HARDWARE AND SOFTWARE CONTROL FOR THE NASA EOS SATELLITE POWER SYSTEM TESTBED

by

XUESI MANG

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

APPROVED:

Fred C. Lee

Fred C. Lee, Chairman

Daniel Mark Sable

Dusan Borojevic

May, 1994

Blacksburg, Virginia

HARDWARE AND SOFTWARE CONTROL FOR THE NASA EOS SATELLITE POWER SYSTEM TESTBED

by

XUESI MANG

Fred C. Lee, Chairman

Electrical Engineering

(ABSTRACT)

For the implementation of the NASA EOS testbed programmable operations, a control/data acquisition system has been designed, implemented, and tested. The work is focused mainly on the hardware design / build of the custom interface board and software development for real-time control and data acquisition of the testbed. Also, an application program was designed to control, collect, and display the data and parameters of the testbed elements, and troubleshoot the hardware. An algorithm was chosen and applied to the battery simulator control after computer simulations. The final experiment results were obtained successfully, which proved the design to be correct, as the testbed could be operated under computer control.

To My Parents

and

My Grandmother

Acknowledgements

I would like to express my sincere gratitude to Dr. Fred C. Lee, the chairman of my graduate committee, for his encouragement throughout my Master's program. I am also very grateful to Dr. Daniel M. Sable for his support and fruitful suggestions of this research work. I am thankful to Dr. Dusan Borojevic for his review and comments on this thesis. Many thanks should go to all members of the Space Power Lab: Steve J. Butler, Jim Noon, Zvi Gur, David Crow, Paul Duncan and Ashok Pile, for their very valuable information and good cooperation. The simulation results of the battery model from EASE 5 were obtained by Ashok and Zvi, which allowed me to get control algorithm much more easily. Steve and Jim helped me a great deal to finish the work quickly when we were testing the interface with the solar array simulator and battery chargers.

Table of Contents

List of Illustrations

List of Tables

Chapter 1 Introduction

1.1 Background

A testbed for the NASA EOS satellite power system has been designed as a means of verifying computer simulation results and of providing design valuable insight for a high voltage DC-regulated spacecraft bus [1]. The entire system is controlled by a personal computer (PC) via an IEEE-488 bus.

Figure 1.1 shows the block diagram of the testbed. The testbed consists of test system elements (outside the dotted line) and power system elements (inside the dotted line). The test system includes a solar array simulator, two battery simulators and a load simulator. The solar array simulator provides the I-V characteristic of 22 simulated solar array strings [2]. Two Ni-Hi battery simulators contain a hardware element and a software element. The hardware component has the capability to source and sink current with the voltage range from 53 V to 84 V. The software component possesses the capability to emulate Ni-Hi battery characteristics [3]. The load simulator

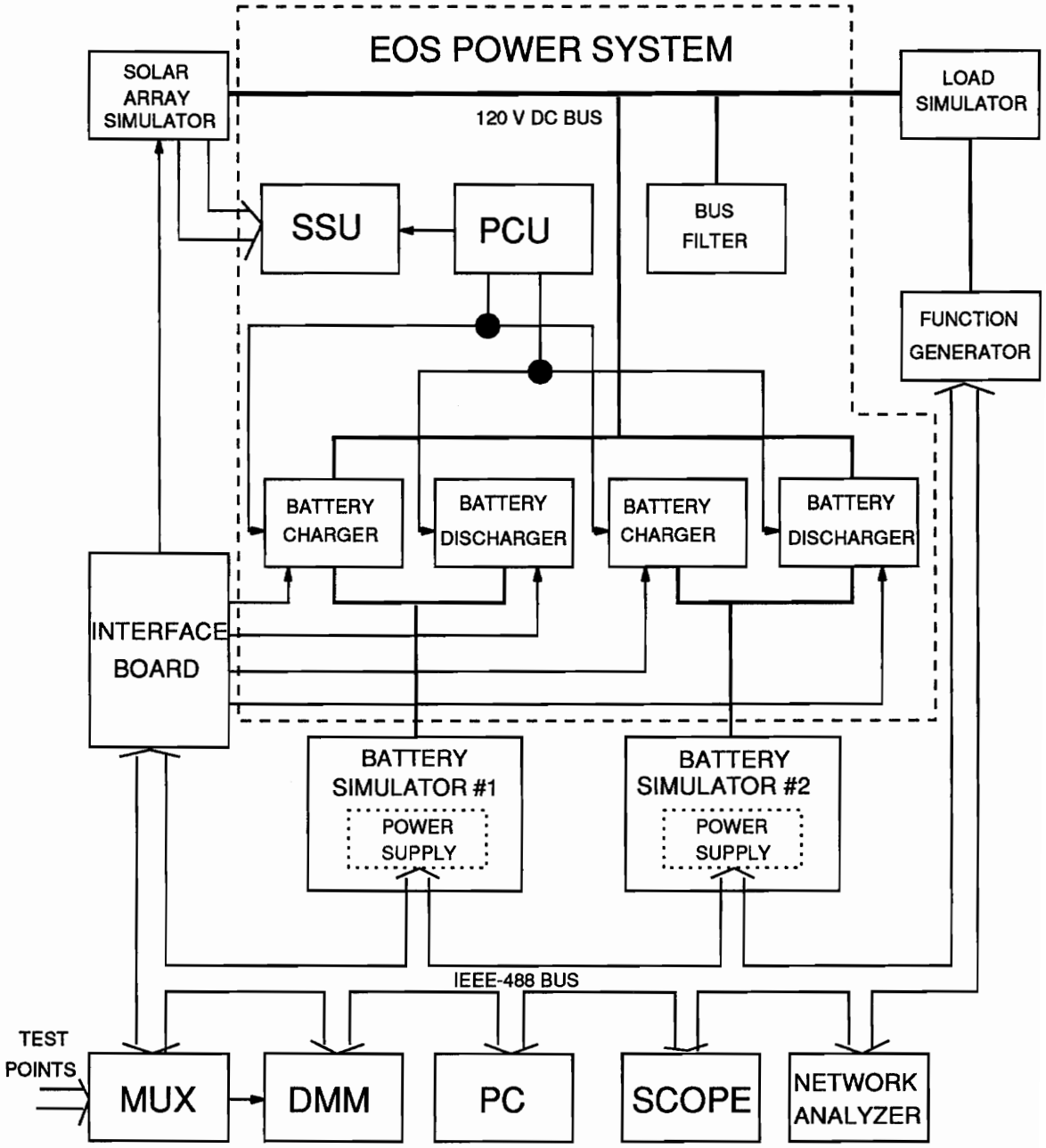


Figure 1.1 Block Diagram of the NASA EOS Testbed

is driven by a programmable function generator to provide a load up to 3 kW with various wave shapes.

Since the PC performs as a master controller, more test and measurement facilities can be added to the testbed system, such as digital oscilloscope, network analyzer, and multiplexer, which are able to make this power system test mission more efficient, accurate, and user-friendly.

The NASA EOS Satellite Power System contains a 120-V regulated bus energy transfer power system which has two battery discharger regulators (BDRs) [4], two battery charger regulators (BCRs) [5], and a solar array shunt switching unit (SSU) [6]. The power control unit (PCU) controls the entire system operation by the function of regulating the 120-V bus voltage within the band specified, from maximum discharger load to maximum solar array shunt current.

In each orbit during the sunlight period, the 120-V bus is regulated by the solar array shunt switching unit (SSU), while the batteries are charged to a command rate. During the eclipse period, the battery dischargers are activated to provide the satellite power.

When in the battery discharge mode, the spacecraft bus is regulated to about 119 V. When the solar array has inadequate power to provide the load and charge the battery to the commanded level, the battery chargers regulate the bus to 120 V. When in the shunt mode, the bus voltage is regulated to 121 V. One-volt dead bands between modes insure that the battery charger and discharger never operate simultaneously and that the shunt never activates until the battery chargers are operating at their commanded rates.

Each of the two BDRs outputs 120 V at a maximum power level of 1800 W. The two BCRs provide 16 commandable constant current charge rates up to a maximum of 23 A. The SSU regulates the solar array voltage by either connecting or shunting a combination of the 22 solar array strings. One of the 22 solar array strings is pulse-width-modulated.

1.2 Major Contributions

The purpose of this work is to design a useable hardware and software of a command and data acquisition system interfacing to a testbed for the EOS satellite power system via an IEEE-488 bus. The system is to control parameters of the testbed elements, to control the real-time operation of the battery simulator, and to collect test data during the simulation.

The work discussed in this thesis was finished on September, 1992; it was a part of the beginning stage of the testbed implementation. It also created a solid basis for the later software development of the graphically-based command and data acquisition system.

The major work I have done is listed as follows:

1. Design and implementation of a custom hardware interface board to provide digital commands for the various testbed elements via an IEEE-488 bus.

2. Design of a menu-driven control and data acquisition software.

3. Design of a complete software driven self test and diagnostic system for the testbed.

4. Computer simulation of control algorithm for the battery simulator.

5. Design of a real-time control software for battery simulator to provide an orbit simulation.

1.3 Outline of the Thesis

The design requirements and principal configuration of the testbed elements on the IEEE-488 bus are covered in Chapter 2. The interface hardware design is detailed in Chapter 3. The menu-driven software development is included in Chapter 4. Chapter 5 describes the implementation of the battery simulator software algorithm. The control scheme of the solar array is illustrated in Chapter 6. Experimental results are shown in Chapter 7. Conclusion and some suggestions for the future work are discussed in Chapter 8.

Chapter 2 Configuration of the Controller and Data Acquisition System

2.1 Objective

The purpose of the testbed command and data acquisition system is to provide power system commands in order to obtain the computer-controlled operations and experimental data acquisitions. The tests such as battery simulator with the characteristics of Ni-H2 type modeled by the software, and the orbit simulation can be implemented under computer control only.

As shown in Figure 1.1, it is desirable that all the functions of both test system and power system except SSU and PCU be programmable via this control/data acquisition system. The IEEE-488 Bus (also called GPIB or HPIB Bus) is chosen as the system communication/control bus providing the central construction of this control/ data

acquisition system. A model 55SX IBM PS/2 computer with an HPIB controller card is used to stimulate and monitor the operation of the testbed.

The testbed should be able to run in one of the two computer-driven modes: menu-driven or orbit simulation. The control/data acquisition system would provide these approaches to programmable operation of the testbed, including real-time operation of the battery simulator.

In the first mode, the control/data acquisition system functions as an interface between user and testbed elements and commands each element properly based on the user's input from the keyboard. In addition, a programmed procedure, self test, is also provided in this mode to test part of the power devices and power supplies of the testbed in a very quick and convenient way, to make troubleshooting easier and reduce the risk of further damage.

The function of the orbit simulation is a real-time control process designed to command the testbed with the programmed parameter setting based on the predefined timing which represents an orbit operation environment.

Furthermore, it is necessary for the testbed system operated by computer control to have more comprehensive programmable tests in the future. The characteristics to be tested include efficiency, output impedance, step load response, bus ripple, bus regulation, and charge regulation.

2.2 System Block Diagram and Design Requirement

As shown in Figure 1.1, this control/data acquisition system consists of a computer(controller), several organized IEEE-488 programmable instruments, and a custom interface which connects each control circuit of the testbed hardware units to the IEEE-488 bus and makes them programmable via the Bus. The controller is an IBM PC, PS/2 model 55 sx with an installed National Instruments IEEE-488 control board. With this system, the user can obtain the programmable operation functions of the power system under the different simulated conditions either by changing the control parameters of the specified power units through the computer keyboard each time, or by executing a particular pre-designed real-time control software. The system is also able to collect and save the experimental data for future analysis upon the user's requirements during the testbed operation.

The basic instruments used in the system are:

1. HP3478A digital multimeter

   The main medium for voltage and current reading, it has six basic measure functions, including two-wire and four-wire resistance measurements, and two pairs of switchable front and rear connectors.

   This unit is to be used with the connection of a multiplexer. All of the information about the control/data acquisition system needs can be read from this meter. A delay is needed after each reading if it is switched from a high to a low voltage,

since a transient time exists because of a very high input impedance of the meter.

2. HP8904A multifunction synthesizer

   It is used for a load profile generator controlled by the computer. As a control signal, synthesizer's output is sent to two dynamic loads, Dynaload DLR-400-15-2500A. The generation of 0 to 10 V dc voltage is corresponding to 0 to 51.2 Ampere load scaled in this testbed. Programmed load profiles can be obtained by the user through the computer software.

3. Philips PM21 multiplexer unit (System 21)

   This unit has four boxes connected by an internal bus. Three of them are switch boxes with a total of 60 individually controllable switches which are employed in this data acquisition system for accessing different measurement points. A master box is attached to the IEEE-488 bus to manage the interface functions.

   A 37-pin connector is at the rear of each switch box, and the unit can be expanded by adding more switch units.

   The following signals are multiplexed:

   - bus voltage
   - bus current
   - battery simulator output voltage

- battery simulator current

In the self test mode, there are more points that have to be multiplexed. A switch on the front panel is for switching between a normal operation and the self test mode.

4. Sorensen Micro-Dap power supply controller

Two Sorenson power supplies (DCR 80-33B) are used for two battery simulators of the testbed. There is a power supply auto/manual switch mounted on the panel of the battery simulator rack for selection of the battery simulator power supply control mode.

In the auto mode, a power supply programmer, Sorensen 488 Micro-Dap, actually a digital-to-analog converter, is connected between the IEEE-488 bus and two Sorensen power supplies. With a microcontroller inside, it receives the digital signals from the bus, converts them to analog signals, and sends them to the power supplies for control of the voltage output and current output. With only one bus address, it has two channels of analog outputs, controlling voltage and current, respectively (two of 25-pin D-type connectors), and eight logical outputs (a 9-pin D-type connector) which can be used to shut down either power supplies or loads.

At each sampling moment, an updated digital voltage command is sent to Micro-Dap, and simulated battery voltage will be output from the power supply which is followed by this command.

The current limit also can be programmed each time. This value may be set at the beginning or given to the power supply at each time with the voltage command.

5. Lambda power supply

Three Lambda power supplies (TL-864) are employed for the solar array simulator. One of them is IEEE-488 controllable (maximum rating: 64 V and 70 A) which is directly hooked on IEEE-488 bus, since the interface circuit is built internally. A routine has to be designed to drive this power supply at any time.

At the self-test mode, only this one is controlled by the computer to generate the desired voltage. A switch on the front panel is dedicated to the switching of this power supply switching between normal operation and the self-test mode.

6. Network Analyzer

This unit is used for automatic measurement of system loop gain and output impedance. Since it has so many features, a dedicated software routine has to be designed.

2.3 Design Considerations

The major design considerations are as follows:

1. In order to obtain computer controlled testbed operation, the following units require a custom interface:

   - two battery chargers;
   - two battery dischargers;
   - a solar array simulator.

   So, a custom interface has to be designed and built and should be able to:

   a. address the power units;

   b. receive the command from the computer;

   c. interpret the received command to the elements of the testbed.

2. The interface circuit design should be as simple as possible. Since the testbed elements that require the custom interface to connect each element of the testbed play the role of listeners only (no feedback data), the custom interface circuit can be built by standard TTL logic chips, instead of using microcontroller. In this way, no additional low level software design is necessary.

3. In the orbit simulation mode, it is useful to have a time scaling variable to allow simulation of a 90-minute orbit in a reduced period of time.

4. The software should include built-in error detection to protect or reduce device damages.

5. A digital filter function may be applied to reduce the measurement noise.

6. The real-time control software has to be designed for battery simulator operation and orbit simulation of the testbed. So, the sampling rate and efficiency of routine execution have to be considered. On the other hand, some kind of hardware limitations in the instruments may affect the execution time of the software, so this interaction between software and hardware should also be included in the design. Since much of data acquisition work has to be done during the limited time period, some approaches, such as parallel execution, have to be used to make the software able to finish the job during the possible shortest sampling interval.

   After all, the control software should be designed so as to make the computer manage the entire system well, to achieve a high performance during the limited time interval.

7. In the final version of the software, the battery simulator real-time control is a major routine. The battery model is adapted from Zimmerman's Model [2]. The mathematic model is described by a group of nonlinear exponential differential equations. There are several algorithms that can be used to solve these equations on-line which will produce different errors and have different com-

plexities. Trading them off, the most efficient method should be chosen for our battery simulators. How to chose algorithms for the battery simulator control should be based on the comparison of the software simulations with the different input signals.

Chapter 3 Hardware Design

3.1 Hardware Design Requirements

Since the interface board to be designed performs only the functions of the listener to the IEEE-488 bus, it should be able to accept the IEEE-488 bus standard function AH1 (acceptor handshake function) and L3 (listener function), which are only necessary to complete the interface functions for this power system. Additionally, the logic circuit needs to be able to ignore the codes not used by this interface board, and not let them affect the board's operation.

The handshake function is the most important part of the interface board, since the IEEE-488 bus is asynchronous and any logic failure in handshaking can stop the data transfer of the bus system.

Interpreting the signal received by the interface to the elements of the testbed is the last stage before the signal being transferred to the control units of the power stage.

3.2 Custom Interface Board

Among the hardware units of the testbed, two battery simulators, a solar array simulator, two battery charger regulators, and two battery discharger regulators need run-time parameter control during each testbed operation. In the auto mode, the controller should be able to access and control each of them. Thus, a special interface circuit board with TTL logic ICs has been designed and built for interfacing each control circuit of the units mentioned above to the IEEE-488 bus, to make them IEEE-488 bus programmable. The circuit board has 15 ports, and each of them has an individual address to access; therefore, the total of 15 hardware units can be controlled via this interface board.

The schematic diagram of the interface circuit board is shown in the appendix, and block diagram is shown in Figure 3.1. Functionally, the entire interface circuitry is integrated by three sections: handshake management, address/command decoding, and data latching.

3.2.1 Handshake Management

This part of the interface board has four delayed pulse generators (U2, U3, U4 and U6) and two D-type flip-flops residing in a 74LS74 (U5). Each delayed pulse generator contains a 74LS221, dual non-retriggerable multivibrator. They are identical, except the delay time and polarity of the trigger signal transition.

