n9

N91-31793

Development of the CELSS Emulator at NASA JSC

Hatice S. Cullingford
NASA Johnson Space Center

ABSTRACT

The Controlled Ecological Life Support System (CELSS) Emulator is under development at the NASA Johnson Space Center (JSC) with the purpose to investigate computer simulations of integrated CELSS operations involving humans, plants, and process machinery. This paper describes Version 1.0 of the CELSS Emulator that was initiated in 1988 on the JSC Multi Purpose Applications Console Test Bed as the simulation framework. The run module of the simulation system now contains a CELSS model called BLSS. The CELSS Emulator empowers us to generate model data sets, store libraries of results for further analysis, and also display plots of model variables as a function of time. The progress of the project is presented with sample test runs and simulation display pages.

THE LONG-DURATION FUTURE SPACE MISSIONS including settlements on lunar and planetary surfaces will require a spectrum of human life support systems that regenerate air, water, and food from process wastes. Such life support systems i.e. Controlled Ecological Life Support Systems (CELSS) will be expected to provide highly reliable service with integrated biological and nonbiological components of limited reliability.

The conceptual design of a CELSS to sustain a crew of 6 was reported by the author last year for a piloted Mars sprint (1,2). An integrated operation was achieved in this study with air, water, and waste processing and supplemental food production. A computer model of the specific life support system design was developed by utilizing the SINDA '85 code to support the analysis. The system performance was analyzed in terms of the physical-chemical subsystems or components and a greenhouse. Two crops, lettuce and winged beans, were chosen for bioregeneration. The crew cabin and the greenhouse were physically separated but dynamically interfaced with mass and energy flows. The plants provided 9, 29, 22, and 50 percent of air revitalization, water reclamation, wet food supply, and waste processing functions, respectively.

On the other hand, a computer model called BLSS was developed earlier by Drs. John Rummel and Tyler Volk to analyze a generic CELSS with a wheat-based crop cycle (3,4). BLSS uses a reservoir approach in modeling, i.e. subsystems are tracked not phenomenalogically but instead in their response to the reservoir inputs and outputs. High-productivity wheat is the sole driver for biogeneration functions including the crew diet.

Currently, the CELSS Emulator of the Solar System Exploration Division at JSC is being developed on a computer test bed system designed at JSC for the Space Station Freedom, called the Multi Purpose Applications Console, or MPAC Test Bed, with the incorporation of the BLSS model for first CELSS simulation experience. This paper is intended to describe the Emulator and its progress with results.

SIMULATION SYSTEM

The Multi Purpose Applications Console (MPAC) Test Bed was developed by the Systems Development and Simulations Division at JSC to provide an environment for the prototyping of the Space Station Freedom's display and control techniques. The development system consists of a MicroVAX II Computer employing the VAX/VMS Version 4.7 operating system, a Raster Technology Model One/85 graphics terminal, and a Dataviews graphics management system. The MPAC Test Bed provides the display and control of data values generated internally by simulation models or from data collected via one of the networks servicing the MicroVAX Computer. The system currently supports software developed in Fortran, C, Pascal, Ada, and Assembly languages.

The User Interface Language (UIL) Demonstration System (UDS) is the simulation system developed at JSC for the purpose of generating data for a UIL workstation . The software system requirements definition for all model measurements is controlled by a data base file . The realtime software provides a user interface for displays and controls in the Space Station Freedom command and control workstation . All realtime measurement values are found in a common memory designated as Current Value Memory (CVM) . All task communication is through common memory and event flag clusters , a service provided by the MicroVMS Operating System. Simulation models drive the values in CVM and provide the output to the Freedom simulation workstation . An offline system provides the user with the ability to add new measurements to the active data base, or to modify existing measurements. The offline system also enables the user to build the CVM structure based on the current data base. The philosophy in the software development is to generate as much information as possible for the realtime system in an offline mode. This offline mode is used in the production of the global common memory map, the generation of display pages, the identification of measurements for the simulation models, and the generation of parameter tables used to identify measurements being transported between models.

The CVM contains the measurement values for all items in the active data base. The entry for each measurement contains a control word for realtime processing as well as the current value of this measurement. The models can access the CVM which is primarily driven by the outputs of other models .

CELSS EMULATOR DESCRIPTION

The goal of the CELSS Emulator development is to have a smart tool to help define the parameters needed for a real CELSS for future space pioneers. The Emulator allows the user to easily set up data sets and manages the output files produced by model runs. Table 1 lists various steps leading to a CELSS design while Table 2 shows the role of the Emulator in this design process. Our approach has been such that the initial version of the Emulator functions in accordance with the existing BLSS model. Hence, the results discussed in this paper are identical to those of the original version of BLSS.

