2005-01-2962

Multi-scale modeling of advanced life support systems

Eric-Jan Manders
Institute for Software Integrated Systems, Vanderbilt University

Scott Bell
S&K Technologies Inc. at NASA Johnson Space Center ER2

Gautam Biswas
Institute for Software Integrated Systems, Vanderbilt University

David Kortenkamp
Metrica Inc. at NASA Johnson Space Center ER2

ABSTRACT

Regenerative life support systems for long duration human space exploration missions present unique design challenges that are also reflected in constructing behavior models of these systems for analysis purposes. These systems have multiple modes of operation and complex non-linear dynamics that occur at multiple time scales. Coarse grained analysis of the complete system over long duration and fine grained analysis of smaller system elements while avoiding computational intractability can be achieved by using multiple modeling and simulation paradigms. We describe a multi-level simulation model of an advanced life support system. The simulation model couples a discrete-event approach at the system level, with more detailed hybrid (continuous/discrete) physical system modeling at the sub-system level.

INTRODUCTION

Advanced Life Support (ALS) systems that can support human life for extended duration in the hostile environment of space are expected to become a crucial component of the NASA manned space exploration program. For any manned mission, maintaining a habitable crew environment is a mission critical requirement. However, future missions that include lunar and Mars habitats, long range transit, or extended stays in orbital stations present a new set of complex design challenges. The ALS system will have unprecedented requirements for robustness, must be able to recycle and reuse resources, and its operation should not distract the crew from executing the primary mission and science related tasks.

An advanced regenerative ALS system comprises varied sub-systems that have mechanical, electrical, hydraulic, chemical, and biological components, and processes with complex interactions that involve multiple time scales. Requirements for autonomous operation also imply a need for advanced control techniques that include adaptive resource management, and the need for diagnostic capabilities and fault adaptive control in the system [1].

Model building and simulation of those models play an important part in modern system design and testing. For space applications, as missions become more complex and of longer duration, they will likely become important as well for mission planning and evaluating “what-if” scenarios. System behavior must be modeled across multiple time-scales and multiple physical domains, to capture the dynamics both as a set of interacting processes that transform the available resources, and as a system of connected sub-systems that can be hierarchically refined to their underlying physical components, such as electric motors, pumps, compressors, valves, and heat exchangers. The design of advanced control architectures also depends critically on having accurate models of the complete system, and corresponding advanced simulation methods that inform the design process and help in validating the controllers.

We have developed a multi-scale ALS simulation model that is built using two coupled simulation engines. A high-level model that represents an integrated ALS system with all major components is developed in BioSim, a discrete-event simulation engine designed specifically for ALS modeling [9,10]. A physics-based, mixed continuous/discrete (hybrid) component oriented model is developed for one of the principal ALS subsystems, the water recovery system (WRS). This simulation model is created with the Matlab/Simulink visual programming environment [12]. BioSim implements producer-consumer models for the ALS modules that operate on the available resources. The interface between the simulation engines is realized by representing resources that are shared with the WRS module in both simulation models, and using a time-triggered communication mechanism between the simulation engines for control. A supervisory controller for the WRS allows the higher level ALS system model to impose resource limitations on the WRS and affect its operational mode.

The paper first presents a brief overview of the BioSim architecture and the generic ALS model. We then develop the Matlab/Simulink model for the WRS. Next, we describe how the integrated simulation is constructed by using additional interface and communication components in both simulation engines, and then illustrate the simulation results of the WRS under the externally imposed resource constraints of the ALS system. We end the paper with conclusions and a discussion of future work.

DISTRIBUTED DISCRETE EVENT SIMULATION OF ALS SYSTEMS: BIOSIM

BioSim is an active research project at NASA JSC and Metrica, Inc. Its primary objective is to supports intelligent controls research for ALS systems [9,10].

ARCHITECTURE

Simulation Engine – The discrete event system has a single global simulation clock that advances the process time in fixed one hour time steps, each step is a simulation 'tick'. The principal modeling entities are the functional processes and the resource stores, and the model relations represent the exchange of resources. A simulation continues until a "mission" concludes, when a resource is depleted, or a system failure event occurs in one of the processes, e.g. all crew members have expired.

BioSim supports stochastic processes that can be used to generate random events in the system, e.g., the occurrence of a fault in a sub-system, or a crew member becoming sick. Furthermore, it allows for the modeling of inherent process noise in a sub-system. Processes are explicitly parameterized in the model equations that represent the resource consumption/production relations for a simulation object.

Interfaces – The architecture exposes sensor and actuator interfaces to the objects in the simulation. This allows for BioSim to interface with external controllers. The sensor models in the interface support sensor noise as well. A graphical user interface provides visualization for the functional processes as well as the resource levels in the system.

Distributed Framework – BioSim is implemented in a distributed framework using CORBA as the middleware layer. Consequently, components of the simulation, or the controller, can be physically distributed over a network. In addition, the only constraint on the suitability of a modeling tool, or programming language to implement a BioSim component, is that it supports, or can be extended with a CORBA interface. The core system is implemented in Java with extensive configuration capabilities to synthesize a specific ALS realization.

