Autonomous Control of an In-Situ Propellant Production Plant

Daniel Clancy, William Larson, Charles Pecheur, Peter Engrand , Charles Goodrich

Introduction

Utilization of extraterrestrial resources, or In-Situ Resource Utilization (ISRU), is viewed as an enabling technology for the exploration and commercial development of our solar system. A key subset of ISRU is In-Situ Propellant Production (ISPP), which involves the partially autonomous production of propellants for planetary ascent or Earth return. NASA has scheduled pilot ISPP demonstrations on Mars starting with the 2001 Mars Surveyor Lander, with human Mars mission scenarios as early as 2011. Such automated manufacturing facilities could also be applied to terrestrial space-port systems in the automation of launch vehicle and payload test, checkout and launch operations. Automation would allow for the more efficient use of personnel resources and enhance the safety of safety-critical operations where human intervention would be to slow or undesirable (e.g. situations where system safing would require placing personnel into a hazardous situation).

Operating an In-Situ Propellant Production (ISPP) plant on Mars poses significant challenges. One such challenge is the ability to maintain continuous plant operation without a Mars-based human presence, despite component failures and operational degradation. An Autonomous Controller (AC) can be used to monitor an ISPP plant for anomalous conditions, can diagnose component failures, and can provide recovery recommendations. The Livingstone system, developed at NASA Ames, is a model-based health management and control system that tracks the state of a device, detects and diagnoses anomalies and suggests alternative recovery actions. Livingstone is used at NASA Kennedy Space Center to develop an Autonomous Controller for ISPP. In turn, test and validation of the AC design is of critical importance and the employment of automated techniques of software verification and validation (V&V) to augment traditional scenario-based testing is desirable. This paper is in part based upon a more detailed discussion of ISPP and information technology by Gross et al. [2].
The ISPP Test bed

The ISPP Test Bed

Reverse-Water Gas Shift

To support the development and evaluation of ISPP technology, KSC is currently developing a hardware test bed. For the hardware demonstration a test bed using a Reverse Water Gas Shift (RWGS) to generate CO and O_2 from the CO_2 found in the Martian atmosphere. The RWGS reactor converts CO_2 and H_2 into CO and H_2O at a 10% efficiency rate. Thus, the outflow stream from the reactor contains liquid water, and gaseous CO, CO_2 and H_2. After exiting the reactor, a condenser is used to separate the water form the gases and then an electrolysis unit is used to separate the hydrogen from the oxygen. The oxygen is then stored while the hydrogen is fed back into the RWGS

reactor. Similarly, the CO is extracted from the gas mixture and the remaining CO_2 and H_2 are routed back into the RWGS reactor.
Control of the RWGS reactor is straightforward since the system has a limited number of components. A full-scale ISPP device, however, would require various other components along with redundant valves and flow controllers. As the number of components within the system increases, the probability of a failure increases and the discrete control problem becomes more complicated. The RWGS test bed, however, allows the demonstration of how these techniques can be used to control a real physical device for an extended period of time. Figure 1 represents a schematic of the proposed KSC RWGS type ISPP. For more information on ISPP technologies see [4].

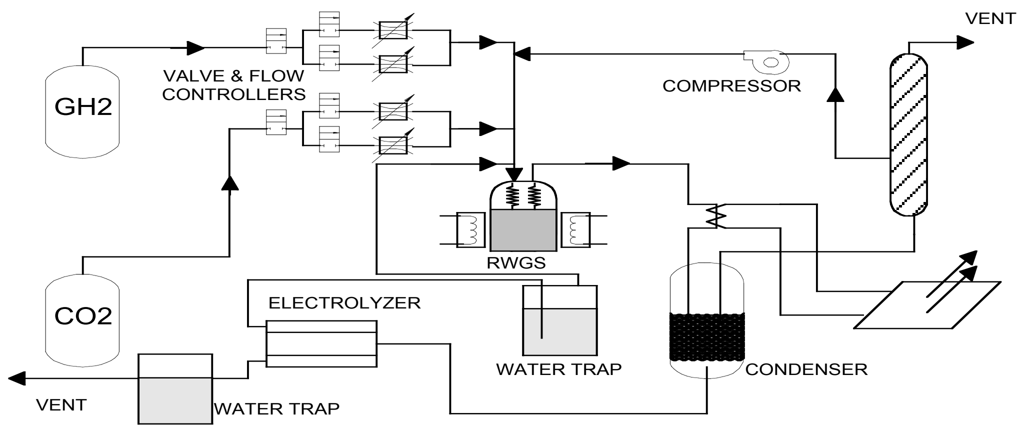

Figure 1. RWGS ISPP Test Bed

ISPP Plant control

ISPP Control Architecture

The current system architecture being developed to control an ISPP plant combines the Livingstone health management system with real-time executive for commanding the device. At the lowest level, embedded analog controllers are used to perform low-level regulatory functions. A real time executive performs nominal commanding of the ISPP. The real-valued sensor data is processed by a set of monitors that abstract the real-valued information from each sensor into a set of a-priori defined discrete values such as *high*, *medium* and *low*.