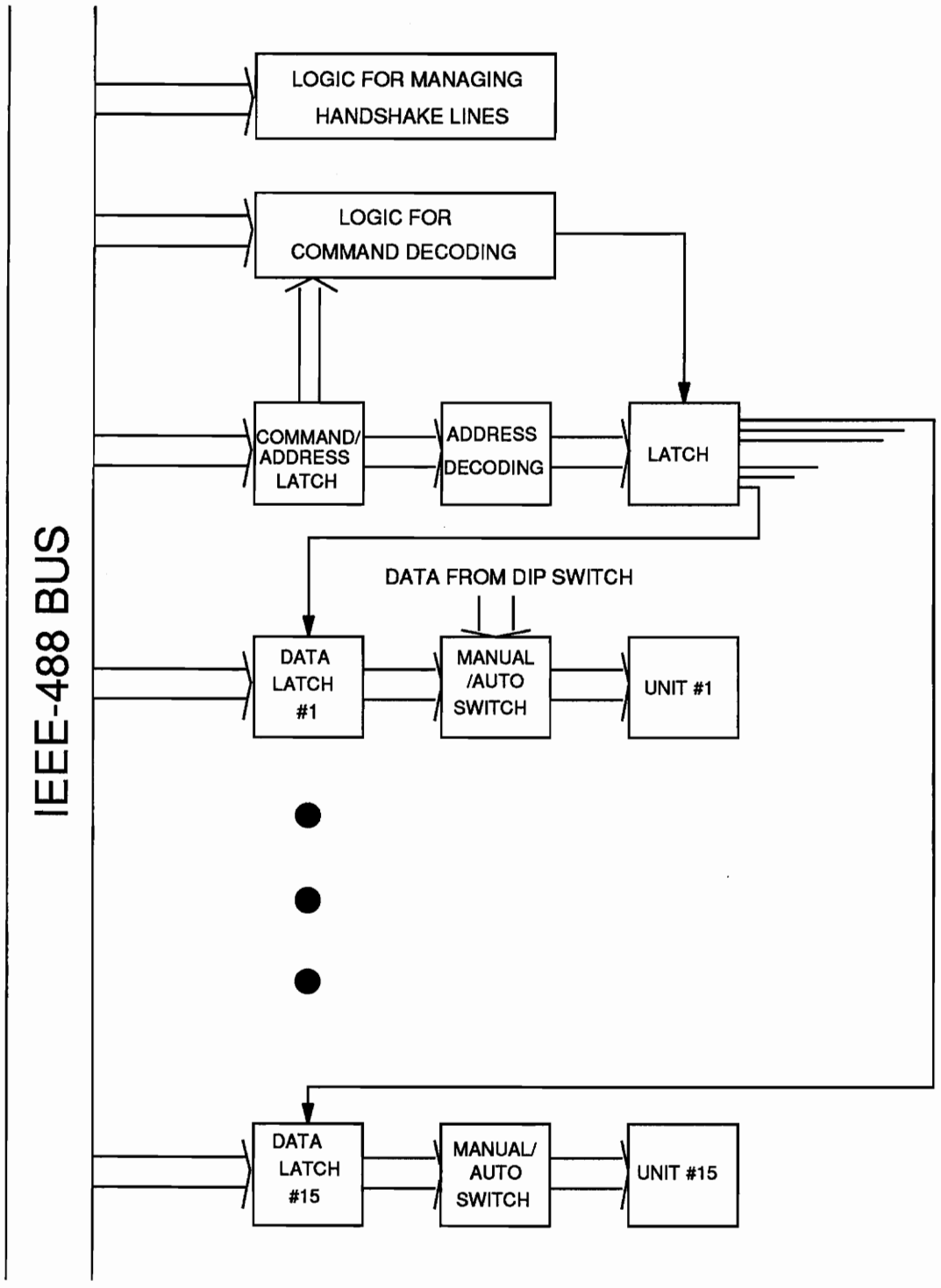


Figure 3.1 Block Diagram of Interface Board

The transmission of each data byte on the IEEE-488 bus starts with a talker or controller putting the valid data message, either the bus command or data message, on the DIO lines of the bus. After the data on the bus is stable, DAV line is asserted low (true) by the talker (note that IEEE-488 bus is negative-true logic, i.e. active low) to tell the listener(s) that the data on the bus are valid. In this circuit board, the DAV, a negative-edge signal, is used as a D input and a clock to drive the flip-flop (U5) output (pin 5 of U5) low after a 30 ns time delay. This output of the flip-flop triggers the data latch for data acceptance and drives the NRFD line low (true) through a bus tranceiver (75162). The signal on NRFD line tells the controller that the board is busy accepting the data from the bus. This 30 ns delay is for meeting the bus timing requirement. On the other hand, since the minimum setup time of 74LS374 is 20 ns, this delay can guarantee the validity of the data latched by the interface board circuit, especially if the talker or controller is in high speed of 488 bus timing of data transfer.

74LS374 needs 20 ns hold time. During this period, the input logic level must be maintained in order to ensure the continued recognition. So, after 30 ns of the DAV line being low, it is considered that the data on the DIO lines has been received by the interface board. At this moment, the NDAC line should become false (high) to inform the talker that the data has been accepted by this unit. Another delayed pulse generator (U3), triggered by the negative edge of the DAV line signal, sets the NDAC line high (false) by setting 74LS74 output (pin 9 of U5) 60 ns after DAV signal becomes low. The talker will acknowledge this by resetting the DAV line back to high as soon as the NDAC line going high has been detected, and will start to change the data on the data bus. The interface circuit knows the talker's acknowledgement by detecting the positive edge of DAV signal, and uses it to assert the NDAC low again after 60 ns by another pulse delay generator (U2) which can be triggered only by positive edge.

The NDAC line has so far completed one handshake cycle and is ready to repeat the next cycle of the data transfer.

The transformation procedure for each data byte is completed with the NRFD line going to high (false). The interface circuit is designed to assert the NRFD going high 60 ns after the NDAC line goes low, to give the controller or talker enough time to change the data on the DIO lines. This 60-ns time delay is made by the fourth pulse delay generator(U6). The talker will start the next byte data transfer after all the listeners set the NRFD line high. Since the NRFD and NDAC lines of the handshake bus in the IEEE-488 bus are designed as line-OR, these lines will be high only after every listener on the bus has set them high. In this way, the bus system allows the instruments with different response rates (delays) to accept and process the data byte in the same bus.

3.2.2 Addressing/Bus Command Decoder

This part of the interface circuit basically consists of three latches(U8, U10, U12) and two decoders(U9, U11).

In this custom interface circuit design, bus commands, also called interface messages, should be latched only by U8 (74LS374), a positive-edge eight-bit latch, whereas the data, also called devices dependent message, should be latched by one of the 15 latches, U21 - U35. All of these latches are trigged by the inverted and 30 nS delayed DAV signal (pin 6 of 74LS74).

If the ATN line of the bus is asserted low at the time when the DAV signal is low (true), it means the data on the bus is a bus command; otherwise it is a data. Based on the ATN line, as a trigger, the inverted output of the flip-flop(pin 6 of U5) driven by the 30-ns delayed DAV signal is sent either to an addressing/bus command latch(U8), 74LS374, or to one of the data latches (U21 - U35). An inverter and an AND gate can prevent the data message from being latched by the bus command/address latch(U8). The latching trigger can only go through an AND gate enabled by the inverted ATN signal to pin 11 of 74LS374 (latch enable). Another AND gate enabled by the port-selected signal will allow the latch trigger to go to the selected latch.

If a defined command or an address has been latched by U8, one of two 74LS138 (U9, U11, one-of-eight decoders), will decode the information. If the value of the code is less than 16, one of 15 decoder outputs will be low, and this port-selected signal puts a desired data latch/buffer(one of U21 - U35) on the data bus and makes it ready to receive the next data passed from the bus.

In case of error operation, the actual port-selected signal is generated from the second latch(U10 or U12) which is triggered by the positive edge of the DAV signal. The signals decoded by U9 and U11 are inverted to be active high in order to control the AND gates U13 - U15, thus allowing the data latch to be triggered only when the corresponding port selected signal is high. There are two purposes of using this second latch: for noise immunity and for decoding UNLISTEN bus command and rejecting undefined code. So all of the latches will be off the bus line if an UNLISTEN command is received. The address commands other than from one to fifteen generate the same responses.

The IFC line of the bus is meaningful to this circuit design. It initializes the NDAC line to be low (ready) by invoking the signal to the reset pin of 74LS74.

Because the address of the controller board in PC is zero, decoding of zero on the interface board has to be ignored. The total number of the ports in the interface is 15, so that two decoders, 74LS138, are used and selected by an inverter based on D4 of the data bus. If a valid address has been latched and decoded, one of the fifteen decoder outputs will give a high level signal which can enable one of the fifteen data latches to accept data information from the bus, if the next piece of information is a data message.

3.2.3 Data Latching

When the controller or a talker puts the data message on the bus, the ATN line will not be active during the DAV line being low. So the command latch (U8) is blocked, and only one of the data latches (U21-U35) which has been enabled by the last addressing command can receive the data from the data bus. The trigger comes from the output of the flip-flop(U5) through two AND gates. One of the AND gates is enabled by the ATN signal, and the other by the port-selected signal from one of the decoders (U10 or U12).

The following is the address table of the interface circuit board which has been used in the testbed:

address 1	solar array temperature controller
address 2	solar array devices self testing
address 3	battery charger #2 charge rate control

address 4	battery discharger #1
address 5	battery discharger #2
address 7	battery charger #1 charge rate control
address 9	solar array illumination controller.

3.2.4 Optocouplers and Peripheral Circuit

In order to protect the computer from the damage caused by accidental high voltage, optocouplers 2501 are used between the data latch output and each control circuit of the power unit. Thus, the computer and the interface board ground are isolated from the control circuit as well as from the power system ground.

Special care should be taken when connecting an instrument chase with the IEEE-488 bus signal ground. The use of a scope may connect power system ground to the computer ground through the utility cord.

Auto/manual switches have been installed for dischargers and chargers, which can make the data latches enabled or disabled. This is the convenient way for a user to decide whether the data come from the IEEE-488 bus or from the dip switches on the board. The dip switch can be set for selection of control parameters when in manual mode. A 74LS244 is applied as a dip switch latch/buffer. At any time, either 374 or 244 is in normal state, and the other is in high impedance state according to the mode selected. Figure 3.2 shows the circuit detail for mode switch.

For the chargers, two 4016 ICs were used to switch manual/auto mode, since the control circuit in charger is CMOS-compatible.

For discharger control, a two-bit digital to analog converter has been built for discharger bias control.

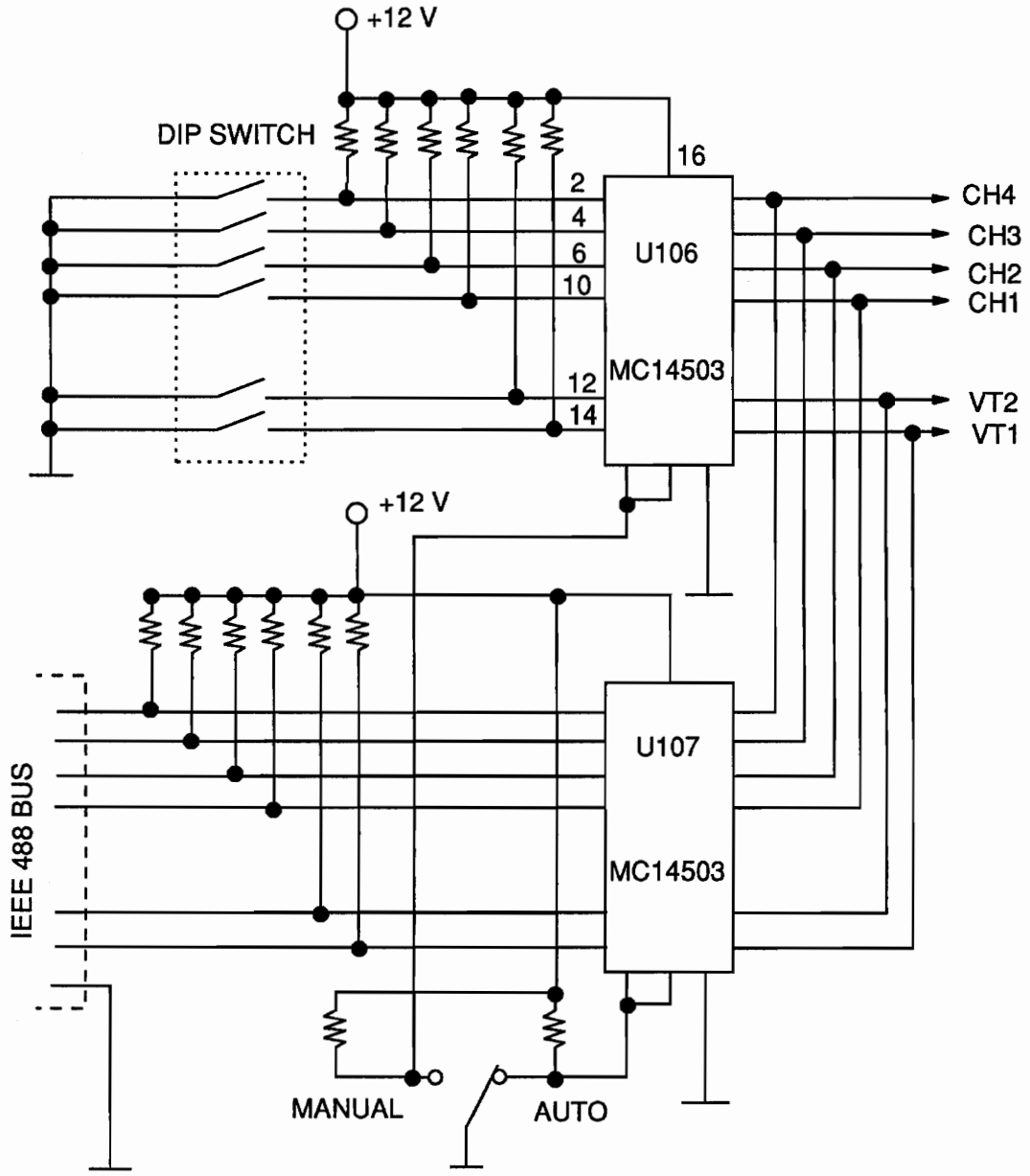


Figure 3.2 Circuit Detail of Mode Switching

Chapter 4 Software Design

4.1 Software Configuration

There are two sets of software provided for the system. One is a keyboard-input user-controlled program, and the other is a battery simulator real-time control program. In fact, the first one is a menu-driven program designed to allow the user to run the testbed unit by unit, step by step. After one or more parameters have been given from the keyboard, the computer will convert the parameter(s) into an internal value and command the proper element of the testbed to operate under the desired condition.

The real-time control software here means that the input and output of the control routine which controls the elements of the testbed, in our case the battery simulators, are the real amounts of voltage or current which are functions of real time. This routine can be easily developed to be an orbit simulator control software with the pro-

grammable control ability of load profile, solar array temperature, and illumination profiles.

4.2 Menu-driven Program for the User-controlled Operation

This program is used when the testbed is operated step by step, following the user's command. The main functions of the menu-driven program are:

1. Self-test:

   to test power devices and power supplies of the testbed.

2. Controlling and monitoring of operations of each element of the testbed:

   to allow the user to change the control parameters of the testbed elements individually.

3. Data acquisition:

   during the testbed operation, to record the specified variables upon the user request.

The details of these features will be given in the following sections.

The main procedures of this program can be briefly described as follows:

1. Initialization.

2. Showing the main menu and waiting for the user input for a selection of the power unit to be controlled.

3. Down to the submenu, displaying the current status (values of the parameters) of the selected testbed element and waiting for the user to input a new command or change the parameters.

4. Interfacing with the the user via keyboard.

5. Interpretation of the information input from the keyboard if it is defined. If not, ignoring the undefined input.

6. Communicating with the desired control unit of the testbed on the IEEE-488 bus line.

7. Commanding the testbed element to carry out an operation.

8. Checking the procedures or results of operations when necessary.

9. Updating the display on the screen.

10. Taking the testbed unit off the line and getting ready to receive the next input or to return to the main menu.

There are corresponding subroutines to control and communicate with each power control unit. Some subroutines, such as keyboard interface and display routine, are designed as common calls.

4.2.1 Initialization

The control parameters of the most testbed hardware elements need to be initialized before the testbed operation starts. Two power supplies for the battery simulators and one power supply for the solar array simulator are set to zero voltage and zero current output. Battery dischargers are set to 100 percent discharger rate, and chargers start with 0.85-ampere charge rate. The solar array simulator is initialized with $-20^{0}C$ of temperature and zero illumination. The load is controlled to 5.2 ampere, and all of the switches in the MUX are open. The DMM is set to DC voltage reading with auto-range. The self-test circuit in the solar array simulator control board is required to be off (commanded with zero).

After initialization, all units are off the IEEE-488 bus line by UNLISTEN command and ready to receive the command input from the user.

4.2.2 Main Menu

After the initialization is completed, the main menu is shown on the screen to user, and the software is ready for the user's input. Each time an input comes, a subroutine for interfacing with the user through the keyboard will interpret the input code.

If the input is appropriate and the value is within the desired range, the input is recognized as a command and it will be processed.

The units which need to be controlled are:

1. battery dischargers #1 and #2,

2. battery chargers #1 and #2,

3. a solar array simulator,

4. a multiplexer unit and a multimeter,

5. two battery simulator power supplies,

6. a Lambda power supply used for the solar array simulator,

7. load profile via a function generator.

For each of these hardware units to be controlled, there is a submenu to indicate to the user which unit is being selected and what the current values of the control parameters are. The program follows the user's selection and goes down to the submenu and waits for receiving the control parameters given by the user. The keyboard interface subroutine is called again to process the user's input code. If a correct input is received, the user's command will be carried out immediately.

The entire power system can be reset to the power-on state by invoking the initialization subroutine upon the user's request.

4.2.3 Battery Discharger Control

There are four various operation conditions that can be chosen for dischargers:

1. 100 % discharge bias,

2. 90% discharge bias,

3. 80% discharge bias,

4. discharger being shut down.

A two-bit digital-to-analog converter circuit has been built to convert the digital commands that come from the controller(PC) into an analog voltage signal to control the bias of the discharger.

The data table is given below:

command code	function
0	no effect to the discharger (100 % discharge)
3	10% down
4	20% down
6	being shut down
others	illegal

4.2.4 Battery Charger Control

The control variable is the charge rate. A logic circuit on the charger control board is implemented to receive the digital code from the data bus and send it to an analog control circuit to change the charge rate.