GENERIC ALS MODEL

Figure 1 illustrates a BioSim model of an ALS system based on ideas described in [5]. This generic model includes the major ALS components and their interactions. Functional blocks include air revitalization, crew activities, food processing, biomass production, water recovery, waste processing, and power generation. All system resources are represented explicitly through their storage facilities. Two environmental areas, that model the habitat, and crop growing area, have direct air-exchange relations with the crew and the biomass production function blocks. Both areas effectively operate as buffers for gas mixtures. Injector and accumulator modules model the transfer mechanism for gasses (O_2, N_2, and CO_2) to and from the environmental areas. External inputs to the model are a crew activity schedule and a crop growing schedule. Alternate ALS configurations that include mission specific elements such as extra vehicular activity, and additional functional modules, such as air humidification and de-humidification components, are described in [9].

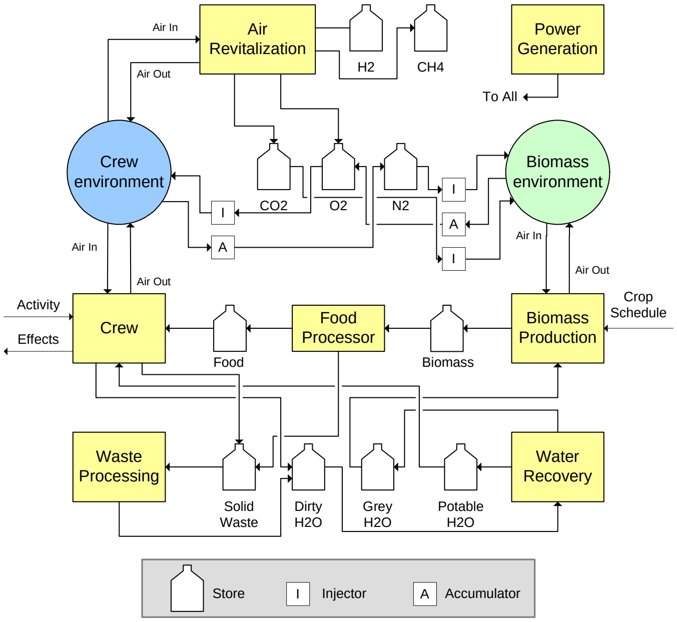


Figure 1: Diagram of a generic advanced life support system implemented as a BioSim model.

HYBRID MODELING AND SIMULATION OF THE WATER RECOVERY SYSTEM

An actual experimental WRS was designed, built, and evaluated at the NASA Johnson Space Center (JSC) in a multi year project [4,11]. The WRS simulation model used in this work, as well as the basic WRS model in BioSim, are based on the JSC design.

THE JSC WRS SYSTEM TESTBED

The JSC WRS test-bed comprises four major sub-systems, as illustrated in Figure 2. A Biological Water Processor (BWP) draws waste-water from a storage tank and removes organic as well as ammonia-based inorganic compounds. Effluent from the BWP is passed to the Reverse Osmosis (RO) System that removes inorganic salts and produces both clean and waste effluent streams. An Air Evaporation System (AES) further processes the waste of the RO, and a Post Processing System (PPS) merges the combined cleaned effluent of the RO (approx 85%) and AES (approx 15%) to achieve potable water quality level. The potable water is collected in a second storage tank. In this work we consider only waste water and potable water as resources [11]. An additional water resource may be modeled, to indicate “grey water”, and the BioSim WRS model represents the influent to the PPS as this resource.

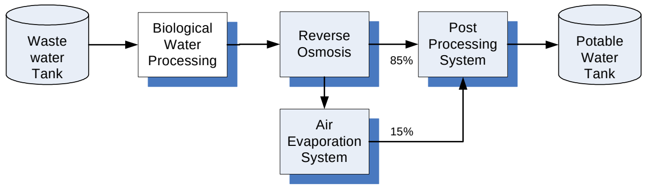


Figure 2: Block diagram of an Advanced Water Recovery System developed at NASA JSC.

MODELING THE WRS SUB-SYSTEMS: A COMPONENT ORIENTED APPROACH

Modeling the dynamics of a physical system can be done at various levels of abstraction. Modeling for diagnosis and control applications typically adopts a lumped parameter approach. In this work we will also take a component oriented modeling approach, which preserves the correspondence between a physical component in the system and its model representation. The WRS sub-systems have both continuous and discrete dynamics. We capture this by using a hybrid systems modeling approach. The continuous models are augmented with switching logic that is realized as finite state automata. The transitions between discrete modes may be controlled either by external signals from a controller, or by signals internally generated from evaluated transition conditions.

