Control and Simulation of an Autonomous Lunar Lander

Corey A. Ippolito\(^1\) and Jeremy Frank\(^2\)
NASA Ames Research Center, Moffett Field, CA 94035

The Autonomous Lunar Demonstrator (ALDER) project was developed at NASA Ames Research Center to demonstrate autonomous decision making on a Lunar lander spacecraft capable of performing autonomous hop maneuvers. Towards these objectives, a conceptual spacecraft design was created to accomplish the mission, a simulation environment was developed, and a control system was designed and implemented to control the spacecraft in response to and integrated with a suite of agents in a higher level autonomy architecture. This paper presents the design of the control system for this autonomous vehicle that can respond to the demands of the higher level reasoning algorithms, and describes the simulation model and environment created to test and demonstrate the conceptual mission.

I. Introduction

The ability to explore multiple sites on the surface of a remote planetary body with a single launch provides a much deeper understanding of planetary geology and greater acquisition of scientific knowledge than a mission that returns a limited sample. The Sojourner Rover mission and the ongoing Mars Exploration Rover missions of recent years have demonstrated the value of mobile robotic systems for planetary exploration. Advances in autonomy have been demonstrated on free-flying spacecraft such as Deep Space One\(^{10}\), Earth Observing One (EO-1) Autonomous Sciencecraft Experiment (ASE)\(^{11}\), and Space Technology Five (ST-5)\(^{12}\). Advances in computer vision coupled with autonomy have been demonstrated for descent and landing, e.g. the DIMES camera during the Opportunity landing\(^{13}\), and on-orbit operations, e.g. Orbital Express\(^{14}\). Low-cost future robotic exploration of the Lunar surface will depend on these new technologies, as they enable the conduct of surface operations safely and efficiently with minimal human intervention and oversight.

The goal of the Autonomous Lunar DEmonstratoR (ALDER) project in the Intelligent Systems Division at NASA Ames Research Center is to demonstrate the practical use of advanced autonomy in a hypothetical, but representative, autonomous robotic Lunar mission and spacecraft. This hypothetical mission relies on autonomous operation during several phases of the remote mission. Several agent technologies are incorporated into the higher level logic for planning, mission-management and diagnosis. The higher level reasoning system provides decision-making over long timescales (minutes to hours), which directs a hierarchical flight control system (FCS) layer to provide control of the spacecraft on shorter time-scales (down to millisecond time frame).

A key driver in the design of the FCS for this vehicle was the ability to communicate with, respond to commands from, and support the objectives of the higher level reasoning system. This paper outlines a conceptual mission design for the ALDER project, describes a point design of a supporting spacecraft developed in support of the mission, and presents a general architecture that supports ALDER goals and integrates a higher-level reasoning layer above an autonomous Flight Control System (FCS). This paper focuses particularly on the GNC architecture in support of autonomous Lunar lander vehicle operations.

Disclaimer- This paper presents a point design for a conceptual mission and a supporting spacecraft solely to support the goals of the ALDER project- that is, a conceptual mission involving a small Lunar lander suitable for exploring the benefits of infusing autonomy. The ALDER concept design presented in this paper does not represent a baseline or favored approach being considered by any ongoing NASA project office or any NASA program, either in the agency or in NASA Ames Research Center. This conceptual design is purely for the purposes of this technology simulation demonstration.

\(^{1}\) Research Engineer, NASA Ames Research Center MS 269-1, Moffett Field, CA 94035, AIAA Member.
\(^{2}\) NASA Ames Research Center MS 269-1, Moffett Field, CA 94035, Non- Member.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.
II. Conceptual Mission Profile and Science Objectives