The range of the rate is from 0.85 to 23.0 ampere discretized by 16 values. The parameter table is:

charger current(AMPS)	command code (in hex)	if input is between (A)
0.85	0H	0.20 - 1.58
2.33	1H	1.59 - 3.07
3.80	2H	3.08 - 4.54
5.28	3H	4.55 - 6.02
6.76	4H	6.03 - 7.50
8.23	5H	7.51 - 8.99
9.71	6H	9.00 - 10.46
11.1	7H	10.47 - 11.95

12.7	8H	11.96 - 13.42
14.1	9H	13.43 - 14.91
15.6	AH	14.92 - 16.39
17.1	BH	16.40 - 17.86
18.6	CH	17.87 - 19.34
20.0	DH	19.35 - 20.83
21.5	EH	20.84 - 22.31
23.0	FH	22.32 - 23.50

For convenience, the user is asked to input an expected value of charge rate, from 0.2 to 23.5 ampere. The software can figure out which one of 16 discrete values in the above table is the closest to the user-expected one, and output the value code to the charger controller by IEEE-488 bus. The user can read the actual charge rate from the monitor.

4.2.5 Solar Array Simulator Control

For the solar array simulator, there are two parameters which need to be controlled by the computer:

1. temperature,

2. illumination.

The details are described in Chapter 6.

4.2.6 Communication with the Measurement Instruments

The following is the address table for the instruments used in the testbed:

address	unit
0	PC (controller)
16	Micro-dap used for Sorensen power supply control
17	MUX
23	DMM
24	Multifunction synthesizer
271	Lambda power supply used for solar array simulator

4.2.6.1 Digital Multimeter and Multiplexer

A multiplexer unit PM21 is used in the system, since there are so many voltage and current measurement points which have to be read during the testbed operation and self-test procedure.

A HP3478A digital multimeter has been chosen as a major measurement instrument. In Cooperation with a multiplexer PM21, it will perform a very important part of data acquisition in this power system.

In order to avoid short circuit due to the failure of the switches inside the multiplexer, each switching execution of the MUX is performed after a test. The test is performed by the software which starts with the opening of all switches. If no voltage can be read from the DMM, the performance will continue. Otherwise, the software will give the user error message and wait for the user's decision of whether the performance should go on or quit.

The software for each reading should complete the following jobs:

1. putting the MUX on the Bus line (make MUX active),

2. resetting MUX if necessary,

3. opening all the switches of the MUX,

4. swapping the DMM on line (remote mode),

5. selecting measure function and telling the DMM to send the reading(s),

6. waiting a moment and checking if the voltage reading is less than 0.1 V. If not, showing error message and waiting for user's response,

7. swapping the MUX on line once more,

8. commanding the MUX to do the switch operations,

9. swapping the DMM on line and measuring the signal values,

10. receiving the data coming from the DMM through the IEEE-488 bus,

11. putting DMM off line.

A subroutine is used for MUX check. Due to the high input impedance of the DMM, after each measurement, a short waiting time is needed to finish the check.

In order to obtain a stable voltage reading, during each measurement, a subroutine to obtain the reading from the DMM will be repeated until the difference between two readings is less than 0.2 V, or other value, depending on the degree of precision.

4.2.6.2 Function Generator

This instrument is for the control of the load simulator profile. The input command can be between zero and 50 amperes. The software converts this command to the actual output of the function generator in DC value with the ratio of

1 volt = 5.12-ampere load.

The updated setting will be shown on the monitor of the computer.

The application of this unit is limited here, since the generator used is not able to generate AC signal plus DC offset. This problem can be solved by adding a D-to-A

converter whose output is added to the generator, so the complex programmed load profile can be gained for orbit simulation or transition analysis.

4.2.7 Battery Simulator Power Supplies Control

There are two Sorensen power supplies installed to implement battery simulators in the testbed. An interface/conversion unit called Sorensen MICRO-DAC is selected to connect the Sorensen power supplies to the IEEE-488 bus and make them computer-controllable. This MICRO-DAC has two channels for output analog control signal and an 8-bit logical output for shutting down the power supplies or switching off some specified circuits.

Soft voltage adjustment is introduced to each updated voltage setting of the power supplies by the software control. Time delay subroutine is invoked each time to make the voltage increase gradually. A diagnostic routine can be executed during this interval and check if there is something out of order. In this way, the voltage overshot can be eliminated, and the operation can be stopped any time when something goes wrong before high voltage is applied.

4.3 Self Test Routine

As a precaution, and to endow the testbed with a fast troubleshooting ability, self-test, or diagnostics was introduced, and a dedicated software routine was designed to perform this procedure.

A total of three self-test procedures were designed. They are for battery simulators power devices, solar array power devices, and auxiliary power supplies, respectively. The self test for solar array simulator will be discussed in Chapter 6.

This self-test can be started from the main menu by the user. Figure 4.1 shows its flow chart. The steps are:

1. test all auxiliary power supplies and show the voltages,

2. ask the user to put the switches on the rack in the proper positions (Normal or Self Test) and to disconnect the cables,

3. ask the user to be sure everything is ready to test battery simulator power devices,

4. set the Sorensen power supplies of the battery simulators to +30 volt,

5. set the Lambda power supply to +38 volt,

6. operate the MUX and DMM to check the values of the test points as well as voltage across the load resistor R_L,

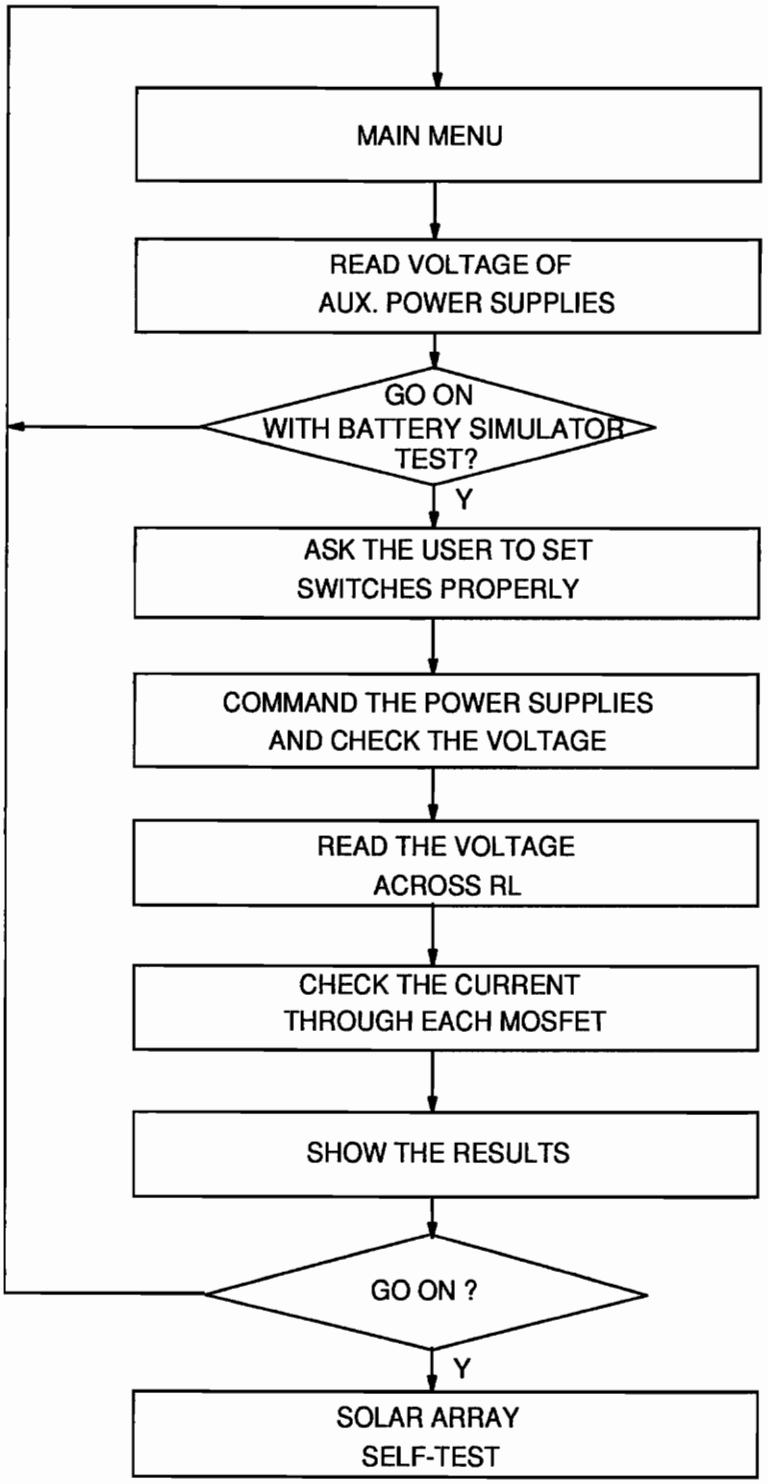


Figure 4.1 Flow Chart of Self Test

7. display the results on the screen,

8. ask the user whether to proceed with the devices test of the solar array simulators,

9. ask whether to go back to the main menu or to repeat the test.

4.4 Data Acquisition Capabilities

For data acquisition, this control/data acquisition system should be able to

- obtain results during the period of testbed operation,

- collect data from the measurement points during the self-test.

4.4.1 Configuration of the DMM and MUX

Data acquisition functions call for the application of a MUX unit and a DMM, optionally a gain/phase analyzer and a digital scope. Connection to a scope has been considered in the MUX application design, and a peripheral circuit for gain/phase measurement of the testbed has been built. The MUX unit, Philips system 21, used in this testbed, consists of an IEEE/IEC interface unit and three universal switch units. Each switch unit has 20 switches with the maximum rating of 30 Vrms switching voltage and 200 mArms switching current. The interface unit, also called master unit, pro-

vides communication between the IEEE-488 bus controller and the system 21 and passes the control signals to the specified switch unit. The power supply for the entire unit is in the master unit.

With the connection of the DMM and MUX, a total of more than 50 voltages and currents of the testbed can be measured, and the data can be read and written into files by the software. Almost 30 of the measurement points are connected to the MOSFET power devices of the battery simulators and dedicated to the self-test.

The limitation for the data collection rate lies in the handshaking timing of the system 21 and the waiting time for gaining the stable reading by the DMM. If a high-rate data collection is required, the multi-channel A-to-D converter, like ADC0809, should be used to greatly reduce the time needed for swapping the channel and waiting for the voltage to stabilize.

4.4.2 Data Collection for the Self-Test

Programmed self-test is a very convenient approach to checking the power system devices, power supplies, and some hardware functions by the computer before the testbed operation starts.

With a software control, two power supplies for the battery simulators and all auxiliary power supplies can be tested continuously. All the measurement results are shown on the monitor of the computer.

There are three test/run switches for switching of the power supplies output connection for the different applications. Two of them are mounted on the panel of the battery simulator racks (left), and the third is on the right rack (battery simulator). Before any test begins, the user should follow step by step the software instructions shown on the monitor, specifying which position the switches should be flipped to.

Three kinds of self-test are available so far. They are:

1. Power supply test

   The outputs of the power supplies for the battery simulator and all of the auxiliary power supplies are connected to mux#1 and #2 via a voltage divider so that their values can be read by the computer at any time. The auxiliary power supplies to be tested are:

   a. +15 V for discharger/charger #1,

   b. +15 V for discharger/charger #2,

   c. +15 V for PCU,

   d. +28 V for SSU,

   e. +5 V for display panel.

2. Battery simulator MOSFETs test

In this test for each simulator, one Lambda power supply is connected to the output of the simulator through two run/test switches on the racks. Sorensen power supply and the voltage across R_L of the simulator can be measured via a pair of switches in MUX #1.

Each group of 12 switches of the mux #1 and mux #2 is connected to the battery simulators #1 and #2, respectively, for self-testing. The measure points are at the drains of each FET referring to the positive of the power supply since the voltage across the drain load resistor can indicate the state of the FET.

3. Solar array simulator devices test

The mux #3 is mainly for data acquisition at normal power system test operation. By mux #3's switch connection, the voltage and current of power system 120 V bus and two battery simulators, and the current of the solar array simulator can be read through the voltage dividers or current shunts.

4.4.3 Network Analysis

The system has an option to use a gain-phase/impedance analyzer to obtain the power system loop gain and impedance through a peripheral measurement circuit. A subroutine is designed to control the analyzer operations under a default setting. The results can be transferred to the computer and saved to files.

Figure 4.2 shows the flow chart for analyzer control.

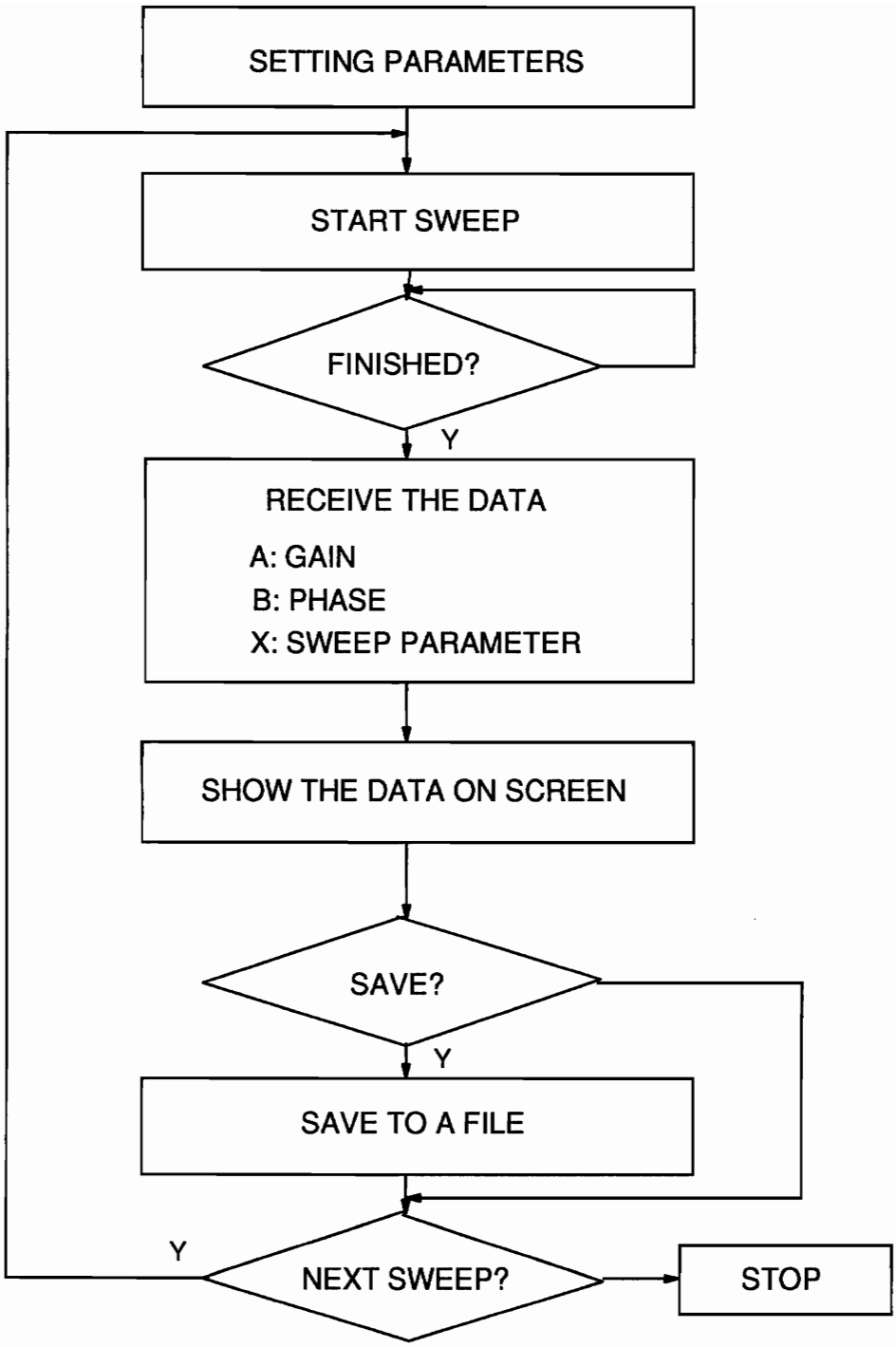


Figure 4.2 Flow Chart of Analyzer Control

The data can be transferred, processed and displayed on the screen, so the user can check them page by page (25 lines of data per page). Upon the user's request, data can be saved to the desired subdirectory of the computer with the desired file name. The program has an ability to stop automatically at the end of data transfer, even though the amount of the data may vary, since each time the number of data points can be set by the user.

Network analyzer output impedance programmed settings are:

analyzer: HP4194A

start frequency: 100 Hz

stop frequency: 100 KHz

sweep mode: log

reference channel input impedance: 1M

average time : 8

osc level : 15dBm

number of measurement points: 61

integration time : medium

Chapter 5 A Nickel-Hydrogen Battery Simulator Real-time Control for the NASA EOS Testbed

5.1 Introduction

In this chapter, the hardware and software design of a Nickel-Hydrogen (Ni-H2) Battery Simulator (BS) with application to the NASA Earth Observation System (EOS) satellite is discussed. The battery simulator is developed as a part of a complete testbed for the EOS satellite power system. The battery simulator involves both hardware and software components. The hardware component includes the capability of sourcing and sinking current at a constant programmable voltage. The software component includes the capability of monitoring the battery's ampere-hours (Ah) and programming the battery voltage according to an empirical model of the nickel-hydrogen battery stored in a computer.

This chapter shows the design of a practical, computer controlled Ni-H2 Battery Simulator suitable for use in the NASA EOS satellite. The simulator is used to simulate real-time battery behavior during different charge or discharge periods. Both simulated and experimental results are presented.

Hardware and modeling work were done by Mr. Zvi Gur and Mr. Ashok Patil. Section 5.3 and 5.4 are included here only for providing a complete documentation about the battery simulator.

5.2 Specifications

The batteries that have been selected for the NASA EOS satellite have 54 nickel-hydrogen series cells, rated at 54 Ah each.