In the current work we use transfer function models to capture the continuous dynamics. In other work we have applied the hybrid bond graph (HBG) modeling paradigm to develop physical system component models [3]. HBG models capture continuous behavior in terms of domain independent energy storage, dissipation, transformation, and source/sink elements, and discrete changes between continuous modes are introduced using dedicated switching elements. From HBG models we can generate Matlab/Simulink simulation models using automatic translators [3]. A detailed description of the HBG model for the RO system is presented in [2,3] and a preliminary HBG model of the AES is described in [1]. A HBG model of the BWP is currently under development. The HBG models for these WRS sub-systems have more refined continuous dynamics but present the same interfaces. The simulation model for the WRS derived from the HBG will be a drop-in replacement for the hand created models used in this paper.

Parameter values for the sub-system models were estimated from the JSC system measurement data [11]. When technical details to model the component physics are not available, approximations are made. In particular, power consumption of components such as motors that drive pumps in this work is abstracted to pressure/flow products of the simulation outputs. Process noise is included in certain physical component models to improve correspondence between the simulation results and the test-bed data. However, sensor noise is not explicitly modeled in the current version of the simulations.

Biological Waste Processor – Figure 3 shows a schematic of the BWP. A feed pump pulls water from the waste-water tank, and injects this flow into the recycling loop. The main components in the loop are a recycling pump that determines the input pressure into the Organic Carbon Oxidation Reactor (OCOR). The output of the reactor passes through a second reactor, the nitrifier. The last component in the loop is a Gas Liquid Separator (GLS) where the flow is split into a small outflow to the RO system, and a recycle flow that feeds most of the water back into the processing loop.

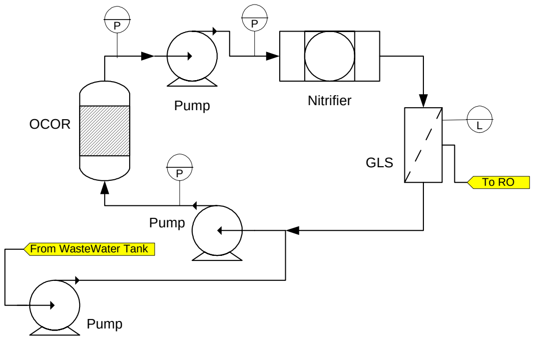


Figure 3: Diagram of the BWP.

Figure 4 shows the Matlab/Simulink model that corresponds to this diagram. Each pump and reactor has a corresponding model fragment that describes the system dynamics as a transfer function and discrete switching logic. Merging of the feed-pump flow with the recycled flow from the GLS is represented by adding the flow variables. Measurements of BWP variables are exposed in the model interface.

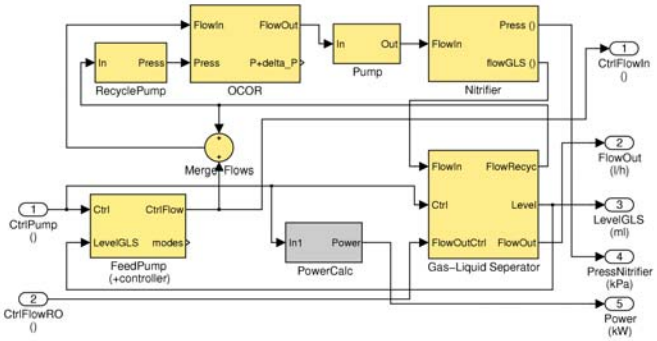


Figure 4: Matlab/Simulink model for the BWP.

Reverse Osmosis System – Figure 5 shows a schematic diagram of the RO system. The system has three operating modes that correspond to the setting of the multi-position valve. In the primary mode (setting 1) a feed pump transfers water from the BWP output reservoir into the system, where it is combined with the contents of the feedback loop in a coiled section of pipe that acts like a reservoir. A pump cycles the water through the loop, increasing the concentration of the brine, and producing. As the brine concentration increases and the membrane becomes dirty, the system efficiency decreases. The system switches to a secondary mode (setting 2) when the efficiency has dropped significantly. In this mode, to maintain a sufficient outflow, the recirculation loop is shortened and the recirculation pump speed increased. A third mode (setting 3), purges the now highly concentrated brine from the loop and flushes it to the AES. The RO cycle then restarts in primary mode. An additional slough mode, not modeled at this time, periodically follows the purge mode to clean the membrane from accumulated particulate material.

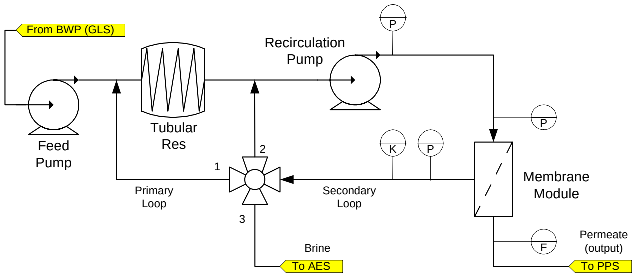


Figure 5: Diagram of the RO system.

The continuous behavior of the RO sub-system involves three interacting physical domains: hydraulic (pumps, valves, and pipes), chemical (conductivity of the brine, a measure of brine concentration), and electrical (pump motors). Power consumption in the system is determined by the two electric motors that drive the pumps. The simulation model used in this work is similar to the model presented in [3], except that the component models are represented as transfer functions, rather than as HBG fragments.