An autonomous mobile Lunar explorer could hypothetically investigate many open scientific questions of current interest to planetary scientists. Among these are the nature of pyroclastic formations, surface composition investigations, understanding biological hazards of Lunar dust, determination of chemical composition of surface materials, characterization of subsurface geology, and investigation of the presence of resources such as oxygen, hydrogen and water in Lunar regolith. In addition to science goals, an autonomous mobile explorer could survey possible sites for potential human habitation. A brief survey of possible science payloads reveals that many are lightweight, low power, and able to fit within the context of a small lunar vehicle mission. Scientific measurements are conducted with a number of scientific objectives, for instance a Raman, neutron or gamma ray spectrometer can provide elemental composition and radiation hazard analysis; multispectral imagers can provide surface structure mineralogy information; ground penetrating radar samples can investigate subsurface structure, density, and permittivity; magnetometer measurements can measure induced fields and provide magnetic ground truthing; and seismometers can provide seismicity, landing characteristic and impact information. However, the design of the vehicle likely limits instrument payload mass on the order of 10 kg, which makes certain technical functions less feasible, such as sample acquisition and preparation using an robotic arm and grinder.

To achieve a realistic baseline design for the mission and vehicle, an initial preliminary design study was conducted by Jet Propulsion Laboratory’s Advanced Project Design Team (Team X). Based on preliminary sizing and costs, the autonomous Lunar mission was projected to be $210 million for a 10-14 Earth day mission without the launch vehicle. Designing spacecraft to survive the Lunar night significantly increases cost, which scopes Lunar landing sites selection and the mission duration. A mission to the Lunar equator permits on order of 10 sols (Earth days) during which sunlight provides maximum power and sufficient thermal protection. The mission is targeted for either a Falcon 9 or Delta II 2925H-10L launch vehicle. A low energy orbital transfer from the Earth to the Moon occurs utilizing a weak stability boundary trajectory; the transfer will take around 98 days and provide a lower delta-V landing requirement than alternative faster transfers. Following Lunar injection, the spacecraft’s entry, descent and landing system (EDS) performs a braking maneuver, descent maneuver, a lateral transfer and then landing maneuver near the equator on the Earth-facing side of the moon, requiring an estimated delta-V of 2571 m/sec. A single bi-prop main engine system provides propulsion for all delta-V maneuvers.

The vehicle must collect scientific data at a set of prospective pre-selected landing and survey sites, and reuses its EDS to enable surface mobility. The spacecraft must perform controlled soft landings in sunlight at each survey location. The spacecraft autonomously conducts scientific measurements at each survey location and relays the information back to mission control on Earth. Upon completion of the scientific operations, the spacecraft must conduct a controlled ‘hop’ to the next scientific survey site location, which consists of a takeoff from the surface, a lateral transfer, and a soft landing in daylight. Each survey location is expected to be 1 KM away from the previous survey site, but will likely be poorly characterized in terms of the knowledge of local hazards around the prospective site, leaving uncertainty in the exact location where a safe touchdown can occur. Scientific operations are repeated at each survey site location, and hops repeat until fuel is exhausted.

The mission profile for the automated hop is shown in Figure 1. The vehicle must perform a liftoff maneuver to achieve altitude, then perform a turn and burn maneuver which enters the spacecraft into a ballistic trajectory relative to the moon’s gravity field to achieve the lateral transfer. At the target location, a retro-burn and descent maneuver occurs to slow the vehicle to the desired hover target altitude. While hovering, the vehicle completes a hazard assessment utilizing onboard vision-based sensors, identifies a location to perform the landing, and then performs the landing maneuver.
Hop Mission Leg Profile

1. Liftoff
2. Pre-Burn
3. Orient for Ballistic Burn
4. Perform Ballistic Burn
5. Coast/Orient for Retro-Burn
6. Retro-Burn
7. Hover 30m AGL
   (Survey Landing Location, Determine Touchdown Coordinates)
8. Hover To Landing Coordinates
9. Land

~1km Distance

Figure 1. Autonomous Lunar Surface Mobility (Hop) Maneuver Profile

III. ALDER Spacecraft Design and Simulation

To support the ALDER mission concept, a point design of an autonomous Lunar spacecraft system was developed; a rendering of the structural design and system layout is shown in Figure 2. The major specification of the lander is listed in Figure 3. Two camera systems are installed on the legs of the spacecraft, of resolution 1600×1200 pixels with a 66 degree field of view, separated by 1.7 meters on the bottom of the spacecraft; this allows the spacecraft to identify and avoid hazards on the scale of 30 cm at 30 m altitude above the landing site.