Figure 1 shows a diagram of the Emulator system. The top portion in the figure deals with the computer system configuration, operation, and offline processing. The middle portion contains the simulation computer system consisting of the BLSS model, a future crop model, the database, the display and control software package, and a future logic processor. The bottom portion of the diagram refers to the display and control station equipment i.e. the graphics terminal. The dotted lines indicate future implementation.

The Emulator operational interfaces are presented in Figure 2. Note that a Preprocessing Utility module provides the capability to develop, manipulate, and store data sets that can be inputs to the BLSS model within the CELSS Emulator. A Postprocessing Analysis module allows us to generate plot files with the data obtained in a simulation run. Central to all the operations is the BLSS Utility Directory (BUD) to organize the storage and retrieval of the setup files and the report files.

SAMPLE RUN

Figure 3 presents the CELSS Emulator operational flow. When the Emulator is running, the data change is observable at appropriate display pages. Any time duration can be chosen for the BLSS calculations. Figure 4 shows a display page for BLSS summary at the end of a 200-day run. There are six distinct panels on this page. The top panel indicates time in days and crew size. The second panel lists the current values for dry food storage and process mass. The BLSS model assumes no leakage, so the process mass remains constant. The next panel on the summary display page shows edible and inedible biomass being produced. The fourth panel indicates the amount of waste being processed.

Displayed right below it are the flow rates of air and water for the plant growth module and the crew compartment. Finally, the Emulator displays information on how much water, oxygen, carbon dioxide, and plant nutrient are available in storage.

Figure 5 shows another display page, the Crew Compartment Atmosphere Status. To change the crew size (see Figures 2 and 3), for example to 6, the number across "CREWSIZE" is replaced with "6" by means of either the slider or the keyboard.

By using the Postprocessing Analysis module, plots such as

Mission Definition	(Objectives, external interfaces, environment)
System Requirements	(Performance objectives, measures of system value)
System Concepts	(System elements and interfaces, element-level performance objectives, parametric evaluations, trade studies)
Feasibility Evaluation	(Alternative approaches to subsystems, comparison of performance with existing technology, formulation of prototypes)
System Definition	(System- and subsystem-level specifications, subsystem and system operation description)
System Design	(Technical drawings and configurations, performance parameters, operational procedures)

Table 1: Steps Leading to CELSS Design

Phase	Input	Output
Mission definition	Scenarios Environment	
System requirements	Performance measures	
System concepts	Element definitions Interface definitions	Performance objectives Parametric studies System performance
Feasibility evaluation		Alternative approaches Comparison of technologies Evaluation of prototypes
System definition		Compatibility evaluations Validation of interfaces
System design		Fidelity evaluation Crew training - later