Air Evaporation System – Figure 6 shows a schematic of the AES. The AES operates as a cycle of multiple heat exchange processes. A reservoir collects the brine from the RO. The brine is absorbed by a wick and evaporated using hot air blown over the wick. The evaporated water is condensed by passing it through a heat exchanger, and collected in a tank before it is sent to the PPS system. The primary power consumers in this subsystem are the blower, which moves the air through the system, and the heating unit, which heats the air to facilitate evaporation of water. The Matlab/Simulink model of the AES is identical to that presented in [1], where the component models for the condenser, blower, heater and wick assembly are represented as transfer functions. Power consumption is computed from the heating level and blower speed.

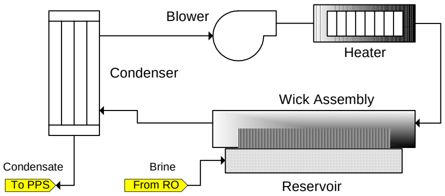


Figure 6: Diagram of the AES.

Post Processing System – The PPS system applies a multi-step treatment to improve the water quality by removing trace contaminants and bring it to potable levels. Modeling of the PPS is preliminary at present because we do not yet explicitly capture water quality aspects of the water in the simulation (the one exception is the use of conductivity in the RO system). The system dynamics of the PPS in the hydraulic domain are relatively simple because the PPS does not have an explicit internal storage capacity. The current model represents the merging of the influent flows from the RO and AES. A simple gain relation with outflow models the power consumption.

SIMULINK MODEL OF THE INTEGRATED WRS

Figure 7 shows the integrated simulation model for the WRS system dynamics. The control interface for the system is determined by the flow from the wastewater tank to the BWP, the BWP recycle-pump speed, the flow to the RO, the pump speed in the RO, The RO mode, flow to the AES, and AES on/off switch. The model has multiple sensors for each sub-system, but only the WRS outflow and combined power utilization of all sub-systems are exposed at the interface for feedback to the controller.

The integrated WRS simulation model can be scaled to realize a specific system sizing defined by a BioSim ALS configuration. However, a large scaling factor will allow the simulation model to scale in a manner that may not be feasible in the actual physical system.

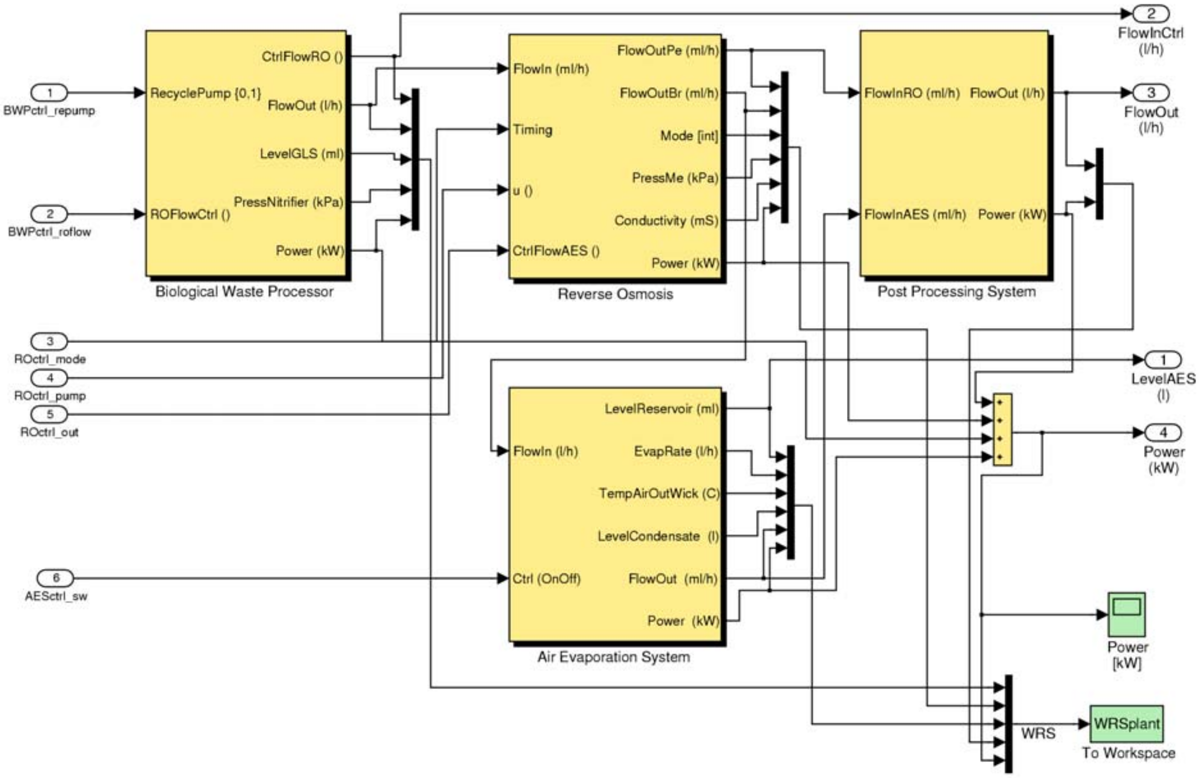


Figure 7: Matlab/Simulink model of the WRS. Many more measurements are available on the system dynamics than are exposed to the model interface.