The voltage and current specifications are given below:

Voltage Range:	53 -84 volt
Maximum Source Current:	38 A @ 53 volt
	24 A @ 84 volt
Maximum Current Sink:	23 A

Total Power (max): 2000 watt

5.3 Hardware Configuration

Figure 5.1 shows a schematic diagram of the battery simulator circuitry. It consists of an IEEE-488-controlled power supply and a P-channel MOSFET current sink. When sourcing current, diode D1 is forward-biased, and the MOSFETs are all off. When sinking current, the diode back-biases by a voltage equal to about the gate threshold of the FETs. The MOSFETs operate in the linear mode and provide a shunt path to ground through resistor R_L. Resistor R_L is sized so that at the minimum battery voltage of 53 V and the maximum battery charge current of 23 A, the MOSFETs remain in the linear range. A value of 2.27 Ω is used for R_L. Eleven 25-Ω, 250-W resistors in parallel are employed. The maximum power dissipation in the load bank is 1200 W. Twelve IRFP9240 P-channel MOSFETs in parallel are employed in order to spread the power dissipation. Worst-case power dissipation in the MOSFETs occurs at the maximum battery voltage and charge current. The total power dissipation under these conditions is 708 W. Resistors R1-R12 are provided as ballast to insure current sharing between the MOSFETs. The resistors are trimmed so that at the peak power dissipation point, the MOSFETs share current precisely. The worst-case power dissipation for any one MOSFET is about 80 W.

The components are mounted on two heat sinks of 3000 sq. in. each. The worst-case temperature rise after 1 hour of operation at full load is 52°C.

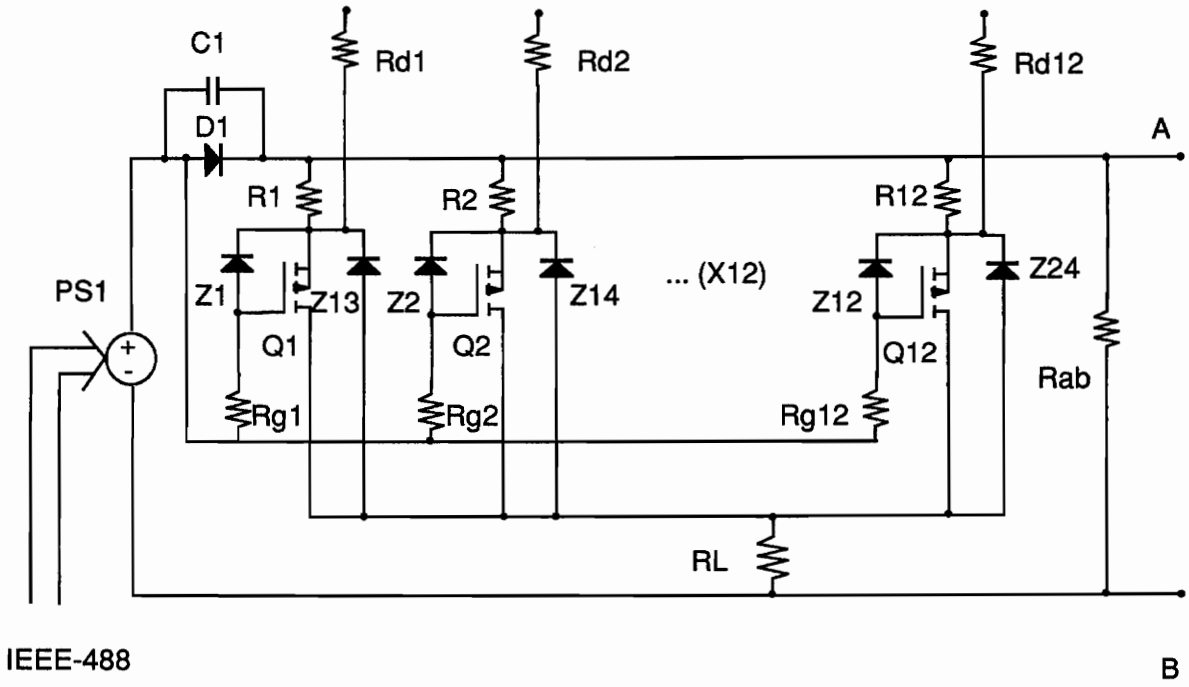


Figure 5.1 Battery Simulator Schematic

5.4 Ni-H2 Software Model for a Ni-H2 Battery

A model of a Nickel-Cadmium (Ni-Cd) battery was previously presented [5]. This model is shown in Figure 5.2. In this model, R1 represents the plate internal resistance. The back-to-front connected diodes D1, D2 represent the hysteresis and non-linear internal resistance effects in the electrochemical cell. They are a function of the current direction (charge/discharge) and the depth of discharge (DOD). The diode D3 represents the charge leakage and overcharge conductive leakage. C1 represents the effect of linear capacitive elements storage across the double layer, and C2 represents the effect of the chemical reaction storage.

The equations that describe the components of the model have been derived in Zimmerman and Peterson's paper [5], and are presented briefly here.

The equations for the diodes D1, D2, D3 are:

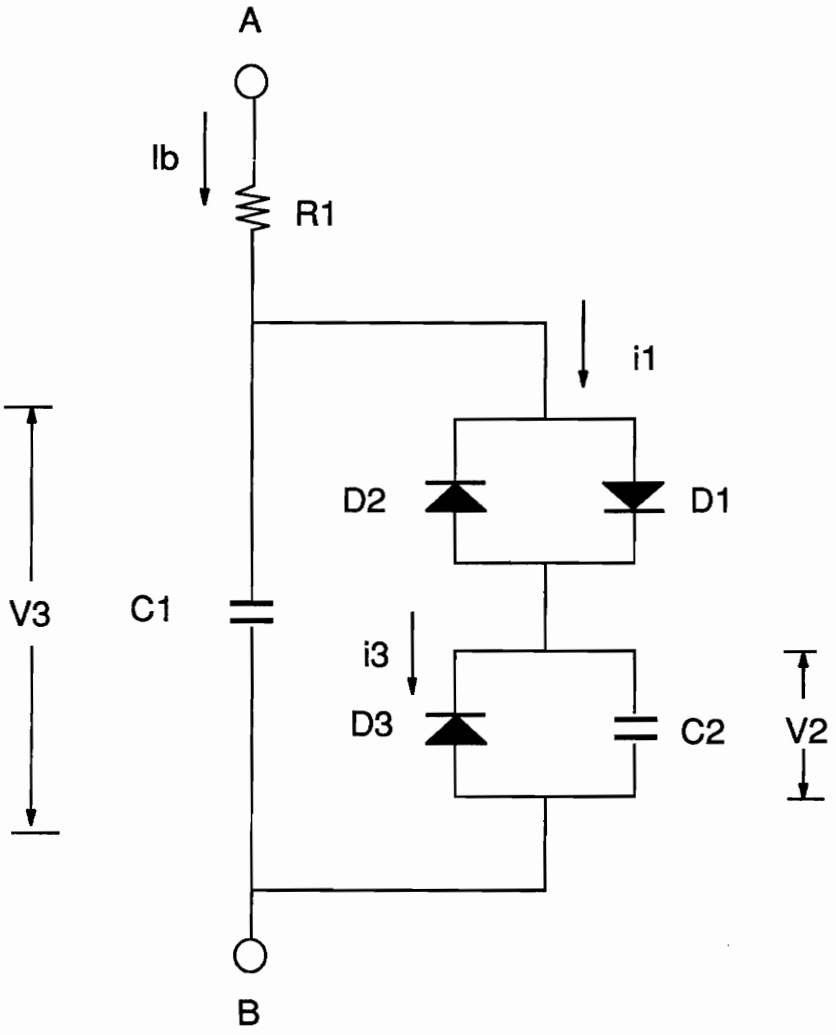
$$I_1 = 2k_1 \sinh(k_3 V_1), \tag{1}$$

$$k_1 = \frac{I_0 e^{40.27 k_2 (1 - \frac{519}{T+459})}}{e^{40.27 \frac{V_0}{L}} - 1}, \tag{2}$$

$$k_3 = \frac{20900}{L(T+459)}, \tag{3}$$

where

I_0 = reference diode current [A] @ 60°F and V_0,



R1 Internal Resistance

C1 Double Layer Charge

C2 Reactive Storage

D1, D2 Double Layer Potential

D3 Self Discharge

Figure 5.2 Ni-Cd Battery Model

I_1 = actual diode current [A] @ T and V_1,

V_0 = reference diode voltage [V] @ 60°F and I_0,

V_1 = actual diode voltage [V],

T = actual diode temperature [F],

k_2, L = constants, which may be determined by analysis of the experimental data.

The equations for the capacitors C1 and C2 are:

$$C = Ae^{-B(V - V_M)^2} + D, \tag{4}$$

where

C = capacitance value [F] @ V,

V = actual voltage level [V] of capacitance,

V_M = median voltage level [V],

A = capacitance in [F] @ $V = V_M$,

B = distribution constant,

D = minimum capacitance [F].

The Ni-H2 battery model is taken to be the same as Ni-Cd with slightly different parameter values. The charge leakage (self discharge) of a nickel-hydrogen battery is

greater than that of a nickel-cadmium battery. Other parameters of the Ni-Cd model are modified to match published Ni-H2 battery data[5], [6]. Table 5.1 shows the new values.

Table 5.1 45 Ah Ni-Cd and 54 Ah Ni-H2 Battery Parameters (Per Cell)

PARAMETER	Ni-Cd model	Ni-H2 model
capacity [Ah]	45	54
R1 (milliohm)	0.5	1.2167
C1 -- A [farad]	80000	96000
C1 -- B	120	120
C1 -- D [farad]	5000	6000
C1 -- V_M [V]	1.348	1.324
C2 -- A [farad]	2050000	2460000
C2 -- B	400	400
C2 -- D [farad]	5000	6000
C2 -- V_M [V]	1.348	1.324
D1,D2 -- V_0 [V]	0.025	0.025
-- I_0 [A]	5	5
-- k_2	0.8	0.8
-- L	1.5	1.5
D3 -- V_0 [V]	1.348	1.324
-- I_0 [A]	0.22	0.264
-- K_2	1.35	1.35
-- L	1.5	1.5

Model software simulations are performed with the use of the EASY5 dynamic systems analysis software. A comparison of published [5],[6] and software-simulated data by the battery model shows there is an excellent agreement between the two. The empirical data generated from the EASY5 simulation is entered in the EOS power system testbed control computer.

5.5 Battery Simulator Integration In The EOS Satellite Power System

The battery simulator is integrated in the EOS Satellite power system testbed to represent the dynamic operation of a Ni-H2 battery during the complete Low Earth Orbit (LEO) cycle. The LEO cycle has been taken to be 90 minutes, 55 minutes illuminated (charge) and 35 minutes eclipsed (discharge).

Figure 5.3 shows the battery simulator integration. A Personal Computer (PC) as the controller runs the program to drive two battery simulators in real time simultaneously. A multiplexer (MUX), a digital multimeter (DMM), and a power supply controller (MicroDAP) are controlled via an IEEE 488 computer operated communication bus.

The direction (charge/discharge) and amplitude of the actual battery current are retrieved from a voltage across a meter shunt. The shunt voltage is one of a number of testbed voltages that are connected to the DMM through the MUX. The information is retrieved from the DMM and passed to the program as the input.

The outputs of the battery simulator, the variables controlled by the program, are battery's voltage (VBatt) and battery's Ah. The battery voltage is programmed into the power supply controller (Micro-Dap) which controls two power supplies of the simulators. Ampere-hours is another output which is a running parameter sent to a display panel.

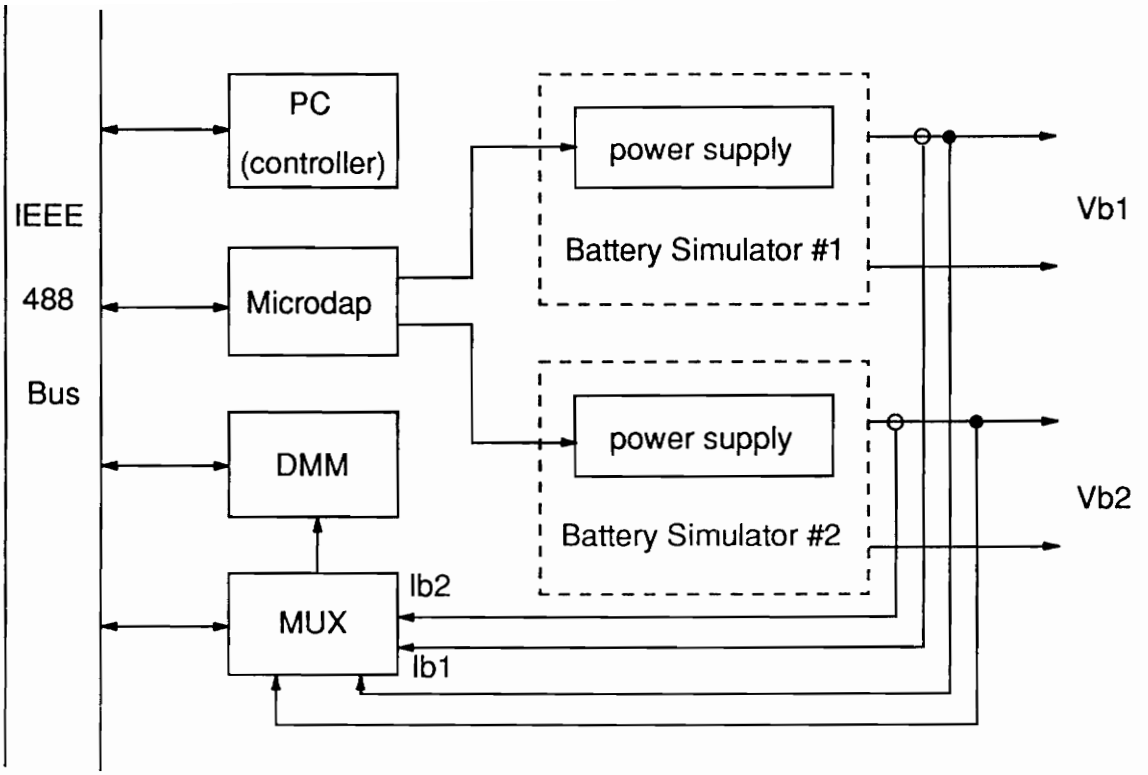


Figure 5.3 Configuration of the Battery Simulator

5.6 Algorithms

The battery model can be expressed in a state-space form as:

$$\frac{dx_1}{dt} = (i_b - i_1)/C1(t), \tag{5}$$

$$\frac{dx_2}{dt} = (i_1 - i_3)/C2(t), \tag{6}$$

$$\frac{dx_3}{dt} = i_b/3600, \tag{7}$$

where (see figure 5.2, equivalent circuit)

$x_1 = V_{c1}$, voltage across C1,

$x_2 = V_{c2}$, voltage across C2,

x_3 =ampere-hour,

i_1 = actual diode current of D1 at T and actual diode voltage V_1,

i_3 = actual diode current of D3 at T and actual diode voltage V_2,

i_b = actual battery current.

On the right sides, i_1, i_3, C1, and C2 are functions of V_{C1} and V_{C2}. This is a non-linear time-varying system. Two possible approaches to solving the system equations can be considered in this case.

The first algorithm is the traditional Runge-Kutta (R-K) method, which can be applied in the first- and the fourth-order forms.

5.6.1 The First Order R-K

Solving for x_3, ampere-hour, is very straightforward. We only focus on x_1 and x_2 and let $y = x_1$ and $z = x_2$. We have the system given by

$$y' = f(t,y(t),z(t)), \quad y(t_0) = y_{0,}$$

$$z' = g(t,y(t),z(t)), \quad z(t_0) = z_{0.}$$

This numerical method, the first-order Runge-Kutta, can obtain the approximate result from

$$y(t_{i+1}) = y(t_i) + \Delta t y' = y(t_i) + \Delta t f(t_i,y(t_i)),$$

$$z(t_{i+1}) = z(t_i) + \Delta t z' = z(t_i) + \Delta t g(t_i,y(t_i)).$$

Since the accumulated error generated by each iteration will be getting more significant along with the increase of time, long period of simulation using the first-order R-K is not recommended. But it is the easiest way to solve the equation.

5.6.2 The Fourth Order R-K

Here the algorithm is

$$y_{i+1} = y_i + \Delta \frac{t}{6}(K_1 + 2K_2 + 2K_3 + K_4),$$

$$z_{i+1} = z_i + \Delta \frac{t}{6}(L_1 + 2L_2 + 2L_3 + L_4),$$

where

$$K_1 = f(t_i, y_i, z_i),$$

$$K_2 = f(t_{i+\frac{1}{2}}, y_i + \Delta \frac{t}{2} K_1, z_i + \Delta \frac{t}{2} L_1),$$

$$K_3 = f(t_{i+\frac{1}{2}}, y_i + \Delta \frac{t}{2} K_2, z_i + \Delta \frac{t}{2} L_2),$$

$$K_4 = f(t_{i+\frac{1}{2}}, y_i + \Delta t K_3, z_i + \Delta t L_3),$$

$$L_1 = g(t_i, y_i, z_i),$$

$$L_2 = g(t_{i+\frac{1}{2}}, y_i + \Delta \frac{t}{2} K_1, z_i + \Delta \frac{t}{2} L_1),$$

$$L_3 = g(t_{i+\frac{1}{2}}, y_i + \Delta \frac{t}{2} K_2, z_i + \Delta \frac{t}{2} L_2),$$

$$L_4 = g(t_{i+\frac{1}{2}}, y_i + \Delta t K_3, z_i + \Delta t L_3).$$

the advantage of this algorithm is that the generated error is smaller than in the first-order R-K even at a large-step iteration because more information about the input signal is used at each step.

5.6.3 Linearized State-Space Form Method

Another approach is a simplified discrete state-space expression which is linearized at each trajectory point. It is known that the non-linear state equations of a system in the vector-matrix form are:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \tag{8}$$

where x(t) represents the n x 1 state vector, u(t) the p x 1 input vector, and f(x(t),u(t)) is a n x 1 function vector. Generally, f is a function of both the state vector and the input vector.

Based on a Taylor series about $x(t) = x_0(t)$, the truncated linearization at a nominal operating point x_0 is

$$\Delta \dot{x}_i \simeq \sum_{j=1}^{j=n} \frac{\partial f_i}{\partial x_j} \Big|_{[x_0,\, u_0]} \Delta x_j + \sum_{j=1}^{j=p} \frac{\partial f_i}{\partial u_j} \Big|_{[x_0,\, u_0]} \Delta u_j, \tag{9}$$

where

$\Delta x_i = x_i - x_{0i}$,

$\Delta u_i = u_i - u_{0i}$.