Table 2: Role of CELSS Emulator in Design Process

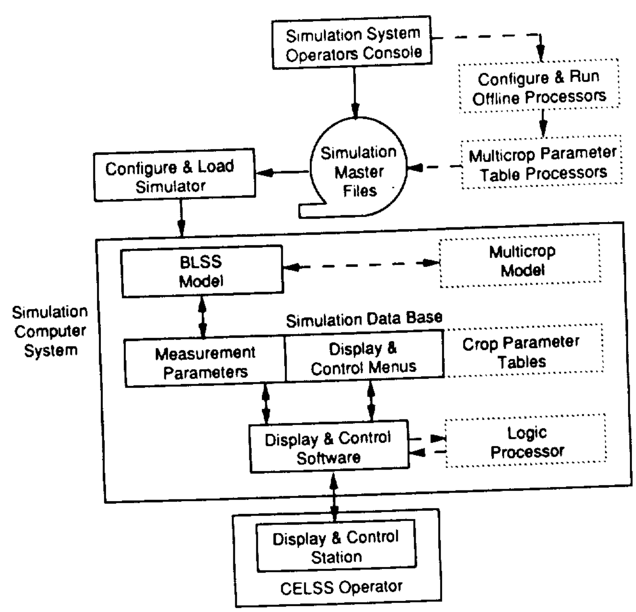

Figure 1 — CELSS Emulator System Diagram

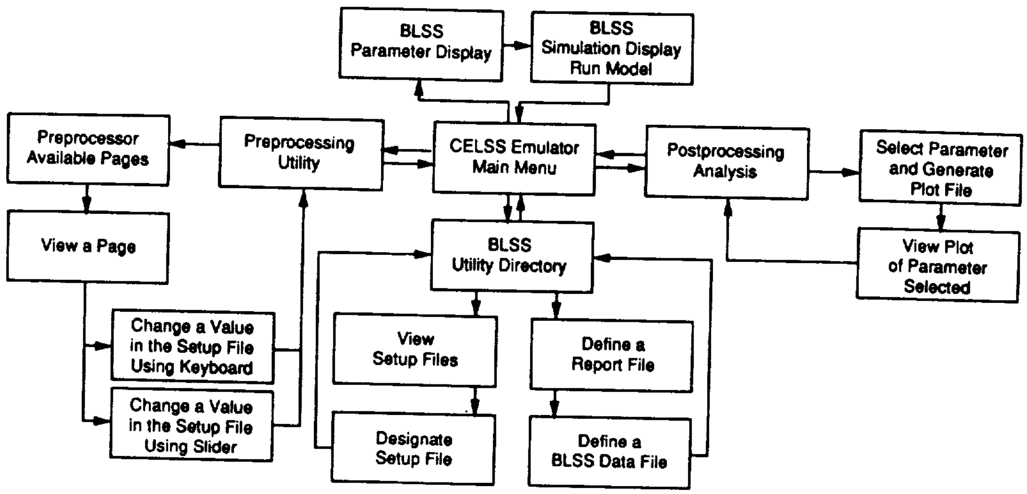

Figure 2 — CELSS Emulator Operational Interfaces

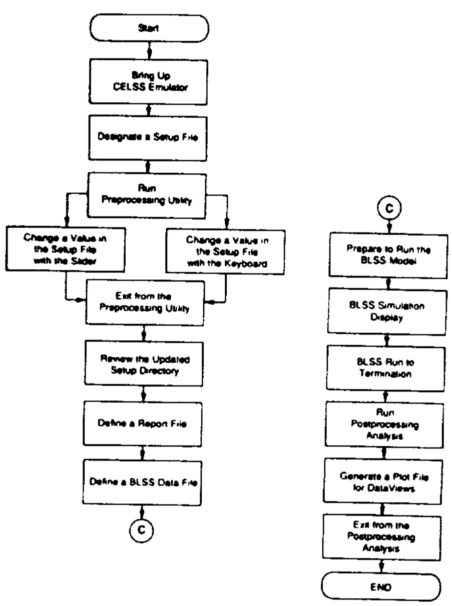

Figure 3 — CELSS Emulator Operational Flow Diagram

those given in Figures 6, 7, and 8 can be generated. Figure 6 shows a time plot for three parameters: dry food, total edible biomass, and total inedible biomass. Note the fifty-five day growth cycle for the wheat crop. The time history for the dry food is represented by the sawtoothed graph. The edible biomass is harvested and stored as dry food. While the next crop is being grown, the dry food in storage decreases as it is eaten by the crew.

Various variables can be compared for different crew sizes. Figure 7 show a plot for dry food storage with a range of crew size from 0 to 6. As calculated, food consumption increases with crew size. Also, at each harvest time, the dry food storage is filled to capacity. The excess harvest is either used as seed or processed as waste. In the first cycle, a small fraction of the dry food is used for seeding the second cycle. This amount is recovered in subsequent harvests.

Similarly, Figure 8 is a plot of the water storage with time for crew size ranging from 0 to 6.