INTEGRATING BIOSIM AND MATLAB/SIMULINK SIMULATION PROCESSES

Coupling the WRS model to the complete ALS model requires a WRS controller that takes the resource constraints imposed by the ALS system as inputs. The architecture that couples the two simulation engines must also ensure that the controlled WRS simulation correctly interacts with BioSim. This is achieved by mapping the shared resources that exist in BioSim to the Matlab simulation model.

WRS CONTROL IN THE INTEGRATED ALS SIMULATION

The resources considered are the waste-water available, the potable water required, and the power consumption allowed. The controller is designed so that it will run either the BWP and RO systems, or the AES during a time step, an actual requirement imposed to conserve power. The PPS is considered to on when any of the other systems is on. The AES module is only run when the input reservoir level exceeds a threshold, and then converts the contents of the reservoir to clean condensate as a single batch. The controller uses the AES reservoir level as a feedback signal. Effectively, this controller is a scheduler that sends either effluent from the cascaded BWP/RO sub-systems, or condensate from the AES sub-system through the PPS to the potable water tank. The controller takes into account that a constraint on the waste water availability does not affect the system when the AES is running, since it operates from its own reservoir.

The cycle time of the WRS model, without being affected by any resource limitations, is thus determined by the combined throughput of the RO and AES systems. The RO system has a four hour cycle, and it takes multiple RO cycles to generate enough brine to trigger the AES. The AES model is configured so that it can process the contents of a full reservoir in one hour.

The controller used in the current work can respond to resource limitations only by turning sub-systems on and off. This simple on/off control is not desirable in a real system, and in other work, the WRS sub-system models have been modified to allow multiple settings of pump speeds or mode switching in the RO to demonstrate the capabilities of a model-predictive controller [1]. These modifications allow more advanced resource management and will be integrated in future revisions of the integrated model as well.

DISTRIBUTED ARCHITECTURE FOR THE COUPLED SIMULATION

BioSim updates its resource store values once every simulation tick. During a tick the resource value is effectively constant. However, the Matlab simulation will update the resource values according to the simulation time step of the Matlab solver, which is much shorter than the one hour interval. We address this by creating a virtual resource store in the Matlab simulation model, and communicating the consumed and produced values with BioSim at the onset of each BioSim tick. In effect, the resources consumed by the WRS simulation are 'removed' from their stores at the BioSim tick start, and the resources produced by the WRS are added to the BioSim stores in a single operation at the end of the simulation tick. The change in BioSim resource levels is therefore discrete, as with the higher level simulation models for ALS models in BioSim. While the continuous simulation could update the BioSim stores at a higher rate, the BioSim architecture cannot currently process such updates. Tighter interaction between the simulation engines will require a higher resolution simulation tick in the BioSim engine.

BioSim models a power consumption constraint on each of the ALS sub-systems, and this constraint applies to the entire tick interval. We interpret this as a constraint on the average power consumption during the interval, and therefore, it is effectively a constraint on energy usage for the sub-system over the tick interval. To map this constraint to a WRS control signal, we compute the average power used by the WRS, and the system remains on as long as the average power does not exceed the power quota assigned.

Figure 8 shows an architecture diagram of the integrated simulation. On the BioSim side, the sub-system model of the WRS is replaced by a proxy process that contains a communication interface to the Matlab process. Additional components on the Matlab side implement the mapping of the BioSim resources to virtual resources in the WRS model. The Matlab model operates on a continuous clock, and the interaction between the simulation engines is determined by the BioSim clock.

Inter-process communication (IPC) between the BioSim and Matlab/Simulink processes is realized through a platform independent implementation of network sockets, and the combined simulation will run on any platform that supports both BioSim (Java), and Matlab. BioSim drives the integrated simulation, with a time-triggered message exchange. The communication mechanism uses blocking receive messages which means that the Matlab process will suspend until new control information is generated from BioSim. This implicitly assumes that we are simulating a controlled physical system, rather than a simulated physical reality under external, real-time control. The latter would require a more complex IPC strategy that takes into account the progression of physical system, time during the computational step of the controller [6]. In a high level controller that may involve deliberation and/or planning, this time may not be negligible, especially under fault conditions or when unexpected events require rapid response times.