Then

$\Delta \dot{x}_i = \dot{x}_i - \dot{x}_{0i}$.

Note that

$\dot{x}_{0i} = f_i(x_0, u_0)$.

Since the input here is the battery current, i_b,

$$f(x(t),u(t)) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} (i_b - i_1)/C1 \\ (i_1 - i_3)/C3 \end{bmatrix}. \tag{10}$$

Note that the output of the system is the voltage across the C1 so the system representation can be written as

$$\Delta \dot{x} \simeq A\Delta x + b\Delta i_b, \tag{11}$$

$$\Delta y = C\Delta x, \tag{12}$$

where

$$A = \begin{bmatrix} \frac{\partial}{\partial x_1}\frac{i_b - i_1}{C1(t)} & \frac{\partial}{\partial x_2}\frac{i_b - i_1}{C1(t)} \\ \frac{\partial}{\partial x_1}\frac{i_1 - i_3}{C2(t)} & \frac{\partial}{\partial x_2}\frac{i_1 - i_3}{C2(t)} \end{bmatrix}, \tag{13}$$

$$b = \begin{bmatrix} \frac{\partial f_1}{\partial i_b} \\ \frac{\partial f_2}{\partial i_b} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial i_b} \frac{i_b - i_1}{C1(t)} \\ \frac{\partial}{\partial i_b} \frac{i_1 - i_3}{C2(t)} \end{bmatrix} = \begin{bmatrix} \frac{1}{C1(t)} \\ 0 \end{bmatrix}, \tag{14}$$

$$C = [1 \quad 0], \tag{15}$$

$$\Delta x = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}. \tag{16}$$

In this way, the increment at each operating point can be calculated. It should be reiterated that A and b are evaluated at each point, that is, matrix A and b are time varying.

To gain the whole operating trajectory, we first need to linearize the equation at each interactive point, and add the increment to the previous value.

Secondly, we need a vector-matrix discrete state equation. For control software, it is possible to discretize the system with a time increment during which the time-varying parameters do not vary appreciably. This makes the problem become one of solving a set of linear time-varying discrete state equations.

In the software, one method used for discretizing the system (8) is to approximate the derivative of $\Delta x(t)$ by

$$\Delta\dot{x}(t) \simeq \frac{1}{\Delta t}\,(x((k+1)\Delta t) - x(k\Delta t)). \tag{17}$$

Writing

$$x(k\Delta t) = x_k,$$

where Δt is a small time interval, We can approximate the state equation (11) by the time-varying difference equation

$$x_{k+1} = x_k + (I + \Delta t A_k)\Delta x_k + \Delta t b_k(\Delta i_b)_k \tag{18}$$

over the time interval $k\Delta t$ to $(k+1)\Delta t$, where i_b is the input (the actual battery current), and x_{k+1} are the voltages across C1 and C2. The first element is also the output (the battery voltage) at the (k + 1)-th sampling time.

5.7 Simulation of the Different Algorithms

In order to get conceptual knowledge for the software design of battery simulator's controller or driver, and to verify the performance of the control algorithm, it is necessary to simulate the battery software model by different mathematical methods. Two pieces of software have been written for this. The first one solves the equations by the fourth order R-K, using very small iteration interval, 0.001 second, which can be thought of as leading to the exact solution, which can be compared with the results obtained by the simulator's control software. The second simulation program solves the equations by three different ways under different time intervals of iteration and

different input signals. Both programs are written in FORTRAN and use double precision.

The input signals for the simulation are a square waveform and a sinusoidal waveform. Both amplitudes are 64.8 A peak-to-peak, and frequencies are 0.05Hz. That is, the simulator changes mode from charge to discharge periodically. The square waveform has much faster changing rate than the sine waveform, so it is the better test signal. The total simulation period is 90 minutes, which is equal to an orbit period. In order to study the effects of different iteration intervals, simulation runs at different time intervals: 1s, 5s, 10s, and 20 seconds.

Three major criteria are used for the selection of the control algorithm of the battery simulator:

1. tolerance,

2. stability,

3. simplicity.

5.7.1 Tolerance

Following the simulation results, table 5.2 shows the maximum error generated by different algorithms at the fastest varying rate of the input signal of the (100*k)-th seconds (k = 1, 2, ..., 54). These maxima occurred during the transient state of the battery simulator.

Table 5.2 Maximum Error at (100*k)-th Second of Simulation

under sinusoidal waveform input

interval	the fourth R-K	the first R-K	linearized
1s	0.00053794	0.00188040	0.11952092
5s	0.00268364	0.00871483	0.08881443
10s	0.00538356	0.01602477	0.21692113
20s	0.01190450	0.03024666	0.19004582

under square waveform input

interval	the fourth R-K	the first R-K	linearized
1s	0.00737676	0.01022876	0.10999942
5s	0.03404386	0.05411162	0.15567620
10s	0.05883502	0.10751661	0.28970683
20s	0.19157655	--	--

Obviously, the errors under square stimulation are ten times larger than those under sine waveform.

When the input signal is a square, infinite differential of the signal may cause the computational overflow or underflow. But in the practical application, the slope of the input signal cannot be infinite. One thing that should be noted is that the fast rate of signal varying could give the algorithm some trouble if the sampling rate is too slow.

From the above results, we can see that the errors tend to be relatively stable to a range. For the fourth order R-K, the error will reduce to 1/10 of that in the transient period under sine input, and to 1/7 of that under square input if the sampling interval is less than 20 seconds. But for the first order R-K, the error is reduced only by less than a half.

In the simulation, if the input signal is a sine waveform with a period of 100 second, the model will respond with a small amplitude of sine waveform, but the error will be larger than for a period of 200 seconds since the input signal changes faster.

Note that the results here present only the error voltage for each battery cell. Since we have 54 cells in the testbed, the total error will be 54 times the value shown in the table.

Here, the length of the data processed inside the computer should be considered. To reduce the calculation time, single precision is usually used, which will generate larger error than the double one used in our simulation.

5.7.2 Stability

A significant result obtained from the simulation is that the fourth order R-K is more stable than the others. In any of our algorithms, when the input signal varies fast and the sampling rate becomes slow, some intermediate value during computation could be out of range, overflow or underflow, which would make the simulation algorithm fail. This is due to the large error generation caused by less input information obtained between two iterations. The algorithm with a better stability is able to work at a large iteration step without failure.

In our case, under a square waveform input, only the fourth order R-K can work up to the sampling interval of 40 seconds, while under a sine input the interval can be as long as 100 second. But in the latter case, the results are so poor that they can hardly be useful.

The capacitors C1 and C2 in the software model of the battery are very large. This size shows that the model has a very large time constant. So the slope of the output voltage of the model cannot be large either. For this reason, we hardly see big differences between the first and the fourth order R-K algorithms when the sampling interval is less than 5 seconds. But this does not mean the fourth order R-K has less advantage than the first order R-K. If the simulated system does not have a large time constant, or the input signal changes fast, it can be seen that the performance of the fourth order R-K is much different from the others.

If longer sampling interval has to be used, the stability of the algorithm becomes the first consideration. If the sampling rate is less than 5 seconds, the simplicity is the major point to consider when choosing the most suitable algorithm.

5.7.3 Simplicity

The first-order method is the simplest one, but when battery current i_b varies at a fast rate, or the sampling rate is low, a large error will affect the performance. When sampling rate is longer than five seconds, the fourth order R-K may be the best choice if the execution time for the software is not critical. While the computing time required at each step is longer, in our case, the iteration step can be larger without greatly reducing the accuracy of the results.

After all, we can see that the fourth order R-K has the best results even at a large iteration interval. The first R-K can also be used if iteration interval is less than five seconds. Obviously, the linearization method makes the largest error and cannot be applied when the interval is larger than one second.

5.8 Software Design

As mentioned in section 5.3, the equation to describe the battery model can be solved recursively by the control program.

Figure 5.4 shows the flow chart of the program for battery simulators. The execution procedure of the software is:

1. Initialization

   The initial values of battery voltage, ampere-hours, and all parameters are given by the user, or default values are employed.

   The initial conditions are:

   $v_{10} = 1.480V$,

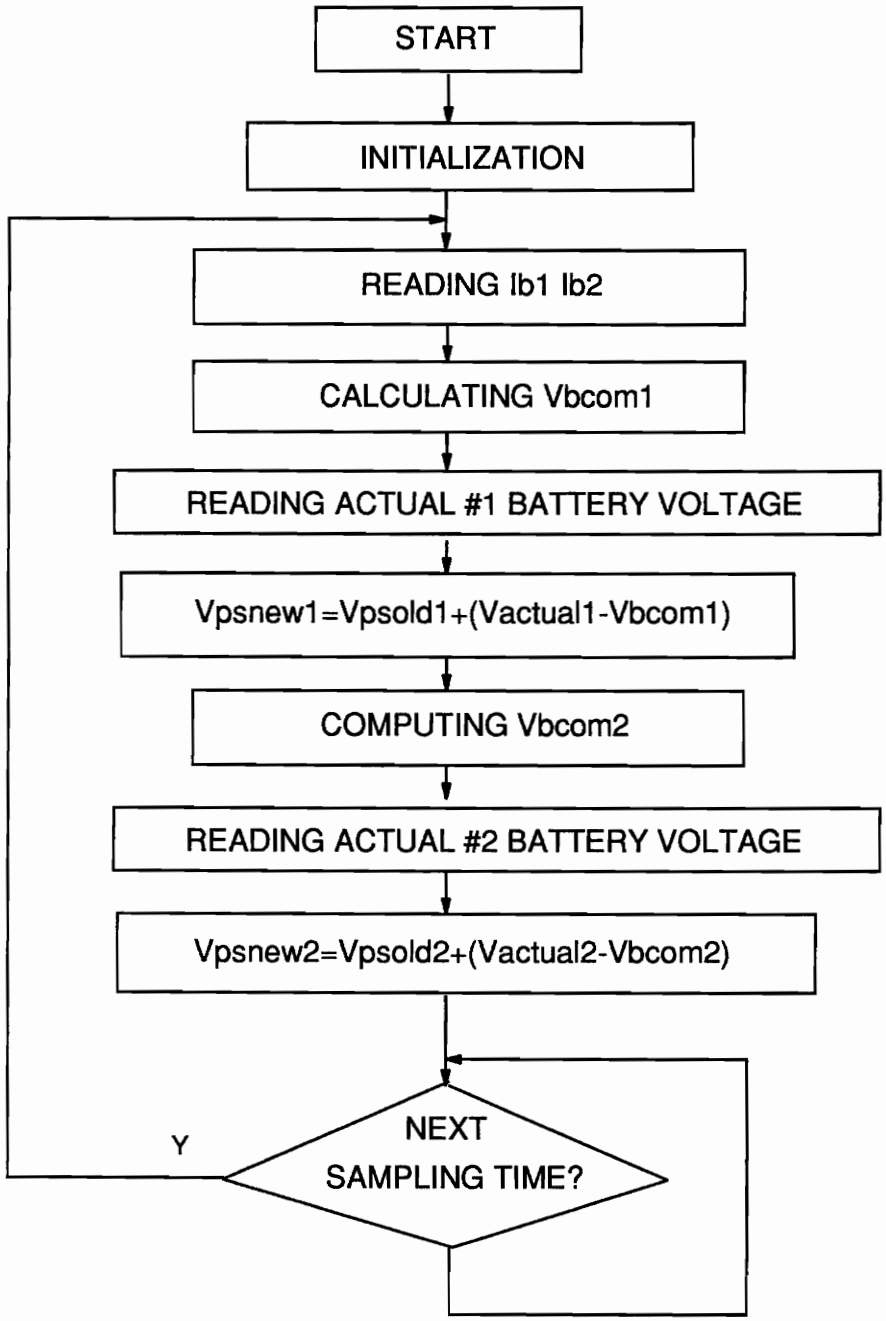
   $v_{20} = 1.400V$,

   temperature = 50°C,

   ampere-hour = 57 Ah,

   Constant values, such as series resistor per cell, median voltage, k_1, k_2, and sampling rate, are also established.

   Each instrument used in the system also needs initialization. The DMM is set to a proper range of DC readings to get maximum resolution. The MUX has to be set to the desired connections. Power supplies need setting to zero output before any operation starts. At the same time, the program also checks if these instruments give correct responses. The warrant message will show the user if any instrument fails to operate.



Vps -- Voltage Commmand Sent to the Power Supply

Vbcom -- Computed Value of the Battery

Figure 5.4 Flow Chart of Battery Simulator Program

The graphic work also needs some initialization. The scale, color, and graphic frame and grid position should be calculated and drawn.

For safety reasons, the soft start of the power supplies is introduced to make the voltage increase slowly from zero to the expected value, and a diagnostics subroutine could be invoked to check for a possible hardware problem. The program halts when a problem is detected.

After these checking and setting operations, the software stops and asks the user whether to go on. If the answer is no, the software will go back to the beginning and repeat the steps.

2. Main loop

The main control loop starts with the measurement of the battery current, i_b, whose polarity indicates whether the simulator is in discharge mode or in charge mode. Using the chosen algorithm, the software calculates the new battery output voltages and the new ampere hours.

To get proper readings of the battery current and voltage at the time of each sampling, the computer will read voltage values more than once and compare them. If the difference between two consecutive readings is less than a range, the computer will think it is a stable reading. Otherwise, the computer will read the values again, continuing the process until a stable voltage reading has been obtained. The DMM has been set to the continuous reading mode, and its sampling rate is much faster than the routine's sampling rate, so we are sure the correct data will be read.

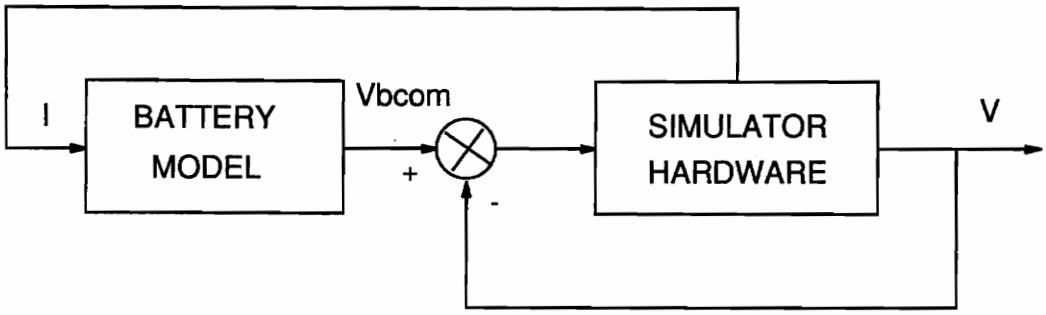
But on the other hand, the repeat time has to be limited in case of the program being dead looping. A digital filter can be implemented here. An average of several consecutive readings is the simplest digital filtering.

The capacitors C1 and C2 are time-varying; therefore, each time they need updating.

Errors exist due to line drops and the hardware configuration; they also result from digital control system characteristics. The current battery voltage is measured, an error voltage is established from this measurement, and the voltage value is computed according to the battery software model. The software determines a new programming voltage of the power supply with the compensation of the error voltage. So, the control loop is closed by the software. The control scheme is shown in Figure 5.5.

When the mode switches from discharge to charge or vice versa, an abrupt voltage change appears at the output of the simulator. This change is caused by the voltage drop on FETs. The control software detects this change at the next sampling moment and compensates for it. The sampling rate used in this system is 0.2 Hz if two simulators are driven simultaneously. Thus, the maximum time period an error exists would be equal to the sampling interval, 5 seconds.

There would be two approaches to reduce these unexpected peaks. The first is done by a hardware, putting the power supplies in remote mode with sensing the output of the simulator instead of the output of itself. The second approach is compensation by software. Each time the battery simulator changes mode from charge to discharge or vice versa, the routine automatically compensates the



Vbcom -- Computed Value for the Battery Simulator

I ---------- Battery Current

V ---------- Actual Output Voltage of the Battery Simulator

Figure 5.5 Battery Simulator Control Scheme

approximate voltage drop of the MOSFETs. The complete elimination of the peak is done at next sampling time, when the routine obtains the feedback information.

The time constant of the Ni-H2 battery simulator is much larger than 5 seconds, since C1 and C2 exist. The sampling interval can be relatively longer than in a normal real-time control system. This allows the option of running the simulator in expanded or contracted real time. A 90-minute orbit can be simulated in 5 minutes, if there is no special requirement for the shape of battery current, i_b.

3. Graphic display

The software can draw the simulation trajectories for two simulators on the monitor screen while the simulation is going on. Combined with the on-screen display of other test results, this feature allows the user to see every test process during the whole cycle.

The software is written in Quick Basic. The timer inside PC is used as a timing reference for this real-time control.

5.9 Choice of Sampling Rate

Since IEEE-488 bus system is not originally designed for real-time control application, some IEEE-488-programmable instruments have very slow operation rate. In our case, the execution time of the control software is much shorter than the time spent

waiting for some measurement instruments to finish their operations. For instance, the MUX needs 25 milliseconds for each switching operation and, therefore, requires hundreds of millisecond of IEEE-488 bus timing. Due to the fixed configuration of the DMM and MUX employment, the operation of IEEE-488 bus will slow down because the MUX is involved. The other major factor is a very high input impedance of the DMM: it also needs time to obtain a stable voltage reading when a change is made between measurement points with a large difference in voltages, since there is a natural capacity resident in the measurement input circuit and cable. To overcome this delay, the program takes more than 4 seconds to finish a sampling loop. To cut down the voltage transient time so that the waiting time can be reduced, the following sequence is used:

1. measure i_{b1}, in mV range (via a current shunt),

2. measure i_{b2},in mV range (via a current shunt),

3. measure V_{b1}, about 7 to 10 volts (via a divider),

4. measure V_{b2}, about 7 to 10 volts (via a divider).

In this way, the number of voltage transients during the measurement will be reduced to two for each loop. If the signal of battery current can be amplified to the range of 7 to 10 volt, the waiting time for transient would be greatly shortened.

