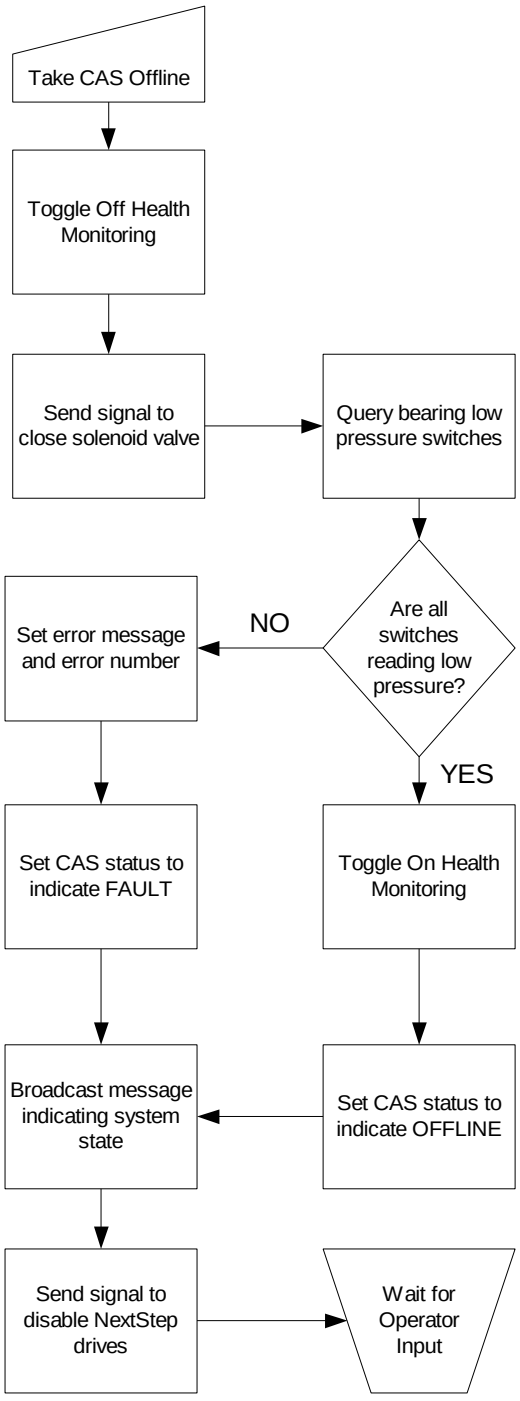
6.4.4.2 POWER DOWN CAS



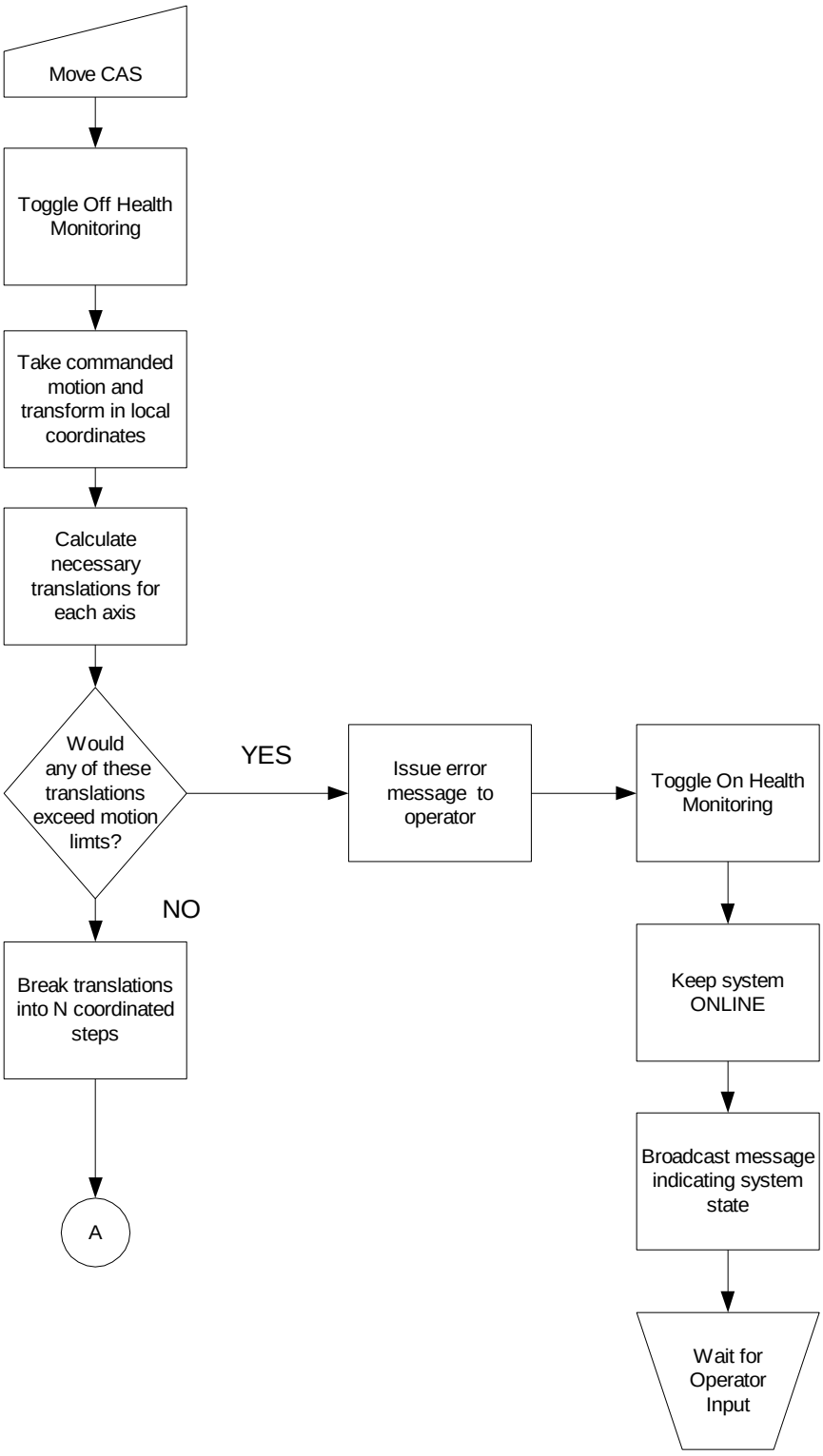
6.4.4.3 BRING CAS ONLINE

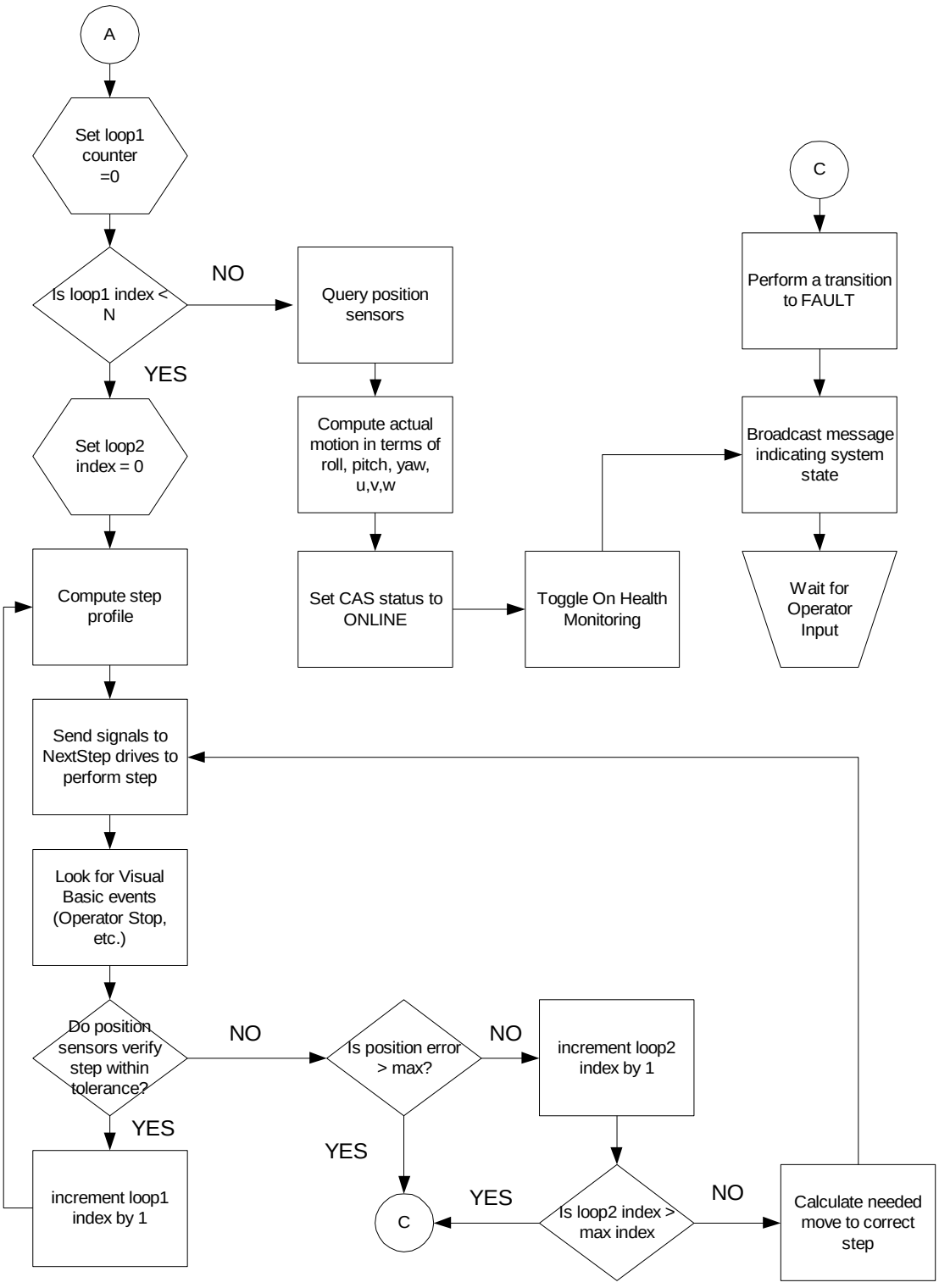


6.4.4.4 TAKE CAS OFFLINE

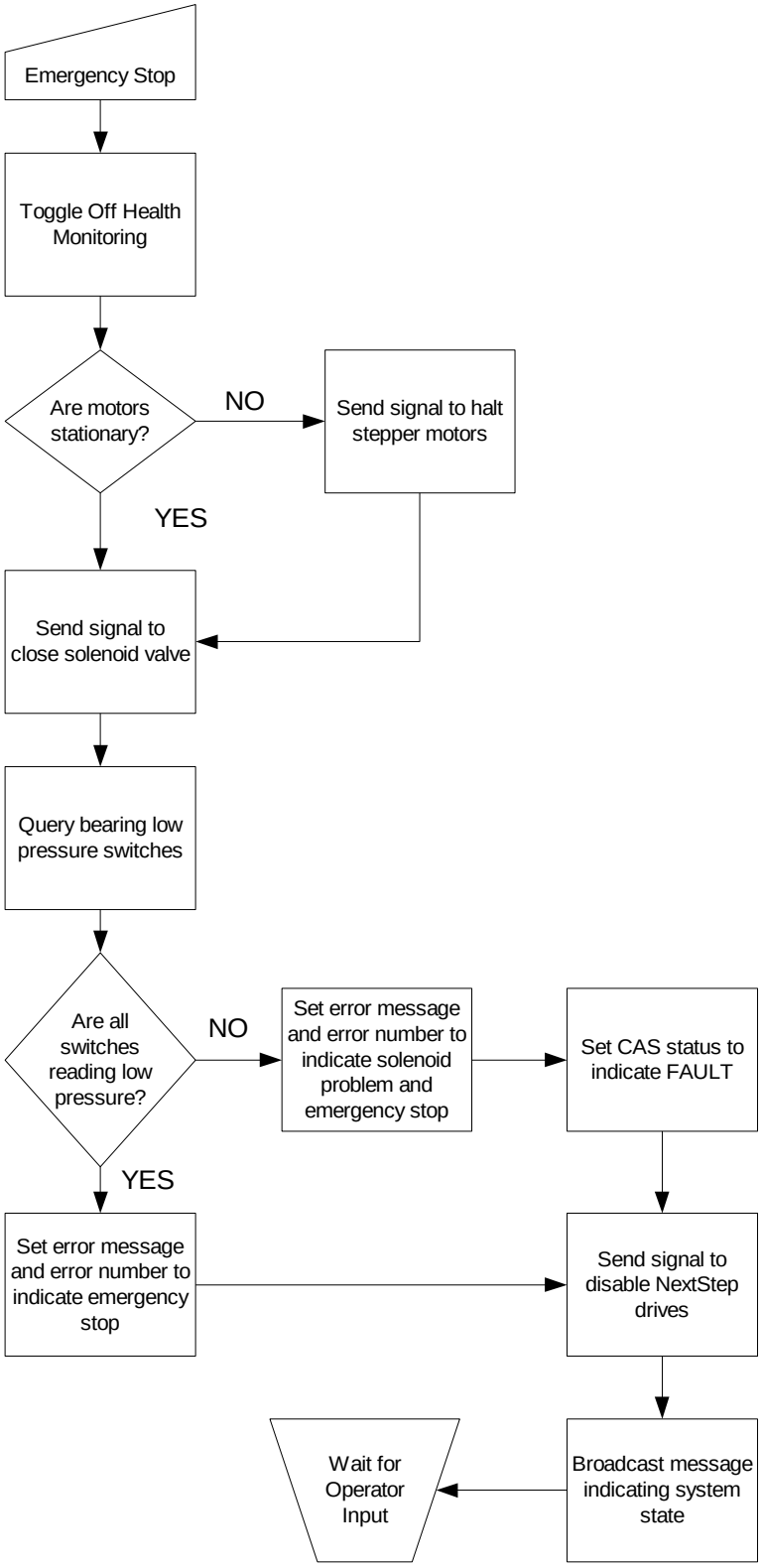


6.4.4.5 MOVE CAS

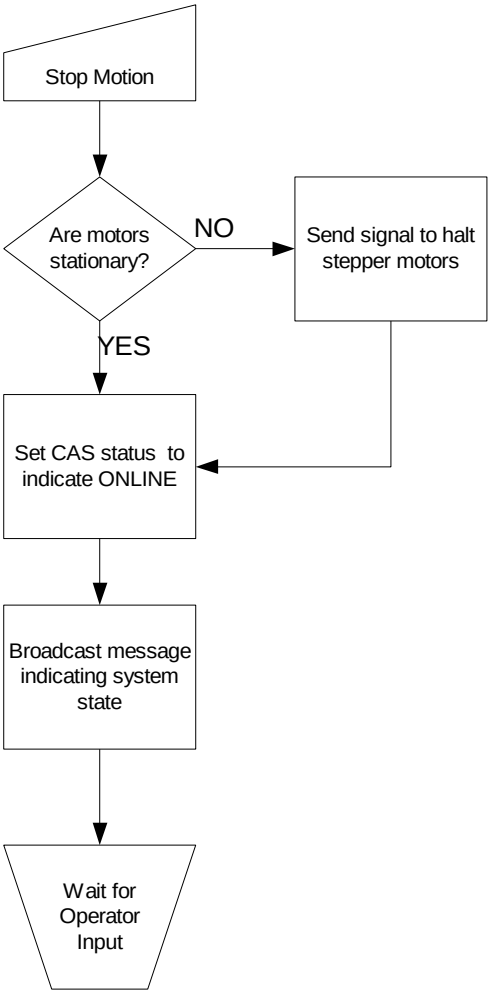



A
Set loop1 counter =0
Is loop1 index < N
NO
Query position sensors
YES
Set loop2 index = 0
Compute actual motion in terms of roll, pitch, yaw, u,v,w
C
Perform a transition to FAULT