The Livingstone AC is based on a mathematical model of the ISPP plant. Livingstone uses the model like a financial analyst uses a spreadsheet - to analyze hypothetical operating conditions and failures. As the plant is operating, the Mode Identification component of the Livingstone System monitors the commands and sensors to identify the expected state of the plant. When a failure is identified, Livingstone notifies the real-time executive. For failures that require a fast response time, the real-time executive might respond reactively in a predefined manner. For other failures, the executive requests a sequence of reconfiguration commands from the Mode Reconfiguration component of

Livingstone and then continues commanding the device. Figure 1 shows a block diagram of the model-based control of an ISPP plant.

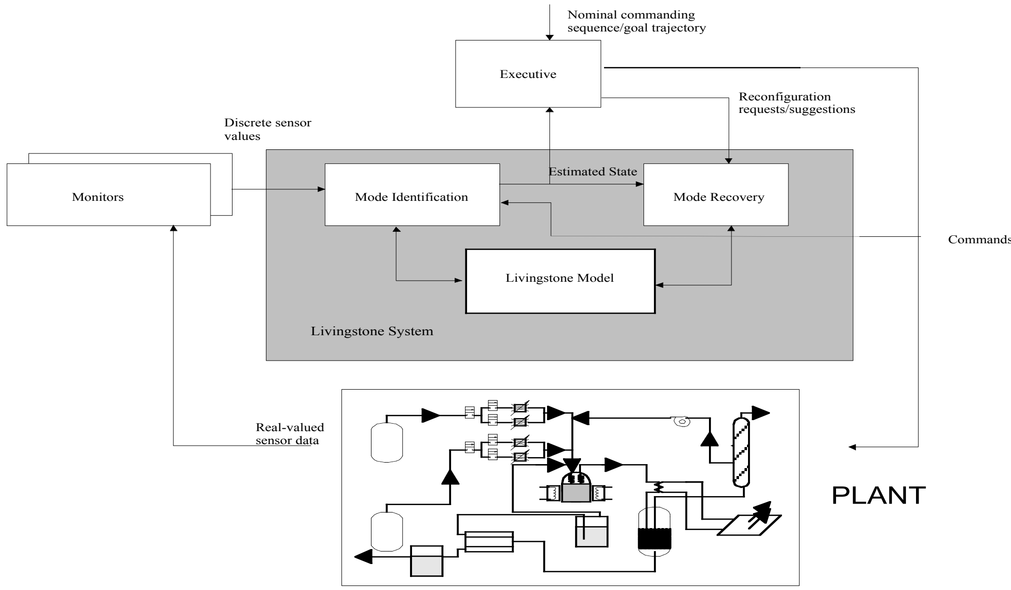

Figure 2. ISPP Control Architecture

Livingstone Models

The model describes the structure of the plant and the normal and abnormal functioning modes of its components. In addition to the description of the behavior of the device for each mode, the model also includes transitions between modes with guard conditions describing when the transition occurs. A cost is associated to nominal transitions, and a probability is associated to failures. One of the key benefits of this modeling paradigm is that the modeler is only responsible for describing the local behavior of each component and the relationships that exist between components. Livingstone then uses this specification to compose a larger, system model that can be used to reason about the global behavior of the entire system given the mode of each component. Furthermore, since the models are qualitative in nature, it is often straightforward to develop these models even before the hardware design is complete.

The Livingstone Reasoning Engine

Given a Livingstone model as described in the preceding section, Livingstone performs two main tasks: 1) state identification based on current (limited) sensor measurements; and 2) producing an optimal set of commands for system reconfiguration following a failure or external perturbation that transitions the system out of the desired state.

To estimate the current state of the system, Livingstone monitors the sequence of discrete commands that are issued to the ISPP plant to track the expected state of the device and compare the predictions generated from its model against the observations received from the sensors. Once a discrepancy occurs, Livingstone performs diagnosis by searching for the most likely set of component mode assignments that are consistent with the observations. The search technique used by Livingstone is able to efficiently search an exponentially large set of failure modes by focusing on the components whose state results in a conflict between the observations and the predictions. Once the state of the system is identified, the same search technique can then be used when reasoning about reconfiguration commands to identify the lowest cost set of commands that can be issued to transition the system into a state that satisfies the current operational goals provided by a higher level executive.

Benefits of Livingstone

The Livingstone-based ISPP AC can react to most component failures without human intervention, thus requiring fewer people to monitor ISPP operation and reducing the cost associated with large Earth-based support teams. In addition, earth-based fault detection and recovery can be significantly hindered by the communications delays associated with blackout periods and the distance between Earth and Mars. Operational time is a precious commodity when Mars missions can only occur every two years and sufficient fuel must be produced for a crew return vehicle prior to their arrival. Another challenge in operating an ISPP plant on Mars is the ability to thoroughly predict how the Martian environment will behave during the period of time in which the plant will operate. An Autonomous Control System can adapt to the environment in which it will operate. For more information on Livingstone see [3].