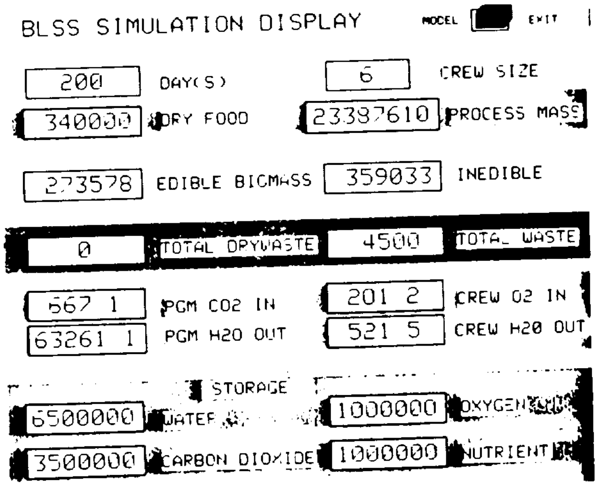

Figure 4 — BLSS Simulation Display Page

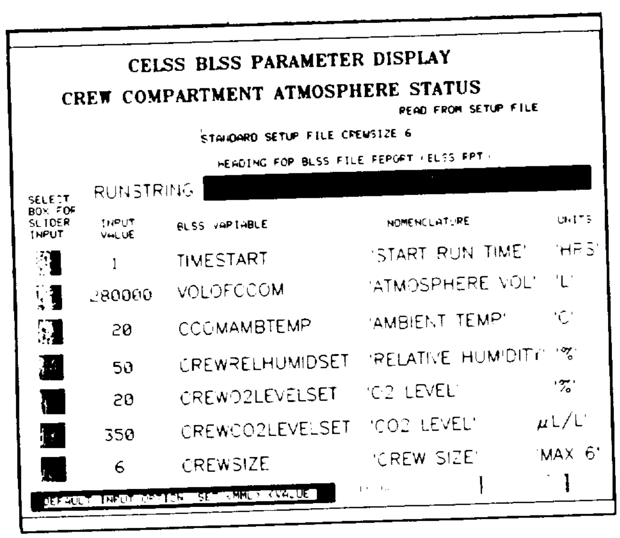

Figure 5 — Display Page for Crew Compartment Atmosphere Status

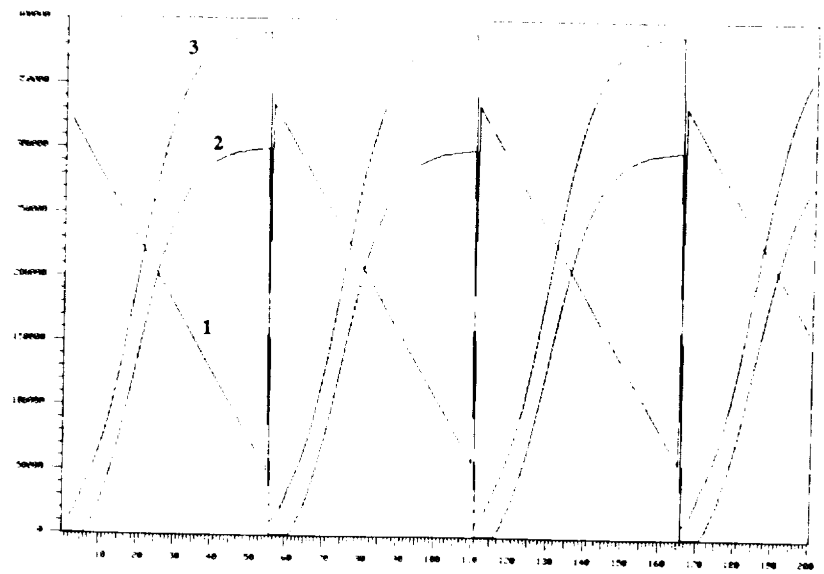

Figure 6 — Plots of Dry Food (1), Total Edible Biomass (2), and Total Inedible Biomass (3) for a BLSS Sample Run of 200 Days.

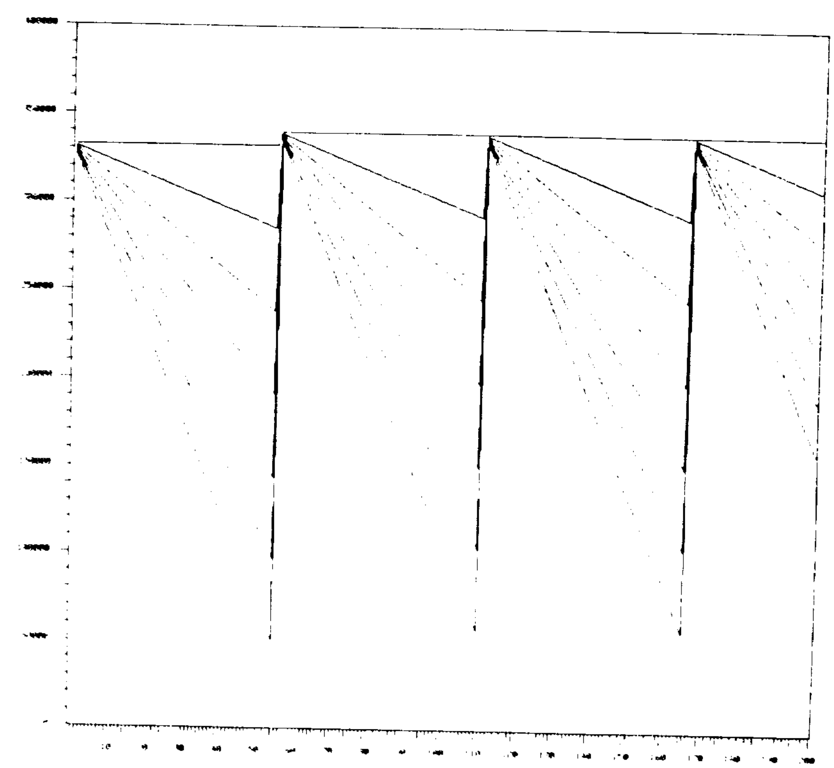

Figure 7 — Plots of Dry Food Storage for Crew Size Ranging from 0 (top) to 6 (bottom).