Broadcast message indicating system state
Compute step profile
Set CAS status to ONLINE
Toggle On Health Monitoring
Wait for Operator Input
Send signals to NextStep drives to perform step
Look for Visual Basic events (Operator Stop, etc.)
Do position sensors verify step within tolerance?
NO
Is position error > max?
NO
increment loop2 index by 1
YES
YES
increment loop1 index by 1
C
YES
Is loop2 index > max index
NO
Calculate needed move to correct step

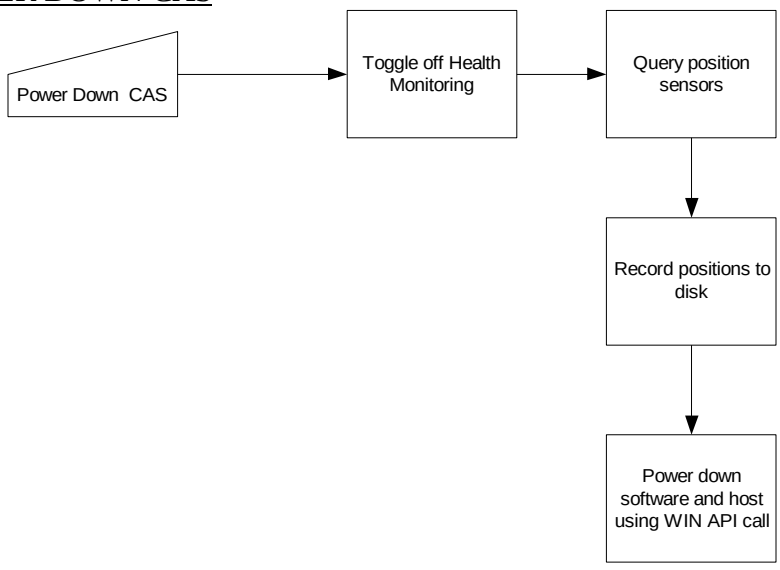
6.4.4.6 EMERGENCY STOP



6.4.4.7 STOP MOTION



6.4.4.8 POWER DOWN CAS



7. Conclusions

HYTEC is developing control system logic that meets the requirements of LIGO_TS_02. The system design, based upon a state and state transition approach, provides a controlled means of achieving these requirements while protecting against failure. Both hardware and software based interlocks are used and in many cases, the roles of the hardware and software are the same.

Particular attention has been paid to controlling the air-bearing subsystem. Failure of these bearing components would have a significant effect on LIGO operations. Additionally, every effort has been made to prevent motion of the CAS actuators under a faulted condition. While not in motion, each actuator receives a holding torque, guaranteeing the stability of these components.

The software provides both local and remote control of the CAS through GUI and network-based interfaces and communicates or controls the CAS hardware using an ISA-bus DMC-1780 controller card. HYTEC is developing the control software using Visual Basic 5.0 and a set of OCX controls provided by GALIL. The software monitors the state of the CAS hardware and is capable of responding to both system faults and operator inputs.

Certain areas of the system design require feedback from CDS. In particular, CDS should state whether or not the CAS electronics will be power-up remotely, using a relay, or locally. Additionally, CDS needs to provide input regarding the remote control interface to the CAS software.