Figure 2. ALDER Hopping Lunar Lander Design
• Mass
  - Dry Mass: 53kg
  - Fuel Mass: 10kg
• Propulsion
  - Bi-Prop- Nitrogen Tetroxide (N₂O₄) and Hydrazine (N₂H₄)
  - 600N Main Thrusters
  - Four 30N Attitude Control Thrusters
• Power
  - Li-Ion (200W-h / 3.6kg)
  - Solar Array Area: 2.55 m²
• Sensors
  - Officine Galileo AA-STR star tracker
  - Draper Labs MEMS IMU
    - Simulated with Crossbow NAV420 IMU Model
  - Radar Altimeter
    - Simulated with CMU MAX Sonar Model, 5-Ray Casts into Scene
  - Stereo-Pair Cameras (1.7m Baseline)
    - MSL MARDI-like imagers
      (600g/10W/4ms capture time, 1600x1200/66° FOV)

Figure 3. ALDER Lander Specifications

A. Simulation Model
   An initial design concept was developed in Solidworks and exported to a real-time simulation environment rendering tools. The simulation was conducted in a real-time simulation architecture, the Reflection Architecture¹⁵, developed to support real-time control and simulation testing of small vehicles. The total dry mass of the vehicle is 53kg, with 10kg fuel mass. Fuel burn was simulated as a decrease in weight, but the effects of inertial changes, fuel sloshing, and CG transfer was not simulated. The lander was simulated as a nine-body dynamic system, with eight sets of equality constraints that each simulated single-axis revolute joints. The main body mass and lander feet were simulated as an inertial box, with the legs simulated as cylinders. Figure 4 illustrates the layout and configuration of the bodies in simulation.

Figure 4. Simulation Modeling of the ALDER Lander

The system allowed for 23 temporary constraints to be added to simulated contact and friction inputs between the lander and the lunar surface. Each of the nine rigid bodies were modeled using thirteen states: \( x = [p \ q \ v_B \ w_B] \), where \( p \) is position of the lander relative to a surface fixed coordinate system, \( q \) is orientation in quaternions, \( v_B \) is
velocity in the body system, \( \mathbf{w}_B \) is angular velocity along the body axis system. A flat-moon lunar gravity model was assumed, which is sufficient for our hop distance of 1km. The governing differential equation of motion for each body is specified by the following differential equations.

\[
\frac{d}{dt}(\mathbf{p}) = \mathbf{R}_{B\rightarrow r} \mathbf{v}_B
\]

\[
\frac{d}{dt}(\omega_B) = -\mathbf{J}^{-1} \hat{\omega}_B \mathbf{J} \omega_B + \mathbf{J}^{-1} \mathbf{T}_B
\]

\[
\frac{d}{dt}\mathbf{q} = -\frac{1}{2} \Omega_0 \mathbf{q}
\]

\[
\frac{d}{dt}(\mathbf{v}_B) = -\omega_B \times \mathbf{v}_B + \mathbf{R}_{1\rightarrow B} \mathbf{g} + \frac{1}{m} \mathbf{F}_B
\]

(1)

The hard constraints in the system were appended to the dynamic equations using the following general form\(^\text{16}\).

\[
J_1 \mathbf{v}_1 + \Omega_2 \omega_1 + J_2 \mathbf{v}_2 + \Omega_2 \omega_2 = \mathbf{c} + C \lambda
\]

(2)

The spring dampener systems were modeled using simple viscous damping and linear spring equations.

\[
Force = k_d \cdot \mathbf{v}_{rel} + k_s \cdot stretch
\]

(3)

The system is propagated forward in time in simulation through a Runge-Kutta 4\(^{\text{th}}\) order integrator, with a two-step constraint solver that approximates solutions to the associated linear complementarity-problem (LCP) using a pivoting algorithm\(^\text{17}\) and approximating the LCP utilizing PD symmetric matrices, using the update specified in the Ref. 3 as

\[
\mathbf{x}^{r+1} = f\left(\mathbf{x}^r\right) = b + B \left| \mathbf{x}^r \right|
\]

(4)

The propulsion system is a bi-prop nitrogen tetroxide (\( \text{N}_2\text{O}_4 \)) 600 N main engine thruster with four 30 N attitude control thrusters. In simulation, propulsion response was simulated using a first order thruster dynamic response model.