As we know from the simulation results in section 5.7, the time constant of the battery model is very large, so the performance cannot be improved easily only by the in-

crease of the sampling rate. On the other hand, the input signal i_b varying rate may not be very fast in a real application. Of course, spikes are a common occurrence on the power bus, but the model in fact behaves like a low pass filter and cannot respond to the spikes with very small period even when they have been detected. So the sampling rate of 5 to 20 seconds is good enough for our application.

The sampling rate will be 0.5 Hz if only one battery simulator is to be controlled, and 0.2 Hz for two simulators run simultaneously.

5.10 Time Scale

Time scale k_s may be introduced to contract the period of the entire power system test cycle for an orbit. The software sees the simulation process as k_s times the actual time, and switches the battery's modes or applies other appropriate profiles according to k_s.

Two approaches can be used in software to vary time scale. One is to vary the time constant in the software model and keep the iteration interval the same as normally. The other is to increase the time interval of each iteration step, but not the real sampling rate. For instance, with the real time interval of 5 seconds, the iteration step is seen as 20 seconds by the software. Thus, k_s is 4. So, an orbit simulation of total 90 minutes can be finished within 22.5 minutes. If K_s is 16, an orbit test can be finished within 5 minutes, which means the iteration step in the software is 80 seconds. One thing that should be noticed is that the time scale of the input signal is

also changed proportionally to K_s by the software. That is, if the frequency of the input signal is 1 Hz in real time, it will be seen as 1/80 Hz by the software if K_s is 16.

By using the first method, one can obtain more accurate results. But more work has to be done in modifying the model, since the time constant in the model is time-varying (mostly defined by C1 and C2). Thus, the second way is much easier.

To keep the same accuracy, one software method could be introduced. When the time scale is large, more information about the input signal is needed. The software can be designed to incorporate more than one time of the input signal sampling for an output. With more sampling data of the input signal, surely a much better result can be obtained. Even with the first order R-K, the performance will be improved a lot.

Chapter 6 Solar Array Simulator Control

6.1. General Description

A solar array simulator has been designed and implemented for this EOS testbed. The simulator is able to simulate accurately the DC and AC characteristics of a high-voltage array over the illumination and temperature ranges which represent the real operation conditions of the solar array in the space. This simulator makes accurate testing of power stage hardware of this EOS satellite power system possible, either when an array is not available or under serious test conditions which may be dangerous to real arrays.

6.2. Specifications

22 strings for simulation of 384 8cm by 8cm solar cells in series.

Short circuit current 2.6 A at full (100 percent) illumination.

Open circuit voltage 180 V at 75°C.

Illumination variation range from 0 percent to 100 percent.

Temperature variation range from −20°C to 80° C.

Availability of computer/manual control.

Built-in diagnostics circuit operated by computer.

The I-V curves for the solar array to be simulated are shown in Figures 6.1 and 6.2, which present the characteristics of each string vs. illumination and temperature, respectively.

6.3. Hardware Design

The simulator hardware has been designed by Mr. Steve J. Butler[4]. The simulator hardware consists of 22 identical string modules, a control module and a diagnostic module. The power is provided by three series-connected LAMBDA 70A 60V power

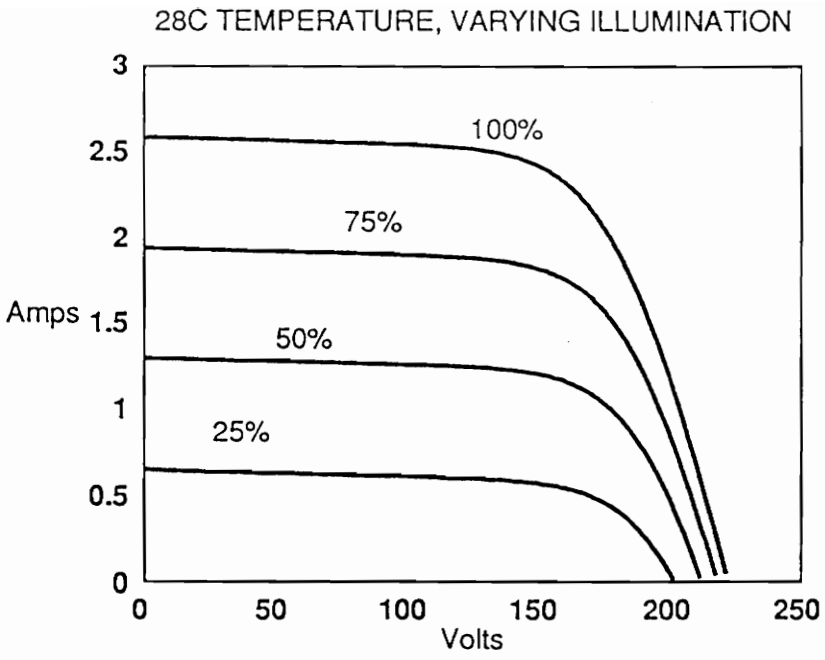


Figure 6.1 Solar Array I-V Characteristics (I)[9]

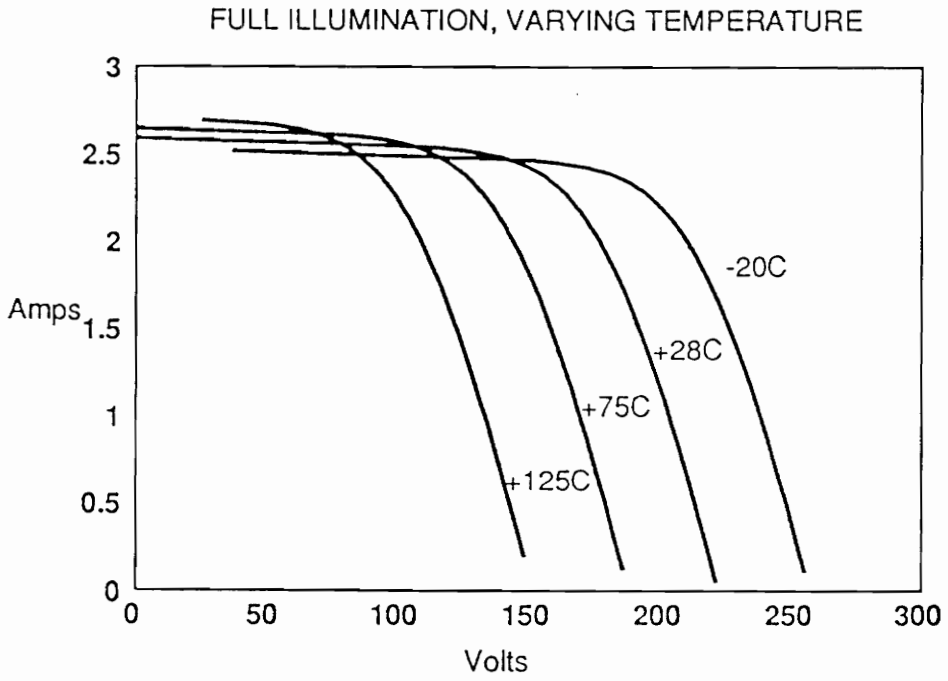


Figure 6.2 Solar Array I-V Characteristics (II)[9]

supplies; therefore, the I-V curves shown in Figure 6.1 are not simulated when the voltage exceeds 180 V, which corresponds to the temperature lower than 75° C with open circuit. The same situation applies to Figure 6.2 at lower temperatures. Each string module simulates one array string and has 8 MOSFETs as a current regulator with each MOSFET's power dissipation of less than 62 W. One common control module provides the control signals, illumination, and temperature references to the string modules. The diagnostics circuit is used for automatic test of the MOSFETs by the computer. Interface with the computer, which includes two digital-to-analog converters and a digital decoder, facilitates computer control functions through an IEEE-488 Bus.

6.4. Control Scheme and Hardware Implementation

The schematic diagram of the string module of the simulator is shown in Figure 6.3. The main control task is the regulation of the output current and voltage which should satisfy desired I-V curves shown in Figures 6.1 and 6.2. To implement this control scheme, a dedicated control module was designed to carry on the operation commanded by the computer or manual input, whereas current feedback loops and voltage feedback loops were employed in each power stage hardware module to form the closed-loop control.

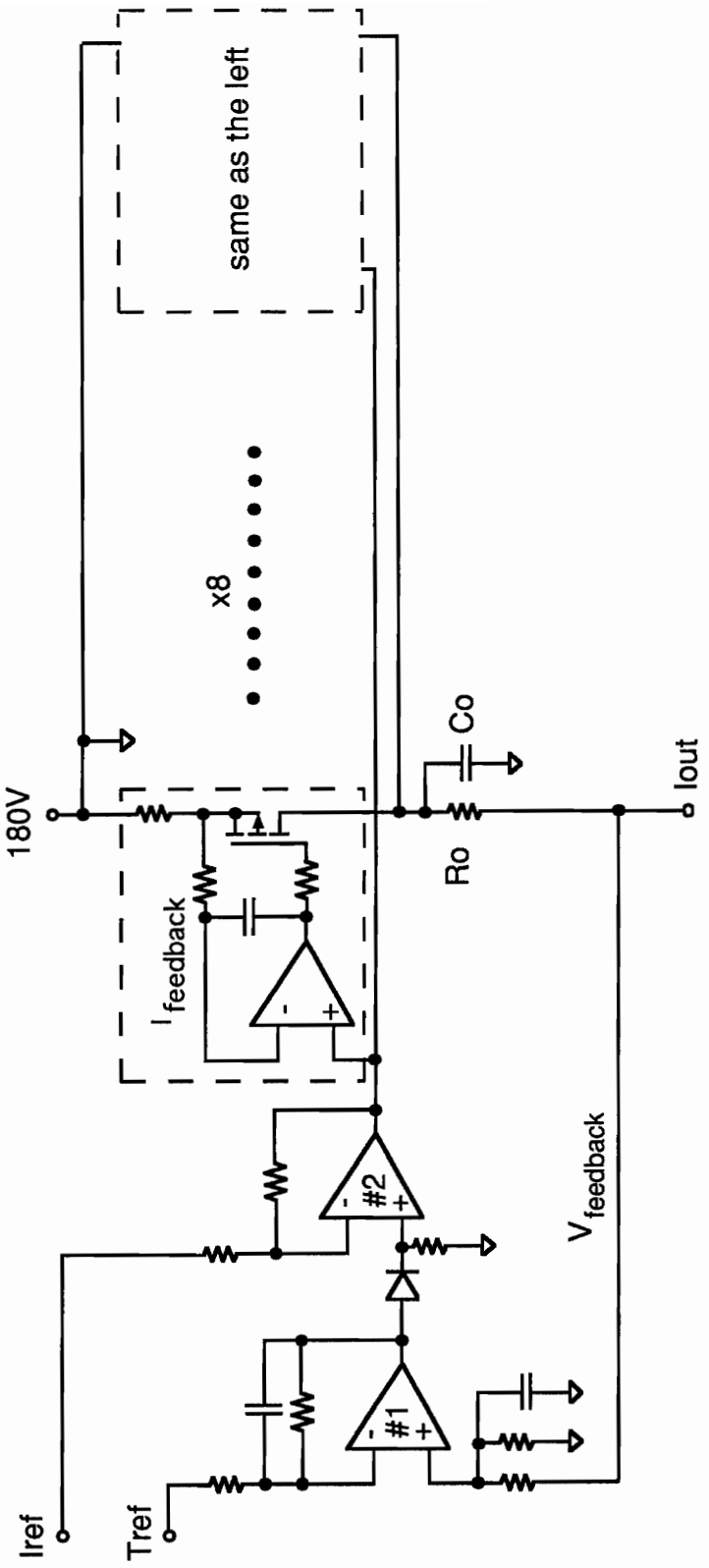


Figure 6.3 Schematic of String Module

6.4.1 Control Module

The control module is in charge of both manual and computer control of temperature and illumination simulation. The hardware consists of two digital-to-analog converters, and level shifters. The signal after shifter is fed to the gates of MOSFETs to control the MOSFETs' current which is the output of the solar array simulator.

The control signals come from either the computer or the manual dip switch, and are processed by the control module, which is able to receive digital commands with six bits over two channels, and to convert them to two analog signals by digital-to-analog converters resident in the module. Two level shifters are used to reference these analog control signals to 180 V, since the signal levels which come from either manual control or computer control are low: from 0 to 5 V. For this reason, a floating auxiliary supply voltage is used for control amplifiers. The output of the control module is sent to 22 string modules, and this signal is the combination of temperature control command and illumination control command.

6.4.2 Illumination and Temperature Control

As a matter of fact, both illumination control and temperature control are achieved using the same approach, which is by varying the reference voltage which regulates the bias of the MOSFETs to obtain the desired current and voltage output. Illumination control signal, IREF, affects the simulator directly through op amp #2 (see Figure 6.3). But the temperature signal input, TREF, is directed to another op amp, #1, after subtracting from the feedback voltage of the simulator output. This error

signal goes through a diode and then is added to the illumination control signal by op amp #2. The output of op amp #2 is fed to the MOSFETs (8 per string module) as the final control bias voltage. In this way, the feedback voltage of the simulator output does not linearly react to the control input, since the nonlinear diode characteristic provides the current cutback at specified output voltage point. Therefore, a nonlinear array I-V curve can be duplicated well.

6.4.3 Current loop

To satisfy the output range of each string from short to open circuit, the MOSFET current regulators are located on the high side of 180 V supply. When short circuit is to be simulated, the string has to drop 180 V while providing short current, which requires 500-W power dissipation. This power is shared by eight MOSFETs to prevent any of MOSFETs' failure from increasing the current in the other MOSFETs; this way, the system can be more reliable. Each MOSFET has its own current feedback loop to maintain this MOSFET's output as a constant based on the reference input only. Also, the feedback loop is compensated with an integrator to make a tight DC control. The response speed of this loop should be fast enough so that the entire system transient response would not be affected by this loop.

6.4.4 Voltage Loop

To reproduce the I-V curve, each module has one voltage loop which is used to decrease the current reference with increasing output voltage. So the string I-V curve

can be accurately simulated by using a diode which has a nonlinear I-V characteristic.

6.5 Diagnostics

There are a total of 176 MOSFETs in the simulator. They are designed so that if any one of them fails (becomes either open or short), the remaining modules will continue their normal operation, and the failure will not be noticed, since the current loops are individual. Self-diagnostic circuit is built specially for fast and easy detection of these unnoticed device failures.

The devices in the simulator can be completely tested module by module. Figure 6.4 shows the block diagram of the diagnostic module. One n-channel MOSFET is used for each string module. So a total of 22 MOSFETs are controlled by the computer through the IEEE-488 interface. During a normal operation, they are open. When the computer runs a solar array MOSFET test routine, these 22 MOSFETs can be individually turned on by digital commands and all 176 MOSFETs can be tested group by group. Being limited to 3A by a common 2.2-ohm resistor, the current will flow through the MOSFETs which have been turned on with the known rate set by the computer. Then the computer routine examines this current in each module and compares it with the desired value. In case of a difference, the computer will notify the user that at least one of the MOSFETs may have been damaged. The software can also tell the user in which module the possible damage is located. Eight

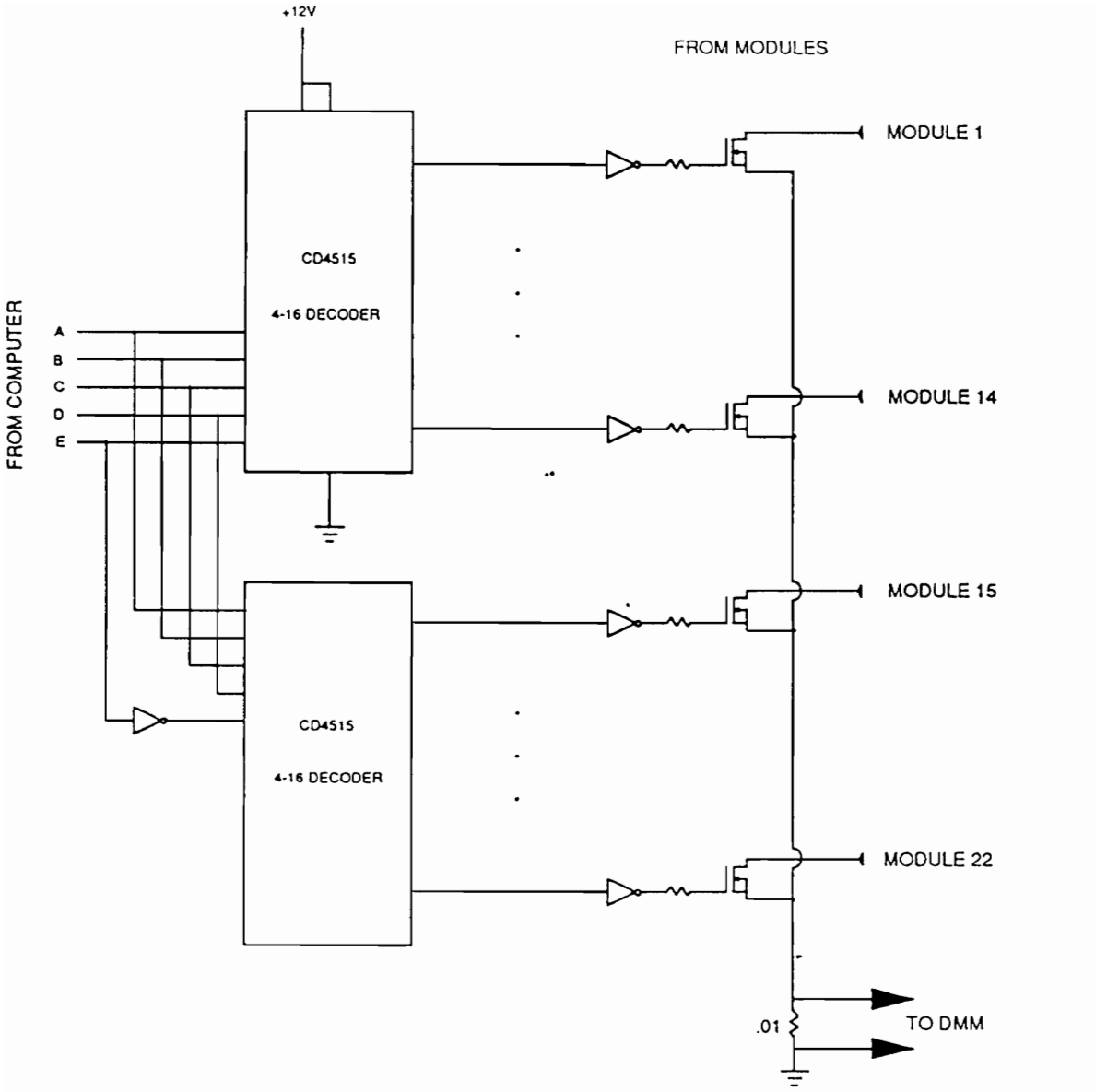


Figure 6.4 Diagnostic Circuit[9]

MOSFETs of this module have to be checked manually to determine which one has been damaged.