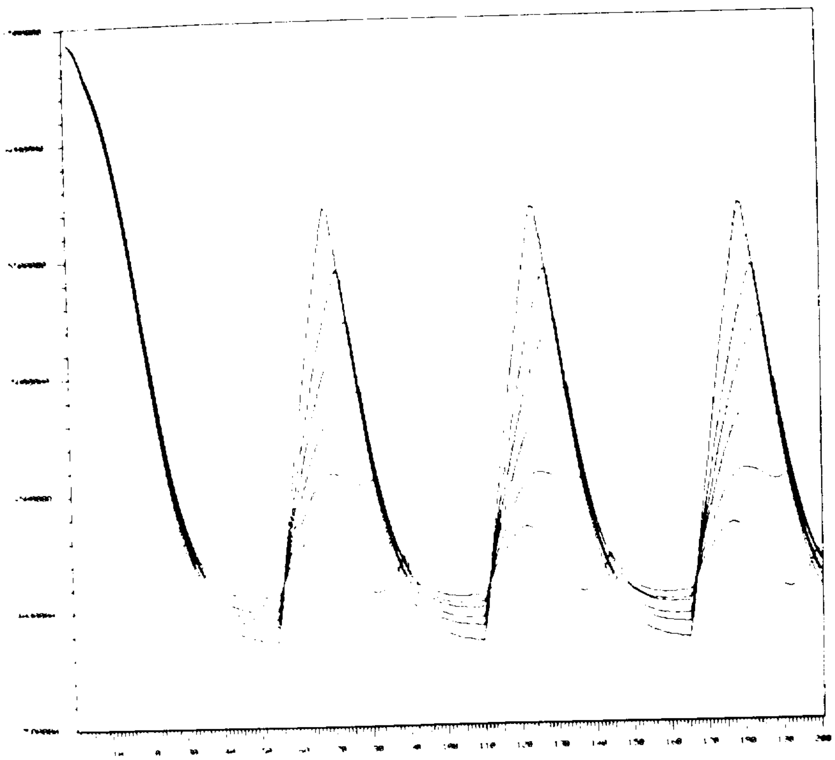

Figure 8 — Plots of Water Storage for Crew Size Ranging from 0 (top) to 6 (bottom).

CONCLUSIONS

The CELSS Emulator Version 1.0 represents the first significant step in the development of a computer-based simulation tool to study stability, monitoring, control, and integration aspects of operating CELSS. Through computer simulation, the CELSS Emulator at JSC will allow researchers to determine the characteristics of future CELSS equipment and its interaction with the crew. CELSS is a very complex system. We need the Emulator to unravel this complexity so we can understand where we are and where we are going.

ACKNOWLEDGEMENT

This work was performed under NASA JSC contract NAS 9-17900 with Lockheed Engineering and Sciences Company (LESC) in Houston, Texas. The author recognizes the technical contributions made by the LESC employees W. A. Holley, W. P. Bennett, P. S. Jones, J. G. Carnes, and D. Hervey and expresses her gratitude to James Duron at NASA JSC for training, support, and inspiration. Special thanks are due to John Rummel, Tyler Volk, Maurice Averner, and Michael Duke for their continuing encouragement and support. Without all the help too numerous to mention this project would not be possible now.

REFERENCES

1. H. S. Cullingford and M. Novara, "Conceptual Design of a Piloted Mars Sprint Life Support System," 18th Intersociety Conference on Environmental Systems, San Francisco, CA, July 11-13, 1989.

2. M. Novara and H. S. Cullingford, "Bio-Isolation of Plants and Humans in a Piloted Mars Sprint," ibid.

3. J. D. Rummel and T. Volk, "Modeling a Modular Controlled Ecological Life Support System," NASA Space Life Sciences Symposium, Washington, D.C., June 21- 26, 1987.

4. John D. Rummel and Tyler Volk, "A Modular BLSS Simulation Model," COSPAR paper 11.22, Advances in Space Research, 1987.