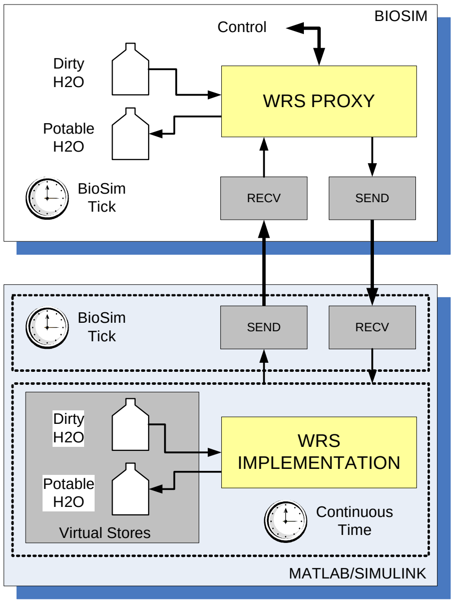


Figure 8: BioSim coupled with the Matlab/Simulink process.

Figure 9 shows the Matlab/Simulink model that implements the architecture of Figure 8. The BioSim tick is generated from the simulation clock and used to trigger the receive and send blocks. The virtual resource stores are both instances of a general fluid tank model with threshold detectors for empty and full. Each tank is initialized by the BioSim inputs at each tick. The waste-water tank triggers a control action when empty, and the potable water tank triggers a control action when the high threshold is reached. Additional feedback signals into the controller are the AES reservoir level, and the average power, which is computed outside of the physical system model. The outputs sent to BioSim at the end of the simulation tick are the levels of the virtual stores, and the average power consumption over the tick interval.

SIMULATION OUTPUTS

Figure 10 shows the WRS data from a multi-day simulation with external resource constraints imposed by the ALS. We plot several measurements for each sub-system, as well as the total WRS power consumption. From the data we observe the general behavior of the WRS cycle that completes when the AES has processed its input reservoir. A complete nominal cycle takes 36 hours. In the second cycle, we introduce a resource constraint and power consumption is reduced over a 6 hour interval (t=40-46(h)). The WRS system shuts down for a part of each BioSim tick and the cycle time increases (it takes longer to fill the AES reservoir). During the third cycle, we introduce a different constraint, during a 3 hour interval (t=90-93(h)) where the waste-water supply is constrained. The effect on the behavior is similar in that it limits the operation of the BWP and RO system, and the cycle length increases.

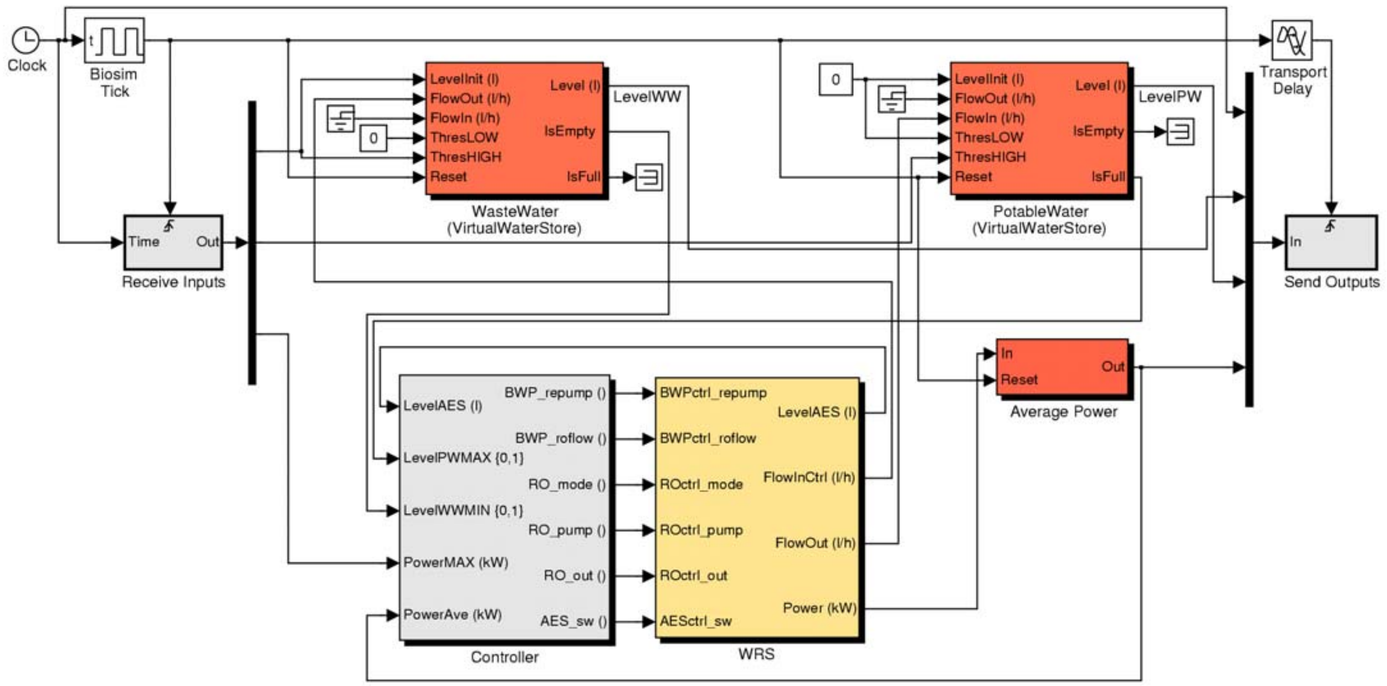


Figure 9: Matlab/Simulink model of the WRS system and BioSim interface components, with time-triggered communication, virtual resource stores, and average power consumption calculation and decision logic.