6.6 Software Control of the Solar Array Simulator

For the solar array simulator, there are two parameters which need to be controlled by the computer:

1. Temperature,

2. Illumination.

The temperature range is from -20^0 C to 125^0 C, and the illumination is from zero percent (total eclipse) to one hundred percent. These values are set by the user for simulation of different sections of an orbit. Since the actual illumination command sent to the simulator is the function of temperature, each time the temperature has been changed, illumination has to be changed accordingly. Figure 6.5 shows the flow chart.

In the menu-driven mode, the software asks the user to input the temperature value first and then converts the input into the value from 29 to 58 (1D to 3A in hex) which represents a temperature from -20^0 C to 125^0 C, since negative value cannot be accepted by the hardware directly. This truncated integer value will be sent to the temperature controller of the solar array simulator. In the next step, the new illumi-

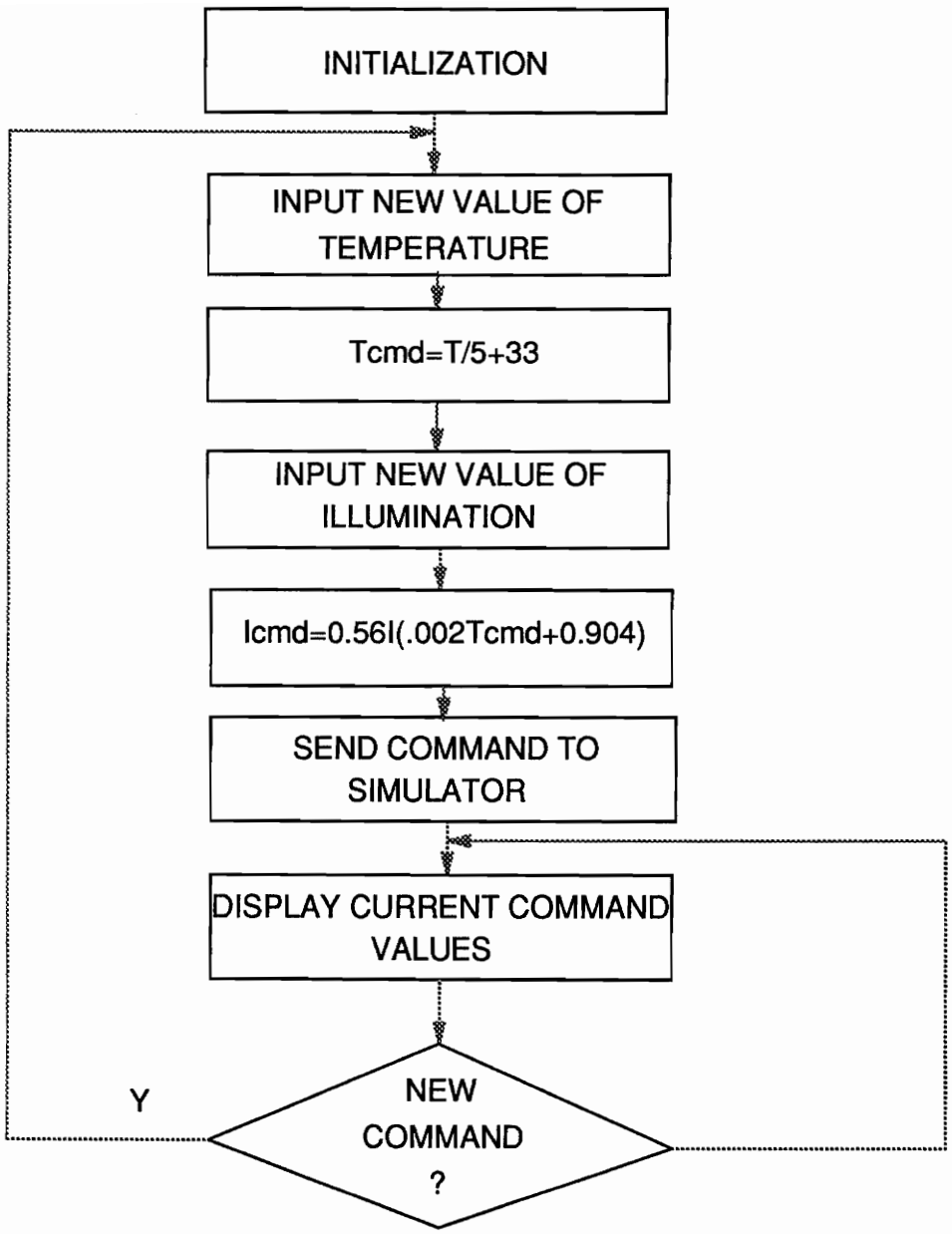


Figure 6.5 Flow Chart of Solar Array Simulator Control

nation value according to the updated temperature value will be calculated automatically and sent to the solar array simulator control unit, too.

The relationship between them is:

$$I_{com} = 0.56 I_L(0.002(T/5 + 33) + 0.904),$$

where

T is the temperature value in degree C, and

I_L is the illumination value in percentage.

I_{com} is the actual illumination command to be sent to the control circuit of the solar array simulator.

Then the software waits for receiving the input with a new illumination value. After the calculation of a new actual illumination command, the software sends a new signal to the solar array simulator and displays the new settings on the monitor.

The software will ask the user for a new illumination input after each temperature input. The user can press the ENTER key to skip any of the inputs and turn to the next one, and the skipped value will not be changed.

6.7 Orbit Simulation

Finally, an orbit simulation is the complete test of the testbed hardware function. By generating the illumination curve and the temperature curve in a software routine either using a look-up table or splined equations, one can obtain the entire orbit simulation for the power system.

The simulated data can be saved to a file for later analysis, and this feature allows hardware testing which would be almost impossible with a real solar array.

The experimental results are included in Chapter 7.

Chapter 7 Experimental Results

After many times of software debugging and hardware troubleshooting, all functions of both hardware and software mentioned in this thesis have been tested and worked well. Here is the list of the tests which have been done:

IEEE-488 custom interface hardware test

IEEE-488 bus test with the custom interface connected

Interface driver software test

Instruments on-line test

Solar Array interface test

Self-test mode function test

Battery simulator operation test by software control

Battery charger interface test

Battery discharger interface test

Solar array simulator operation test under software control

Real-time operation

Among the above, the two important tests are the interface hardware test and the real-time operation of the testbed, so this chapter will focus on these two.

7.1 Custom Interface Hardware Test

Without this circuit completely operational, nothing can be done to obtain computer control of the testbed operation. Therefore, hardware of the digital interface has to be functional first.

The interface board has been tested for the logic generation first. The next test is the IEEE-488 Bus system test which involves the active custom interface board with all the instruments being on the bus line.

This test is necessary because we have to know the compatibility of the interface with the instruments which are on the line and to see if there are any conflicts which will stop the bus from functioning or lead to the loss of data during data transfer.

The test was successful after an improvement of the interface board reset function was introduced. The addressed elements of the testbed are able to receive the command signals. The test showed that the timing of the bus data transfer was greatly slowed down after Philps MUX was put on line. This problem was caused by the design of the MUX. Later in the software design, it was found that asking MUX to be off-line during some time would increase the data transfer speed of the whole bus system.

Figure 7.1 shows the experimental waveforms with the IEEE standard timing waveforms (same as figure 2.2) as a comparison.

The top half of the figure shows two instances of data transfer, each involving one string of eight-bit data. Here only one of the DIO lines is shown, and it is "one" after "zero" (note that IEEE-488 uses negative logic). The second waveform is the DAV signal, which comes from the computer. The next two waveforms are those found on the lines of NRFD and NDAC, generated by the custom interface board.

The complete two cycles of IEEE-488 bus timing sequences can be seen. The data has been successfully transferred, and the timing sequence matches the IEEE-488 standard timing requirement.

7.2 One Battery Simulator Test with Solar Array Simulator

Having been tested for interface function with the computer, the solar array simulator is connected to the testbed and ready for the testbed operation test under the computer control.

Figure 7.2 shows experimental results of one battery simulator operation under the computer control. The top one in Figure 7.2 shows the real battery simulator current, i_b, generated by the battery charger or discharger according to the different mode. Positive values indicate the charger mode, while the negative ones occur in discharge mode. The initial output voltage of the battery simulator is set to 78.5 V by the

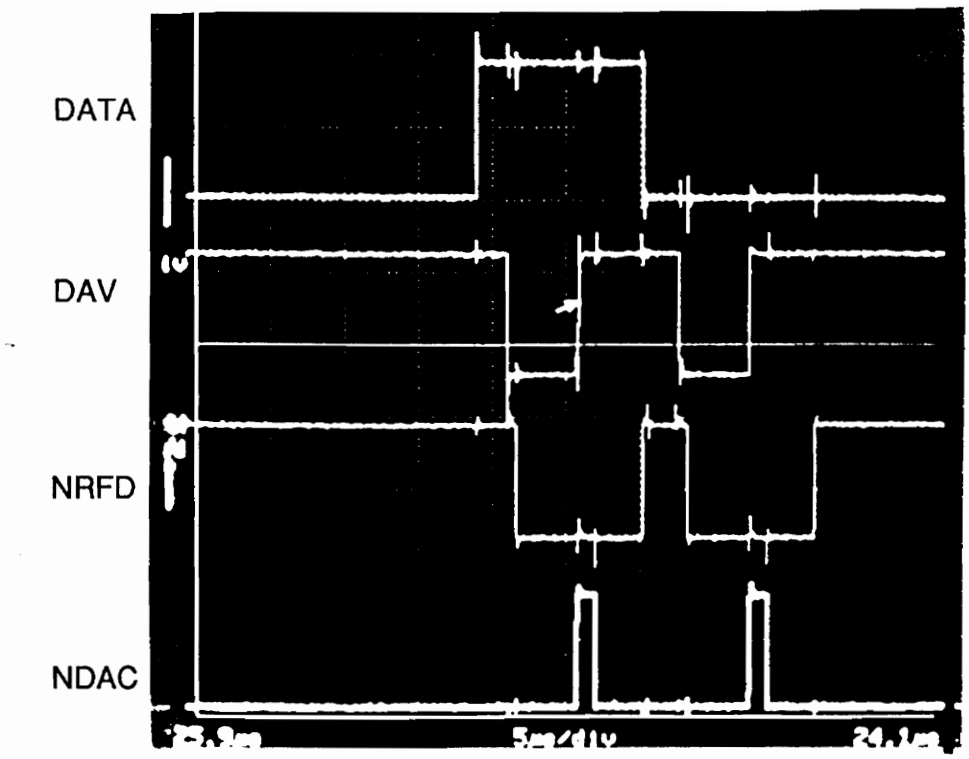


Figure 7.1a Experimental Timing sequence of the Interface Board

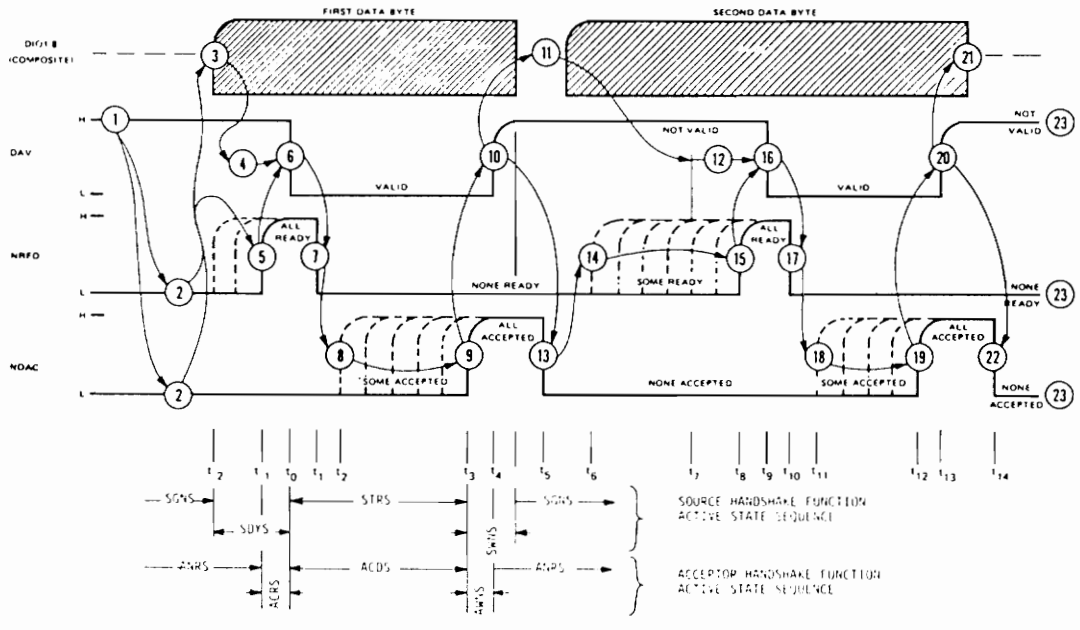


Figure 7.1b Standard Timing Sequence of IEEE-488 Bus[8]

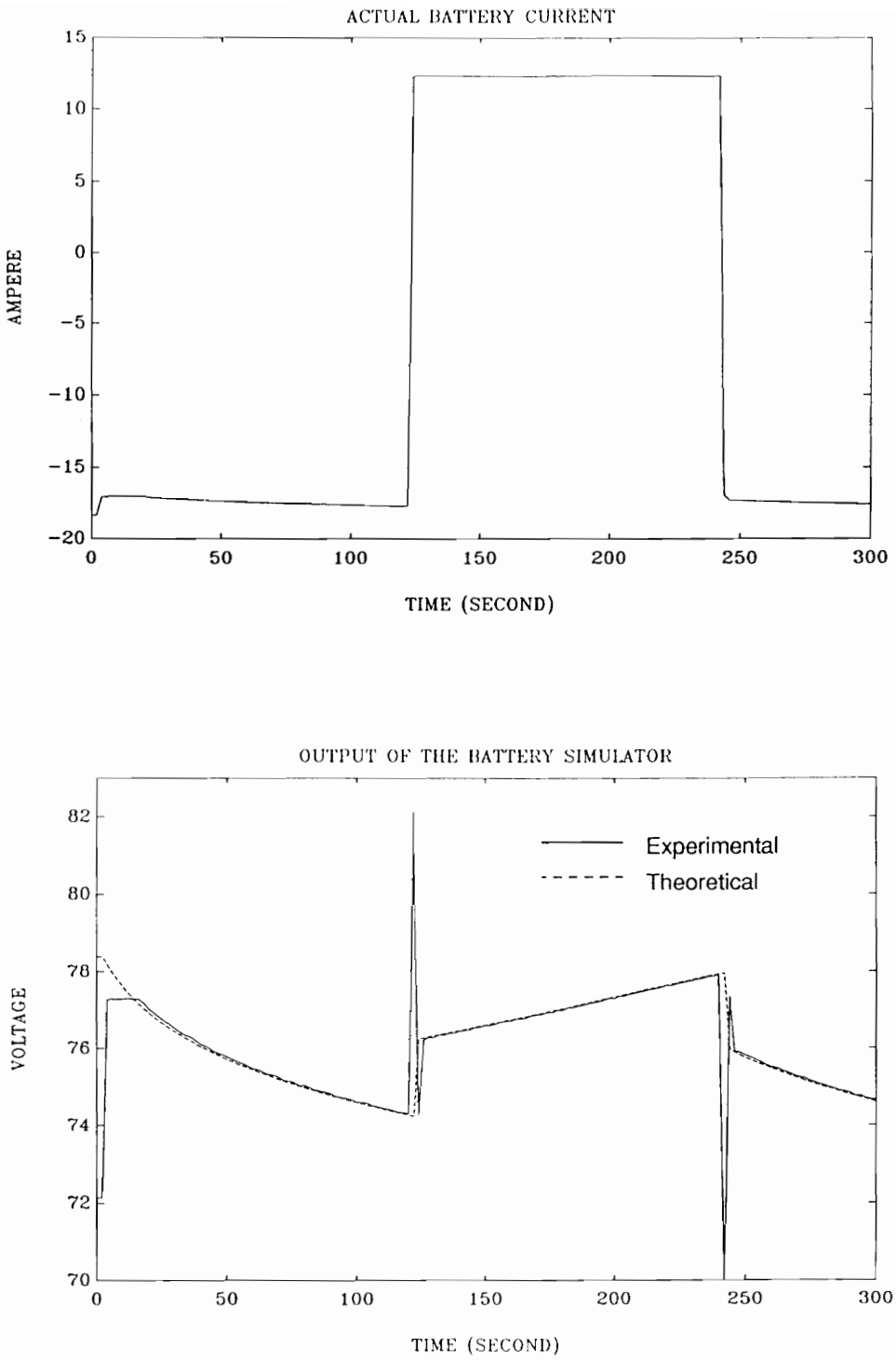


Figure 7.2 Experimental Results of Battery Simulator

control routine in order to be within the working range of the battery charger or discharger controlled by the PCU.

The bottom of Figure 7.2 shows the data record of the output voltage of the battery simulator. The battery simulator starts with the discharge mode at time t = 0 and remains in it till t = 120 seconds. Since the solar array simulator has been set to zero illumination, battery discharger regulates the bus voltage during these 120 seconds. In this period, the battery voltage decreases as expected. The charge current is 17.5 A and the battery voltage is reduced exponentially.

At t = 120 seconds, the mode is changed to the charge mode by setting the solar array simulator at 40 percent illumination, so the battery charger takes the responsibility for bus voltage regulation. Battery voltage jumps 2 V, which reflects the internal impedance of the battery, R_1 in Figure 5.2, and then increases exponentially according to the equation (5.6) in Chapter 5. Since the charge current is only 12.5 A, less than the previous discharge current, the rate of increment is less than the rate of decrement when in discharge mode (17.5 A). Finally, the battery simulator is back to the discharge mode at t = 240 seconds. A very good match with the theoretical simulation results of the Ni-Cd battery (dotted line) can be seen from the experiment.