Verification of the ISPP AC

Model-based autonomy software such as Livingstone presents tough verification and validation (V&V) challenges which need to be investigated before deployment for use in a spaceport environment. Conventional open-loop systems use mostly sequential scenarios, in which most events are visible as external commands and monitoring. They can be verified by black-box testing techniques, which consist in providing sequences of input and observing the generated output. In contrast, a Livingstone-based controller closes the control loop and arbitrates resources on-board with specialized reasoning, even in unforeseen situations. Because of this, the range of possible scenarios becomes very large and uncontrollable from the outside, so that black-box testing provides a very poor coverage.

Symbolic Model Checking of Livingstone Models

Model checking is an analytical V&V technique based on exhaustive exploration of all possible executions of a (model of a) dynamic system. It can provide a much better coverage than traditional testing. It can also be applied at an earlier stage in the development process, thus reducing the costs of fixing errors. Model checking is limited by state space explosion: the number of cases to be explored grows exponentially in the size of the system. In practice, the key issue in model checking real-size systems is to construct an abstract model that is small enough to be tractable, yet precise enough to reveal useful facts about the design.

In collaboration with Carnegie Mellon University (CMU), NASA Ames is developing a V&V technologies for Livingstone applications using the SMV symbolic model checker from CMU [1].

NASA and CMU have developed MPL2SMV, a translator that converts Livingstone models into SMV's input language. The properties to be verified, expressed in a powerful temporal logic (CTL) or using pre-defined specification patterns, are added along with the Livingstone model and similarly processed by the translator. The translator thus enables model checking of Livingstone models by their developers in their Livingstone environment, without requiring them to use or learn the input language of the SMV tool.

Experimental Results

As a demonstration of the utility of the SMV tool to the KSC model developers a portion of the ISPP model containing an improper representation of flow admittance and connectivity within the model representing the flow of CO2 through the Sorption pump/ zirconia cell components of the ISPP. The SMV tool was used to detect this fault automatically by verifying whether or not flow specification properties, within the model were true or not. Flow properties, which were falsified, resulted in the generation of diagnostic traces, which also demonstrated SMV's potential ability to be used as a real-time debugger.

The current, still incomplete Livingstone model, features more than 100 variables and ranks among the medium-size Livingstone models. It produces a reachable state space of the order of 10^{50} states. This is way beyond reach of traditional testing technology, or even of explicit state model checkers, but still tractable for symbolic model checkers such as SMV: with the use of advanced functionality[1], it takes SMV about a minute to process and analyze this model.

Livingstone models already give an abstracted view of the system they describe, and therefore lend themselves well to model checking. This is further facilitated by the fact that Livingstone and SMV are based on a similar paradigm (synchronous transition systems). The current ISPP model can still be verified in full details and generality, but more complex models will likely require a more piecemeal approach: focusing on specific mission scenarios, most likely fault scenarios, or analyzing different components separately.

---

[1] Re-ordering of variables: the performance of BDD algorithms used in SMV depends strongly on an order among variables in the model, and SMV provides options to tune up that parameter.

Conclusions

The benefits of deployment of Livingstone -like technologies for both terrestrial and extra-terrestrial spaceport systems include the following:

- Flight rates can be increased by reducing the amount of time required to certify a vehicle for launch. System complexity correlates to certification complexity and diagnostic complexity should the certification disqualify the vehicle at some point. A Livingstone like system can aid in this reduction by 1) reducing the downtime needed for complex fault diagnostics, and 2) reduce the reliance on a large team of personnel and equipment which have to be coordinated and directed by allowing the system to perform lower level or hazardous control operations.

- By eliminating or reducing safety and functional requirements that require human intervention or actions .

- Design phase benefits include 1) design for on-board system test & check-out, and 2) formal, partially automated design verification using model-checking.

- This technology can be applied to almost any type of dynamic engineering system (e.g. electromechanical, fluid, electrical etc.).

Bibliography

[1] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and J. Hwang, "Symbolic model checking: 10^20 states and beyond", Information and Computation, vol. 98, no. 2, June 1992, pp. 142–70.

[2] A. R. Gross, K.R. Sridhar, W. E. Larson, D. J. Clancy, and G. A. Briggs. "Information Technology and the Autonomous Control of a Mars In-Situ Propellant Production System", to appear in Proceedings of the 50th International Astronautical Federation Conference, Amsterdam, Holland, October 1999.

[3] D. Rapp, "A Review of Mars ISPP Technology", JPL D-15223, Jet Propulsion Laboratory. March 31, 1998.

[4] Livingstone: Onboard model-based configuration and health management**.** Brian C. Williams and P. Pandurang Nayak, "A Model-based Approach to Reactive Self-Configuring Systems", Proceedings of AAAI-96, 1996.