Note that the power consumption of the simulated WRS exhibits a piecewise continuous behavior since this controller does not continuously adjust the inputs to individual pumps and blower in the system.

CONCLUSIONS AND FURTHER WORK

The multi-scale model and simulation tool described in this work illustrates the potential of these tools for systems that are studied and analyzed at multiple time scales and at multiple levels of behavioral abstraction. The discrete event simulation of BioSim supports experiments to evaluate the coarse scale resource utilization, whereas the hybrid physical system models in Matlab/Simulink allow a detailed study of the continuous/discrete dynamics of a complex sub-system. We expect that the integrated simulation models will contribute to more sophisticated approach to designing and sizing ALS systems for a variety of manned missions.

Ongoing and future work focuses on two principal areas. Physical system models of ALS sub-systems will use the HBG paradigm where possible. The direct correspondence of HBG model parameters to physical system components have been exploited in our work on diagnosis and fault adaptive control for WRS sub-systems, and we will apply this to the integrated WRS as well. When HBG modeling is not an appropriate abstraction, higher levels of mathematical abstraction will be applied. The addition of sensor and actuator models that include noise and fault/degradation models of sensor and actuation hardware over the duration of the simulation interval will further enhance the utility of the simulation model. Finally, the real JSC test bed contains limit-based safety control schemes that can be modeled in conjunction with more sophisticated controllers.

On the interacting simulation framework further research will focus on generalizing the resource interfaces to be applicable to other ALS sub-systems. In other work we have applied a model-integrated computing framework to modeling complex systems and controllers for applications in diagnosis and fault adaptive control [7,8]. We will apply these techniques across all levels of detail in the ALS system and allow the modeler to capture multiple aspects of the model in a systematic manner. A longer term goal is to synthesize a comprehensive ALS simulation from those models through automatic translators.