Notice that there are four voltage peaks, which are equal to the voltage drop on the FETs(about 7 volts). The reason has been explained in Chapter 5. These unexpected peaks are eliminated soon by a new voltage command from the controller at the next sampling moment.

7.3 Testbed Power System Operation Test With the Computer Control

7.3.1 Set-Up of the Experiment

This is the entire power stage function test, which can be seen as pre-test for the orbit simulation test. Computer sets each element of the testbed in operation as follows:

battery charger: charge rate 10 A,

battery discharger: 100 percent discharge rate,

solar array: real-time control; here the square test signal of illumination, generated by the computer, is applied. Temperature has been set to zero degrees at the beginning,

battery simulator: real-time control; output voltage is the function of the battery current satisfying the equations in Chapter 5,

load simulator: 25-A load current in total,

total simulation time: 300 seconds.

7.3.2 Operation Experiment

Figure 7.3 shows the trajectories of two battery simulators' operation under simultaneous computer control. The top one in Figure 7.3 shows the real battery current output for both battery simulators. The middle one in Figure 7.5 shows the output voltage of battery simulator #1 and the bottom one in Figure 7.3 shows the same for #2. These voltages are functions of the current based on the Ni-Cd battery software model. The time bases of plots are synchronous.

The experiment results shown in the middle and bottom ones in Figure 7.3 can be explained as in the following section.

The solar array simulator starts with the setting of the simulation of zero percent illumination controlled by the computer so that the battery simulators start operation in discharge mode at this time. The discharge currents are 12 A and 13 A, respectively. The battery voltage is reduced exponentially during this period (0 to 120 seconds).

At t = 120 seconds, a 20-percent illumination command is sent to the solar array simulator, simulating the turning on of the solar array. This change causes the increase of the bus voltage, and the PCU brings the battery charger into operation. Therefore, the direction of battery current is reversed, and the battery simulator is switched into the charge mode.

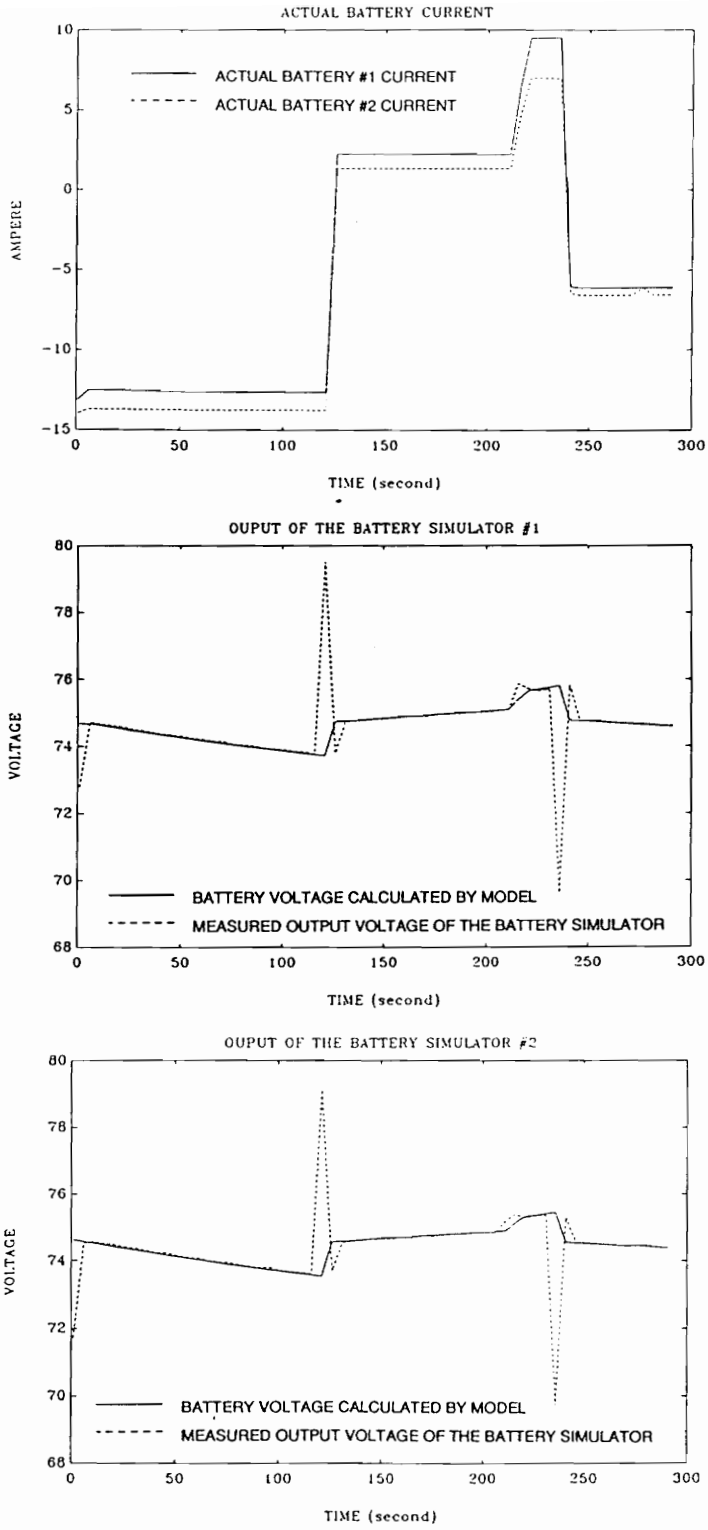


Figure 7.3 Trajectories of Two Battery Simulators' Control

At the moment of the mode change, the output voltage of the battery simulator rapidly jumps up by 5.5 volts due to the voltage drop on MOSFETs (see Figure 5.1), since the polarity of the battery current going through it has been reversed. The computer is not able to notice this jump until the next sampling moment.

At the next sampling point, the computer gets the feedback by sampling the voltage value and sends a correcting signal to the power supply, thus eliminating the error. This correcting signal is calculated by comparing the actual voltage with the desired value at this moment. Then the output of battery simulator reaches the theoretical value of 73.8 V, which is equal to the value of the previous sampling moment, whereas the actual value at this time should be 74.8 V. So we can see that the digital control system always has a one-step delay compared with the analog system. Therefore, if the sampling period is too long, and the input signal of the system changes fast, the system will be unstable.

At t = 210 seconds, the illumination of the solar array is increased to 60 percent. The ripple which can be seen about one volt is also caused by MOSFETs, since the current going through it varies. Soon after the computer corrects this error, the solar array simulator is turned off by the control routine (illumination is zero percent) and the battery simulator changes back to the discharge mode at t = 230 seconds. The spike appears once more and the process is the same as when the mode changes from the discharge mode to the charge mode, as mentioned before. The voltage of the battery simulator is going down because of energy dissipation.

Chapter 8 Conclusions

8.1 Advantages and Limitations

The control/data acquisition system has been implemented and tested. This system provides very efficient and convenient means for testbed operation. Many sophisticated test profiles and different test conditions can be programmed and systematically performed.

Some measurement instruments are not completely suitable for the system if more control or data acquisition functions are added to the system. They have very slow response rates, thus greatly limiting the entire system performance because a limited number of measurement points is a major impediment to the development of the system.

8.2 Recommendations for Future Study

One way we can overcome the above shortcomings is by using a multi-channel analog-to-digital convert board with the sampling rate of 40 to 60 kHz. It can also be combined with the MUX's operation. In this way, the computer can have much more time to take care of fancy functions like damage protection, etc.

Development of the network analyzer can make the measurement work more time-saving and more efficient.

Hooking a digital oscilloscope to the bus is part of the future work towards expanding the data acquisition capability. A digital scope like TEX 11401 can play an excellent role in data acquisition under computer control, since it has a large amount of memory and a built-in microprocessor-controlled data processing function.

Some features that can be added to the system are given below:

- battery temperature control,
- solar array orbit profile,
- charger a/t curve control,
- advanced load profile for transient analysis,
- orbit operation,

- energy optimization operation,

- development of application of network analyzer.

Figure 8.1 gives the basic flow chart for the possible final software function.

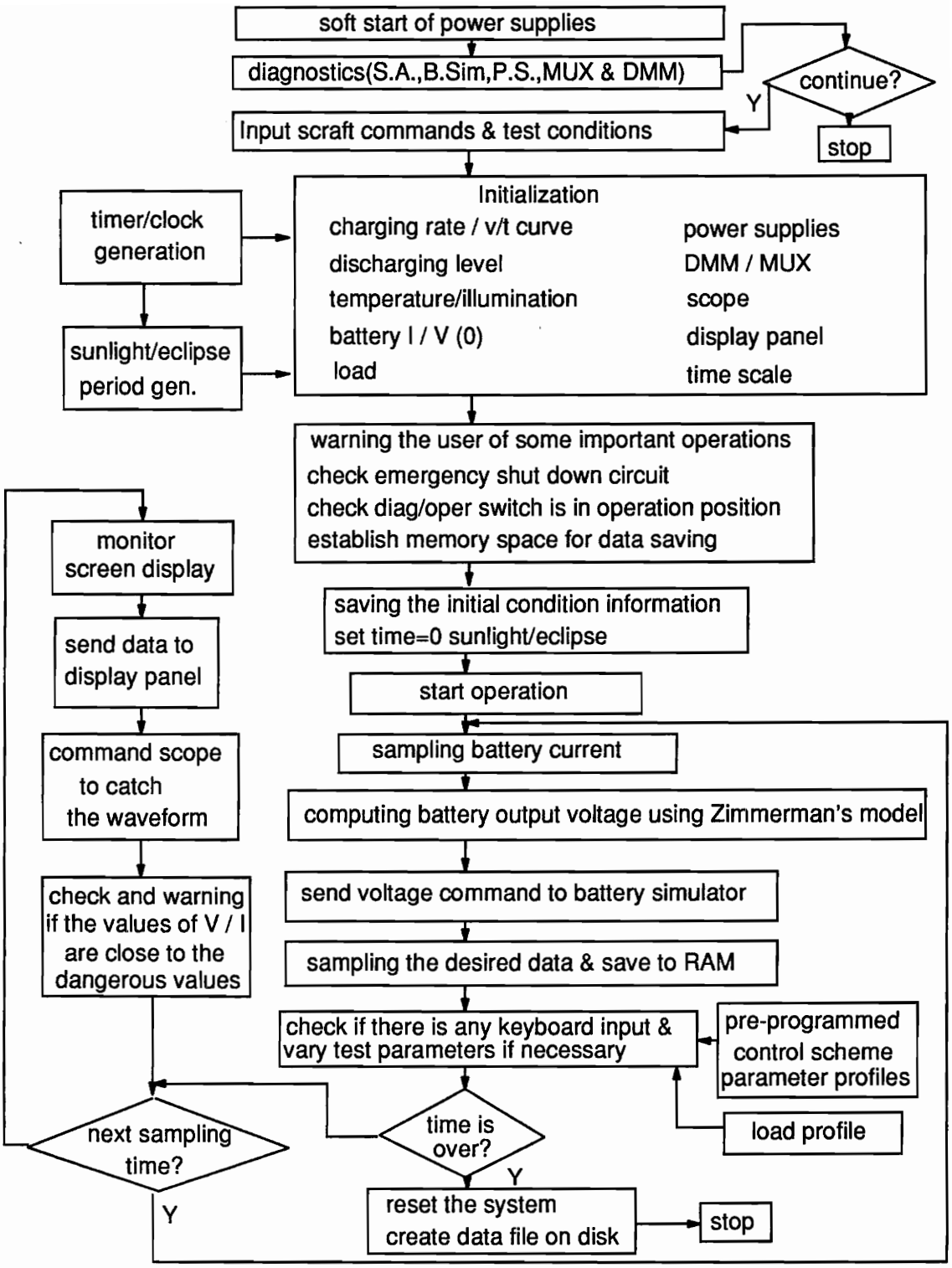


Figure 8.1 Flow Chart of Testbed Control

References

[1] D. Sable, "Design and Development of a Testbed for the Space Platform Power System".

[2] S. J. Butler, D. Sable, "Design of a Solar Array Simulator for the NASA EOS Testbed", IECEC 1992 proceedings.

[3] Z. Gur, X. Mang, "Design of a Nickel-Hydrogen Battery Simulator for the NASA EOS Testbed", IECEC 1992 Proceedings.

[4] A. R. Patil, D. Sable, "Design of a Solar Array Shunt Switching Unit for the NASA EOS Testbed", IECEC 1992 proceedings.

[5] H.G. Zimmerman, R.G. Peterson "An Electrochemical Cell Equivalent Circuit for Storage Battery Power System Calculations by Digital Computer," IECEC 1970 Proceedings, Vol. 6, pp. 6.33-6.39.

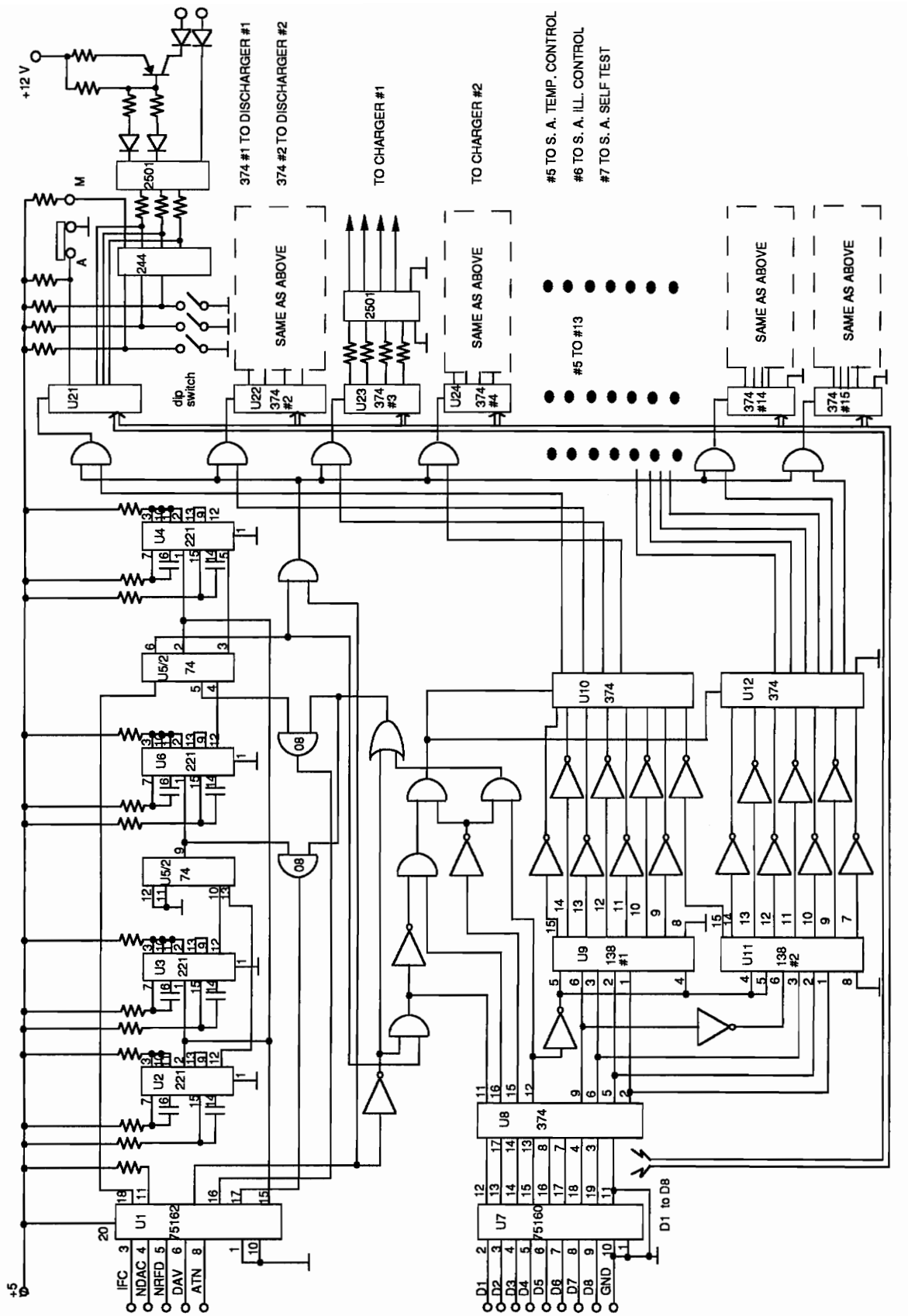
[6] D. Frate, "Nickel Hydrogen Cell Low-Earth-Orbit Life Test Update," IECEC 1991 Proceedings, Vol. 3, pp. 263-268.

[7] P. Leggett, "Olympus F1 Nickel Hydrogen Experience," IECEC 1991 Proceedings, Vol. 3, pp. 269-275.

[8] Standard Digital Interface for Programmable Instrumentations, ANSI/IEEE Std. 488.1-1987.

[9] Space Platform Power System Hardware Testbed, VPEC Quarterly Progress Review, 1991-1992.

Appendix Schematic of the Custom Interface Board


+12 V
2501
244
M
A
dip switch
U21
374 #1 TO DISCHARGER #1
374 #2 TO DISCHARGER #2
TO CHARGER #1
TO CHARGER #2
#5 TO S. A. TEMP. CONTROL
#6 TO S. A. ILL. CONTROL
#7 TO S. A. SELF TEST
SAME AS ABOVE
SAME AS ABOVE
SAME AS ABOVE
SAME AS ABOVE
#5 TO #13
U22 374 #2
U23 374 #3
U24 374 #4
374 #14
374 #15
U1 75162
U7 75160
U8 374
U9 138 #1
U10 374
U11 138 #2
U12 374
U2 221
U3 221
U4 221
U6 221
U5/2 74
08
IFC
NDAC
NRFD
DAV
ATN
D1
D2
D3
D4
D5
D6
D7
D8
GND
D1 to D8
+5

Vita

The author received the Bachelor's degree from the Harbin Institute of Technology, Harbin, main land of China in 1982. He studied there in control engineering and was employed by the same university as an assistant professor. Within the five years after his graduation, in addition to the enjoyment of his teaching, he did many research works on the design of control systems, which were used for industry and defense. He came to the U.S. as a visiting scholar at the end of 1986, and worked in the field of motion control at Virginia Tech. He started his Master program in 1990, and worked in the Space Lab of Virginia Power Electronics Center (VPEC) from 1991 to 1992.