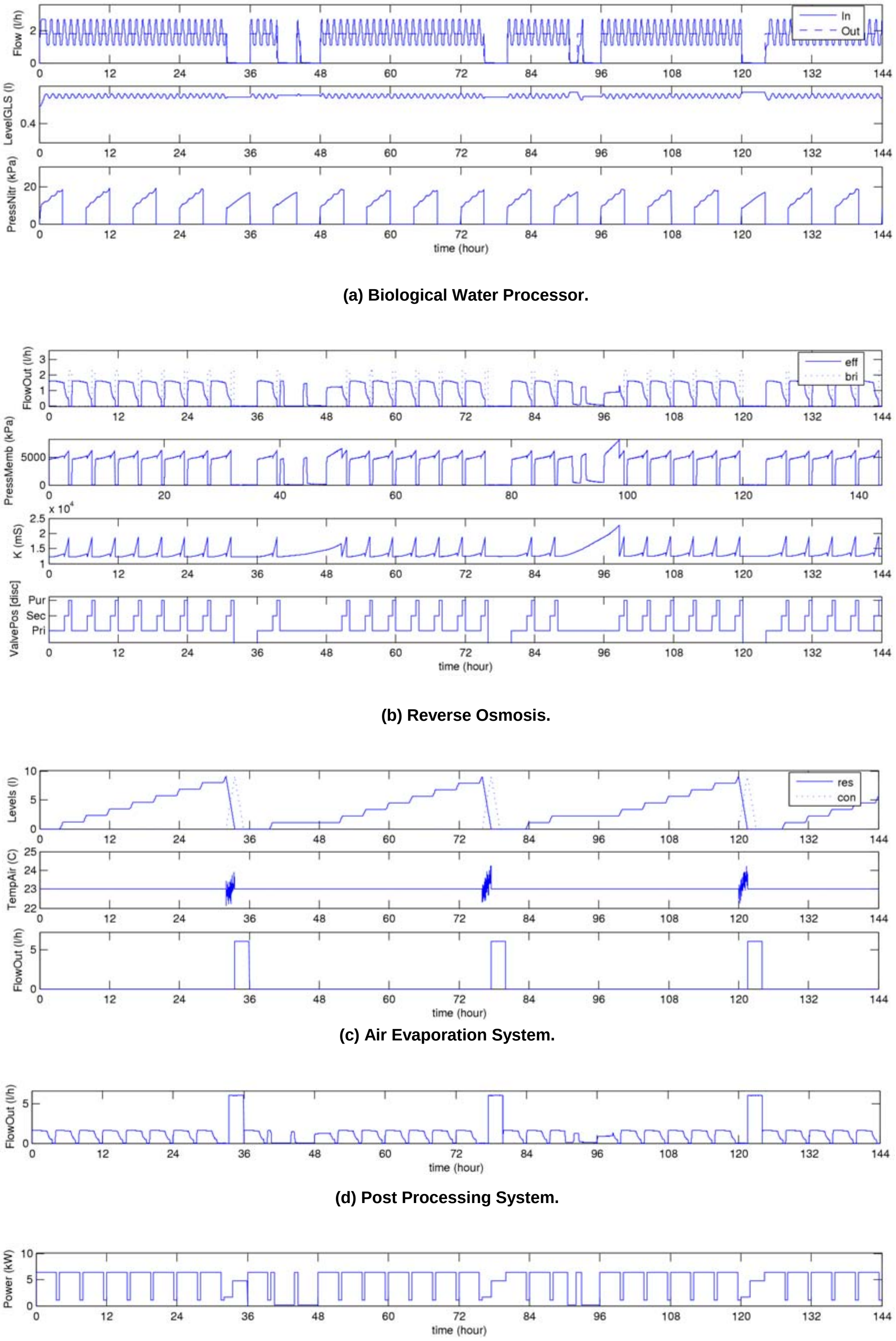
ACKNOWLEDGMENTS

This work was supported in part through grants from the NASA-ALS program (Contract number: NCC 9-159). We gratefully acknowledge the help provided by the members of the ALS group at NASA JSC. At Vanderbilt, we acknowledge Jian Wu and Jyoti Gandhe for modeling sub-systems of the WRS, and Nagabhusan Mahadevan for inter-process communication code.

REFERENCES

1. S. Abdelwahed, J. Wu, G. Biswas, J. Ramirez, and E.-J. Manders, “On-line hierarchical fault adaptive control for advanced life support systems,” in *Proc.*

32nd Intl. Conference on Environmental Systems, Denver, CO, 2004.



(a) Biological Water Processor.

(b) Reverse Osmosis.

(c) Air Evaporation System.

(d) Post Processing System.

(e) WRS Power Consumption.

Figure 10: Simulation results of a controlled WRS under generic ALS resource constraints for available power, waste-water supply, and potable water demand.

2. S. Abdelwahed, J. Wu, G. Biswas, J. Ramirez and E.-J. Manders, “Online Fault Adaptive Control for Efficient Resource Management in Advanced Life Support Systems”, Habitation – Intl. Journal for Human Support Research, 10(2):105-115, 2005.
3. G. Biswas, E.-J. Manders, J. Ramirez, N. Mahadevan, and S. Abdelwahed, “Online Model-based Diagnosis to support autonomous operation of an Advanced Life Support System,” Habitation - Intl. Journal for Human Support Research. 10(1):21-38, 2004.
4. P. Bonasso, D. Kortenkamp, and C. Thronesbery, “Intelligent control of a water recovery system: Three years in the trenches,” AI Magazine, pages 19–43, 2003.
5. B. E. Duffield, A. J. Hanford, and K. E. Lange, “Advanced life support requirements document,” Tech. Rep. JSC-38571, Rev. B, NASA-Lyndon B. Johnson Space Center, Houston, TX, Sept. 2002.
6. J. Hooman, N. Mulyar, and L. Posta, “Coupling Simulink and UML Models”, in *Proc. Symposium FORMS/FORMATS 2004*, pages 304-311, 2004.
7. G. Karsai, G. Biswas, S. Abdelwahed, T. Pasternak, S. Narasimhan, G. Peceli, G. Simon, and T. Kovacshazy. Towards fault-adaptive control of complex dynamical systems. In Tariq Samad and Gary Balas, editors, Software-Enabled Control - Information Technology for Dynamical Systems, chapter 17, pages 347-368. Wiley Interscience, 2003.
8. G. Karsai, J. Sztipanovits, A. Ledeczi, and T. Bapty, “Model-integrated development of embedded software,” *Proceedings of the IEEE*, 91(1):145–164, 2003.
9. D. Kortenkamp, S. Bell and L. Rodriguez, “Simulating Lunar Habitats and Activities to Derive System Requirements”, in *Proc. 1st AIAA Space Exploration Conf.*, Jan. 2005.
10. D. Kortenkamp and S. Bell, Simulating advanced life support systems for integrated controls research, in *Proc. 33rd Intl. Conference on Environmental Systems*, July, 2003.
11. K. D. Pickering, K. R. Wines, G. M. Pariani, L. A. Franks, J. Yeh, B. W. Finger, M. L. Campbell, C. E. Verostko, C. Carrier, J. C. Gandhi, and L. M. Vega, “Early results of an integrated water recovery system test,” in *Proc. 29th Intl. Conf on Environmental Systems*, 2001.
12. Matlab/Simulink, The MathWorks, Inc., MA, USA. http://www.mathworks.com.

CONTACT

The authors may be contacted through email at eric.j.manders@vanderbilt.edu or scott@traclabs.com. BioSim is available at http://www.traclabs.com/biosim

ACRONYMS, ABBREVIATIONS

ALS: Advanced Life Support

WRS: Water Recovery System

RO: Reverse Osmosis

BWP: Biological Water Processor

OCOR: Organic Carbon Oxidation reactor

GLS: Gas Liquid Separator

AES: Air Evaporation System

PPS: Post Processing System

HBG: Hybrid Bond Graph