The onboard electrical systems power is stored in a 200W-h battery system utilizing lithium-ion chemistry, with an estimated mass of 3.6 kg. The power system simulation model used the power curve of a NiMH battery supply, due to difficulties in finding a Li-Ion battery model validated in a space environment. The power loading on the battery system was based on expected power draws for operational subsystems in a given operational mode. Energy is regenerated utilizing an array of solar panels with total of 2.55\( \text{m}^2 \) of surface area, though energy recharging utilizing solar panels was not implemented in simulation given a lack of time.

### IV. Flight Control System Design

#### A. Software Architecture for High Level Autonomy

While high degrees of on-board automation have proven themselves in a variety of recent experiments, such software is still considered a risk in many space missions. For this reason, the high-level reasoning software architecture, shown in Figure 5, maintains a separation between highly reliable traditional flight software functions, and higher-order functions making use of automation technologies like planning, control agents, and intelligent systems health monitoring.
The Executive is the main coordination component of the architecture, tying many of the subsystems together. The executive is built from the Intelligent Deployable Execution Architecture (IDEA)\(^4\). The Executive requests and manages the general plans based on the current state of the system in order to achieve the goals of the mission, including the conditions under which various parts of the plan are considered complete, and manages the execution of that plan according to a system model that describes how the system works, coupled with high-level goals that the plan must achieve.

The Planner is built from the Extensible Universal Remote Operations Planning Architecture (EUROPA)\(^5\). The Planner is responsible for generating the executive’s mobility plan utilizing current state information, such as location, destination, current propellant and oxidizer levels, and battery state. It determines if there are any resource deficiencies preventing a successful operation. If solar power is not available, the planner must check that total energy consumption does not exceed energy in the battery. It is possible that there will be no place to land, and that this will only be determined once the spacecraft is hovering over the intended landing spot. In this case, the spacecraft will have to return to its takeoff location, requiring sufficient fuel and energy reserves.

The Reflexive Intelligent System Health Monitoring (RISHM) component is a simple fault detection system for the spacecraft, monitoring spacecraft parameters to look for problems that can be calculated using basic logical inference. For example, a sudden and sustained drop in available power indicates a problem with the spacecraft power system. RISHM is based on the Spacecraft Health Inference Engine (SHINE)\(^6,7\).

The Deliberative Intelligent System Health Monitoring (DISHM) is built on the Hybrid Diagnosis Engine (HyDE)\(^8\), and is capable not only of detecting faults, but of performing a search over possible diagnoses to determine what caused the fault based on a combination of observations of the spacecraft and an internal model of the spacecraft subsystems. The DISHM monitors faults in the main engine and attitude control system; inputs include pressure, temperature and on/off readings from the tanks, valves, and regulators.

The Visual Hazard Avoidance system was built using the NASA Vision Workbench. This system takes a pair of images acquired from the simulated downward facing cameras and analyzes the landing site to find a safe landing location. The system first constructs a disparity correlation map between the two images using a fast block stereo correlation algorithm. The disparity map is converted into a digital elevation map of the landing site, and the resulting scene is analyzed to find a region large enough for the spacecraft to land in that is sufficiently flat, has no obstacles in it, has no missing pixels in it (i.e. regions of uncertain geometry due to shadows, glare, or failed correlation), and is as close as possible to the center of the scene.
B. Flight Management Computer

The Flight Management Computer (FMC) comprises the software and (simulated) hardware that executes most of the low-level spacecraft commands, such as firing the main engine and attitude control system, acquiring camera images, and so on. This subsystem determines which engines are used, and for how long, in order to execute the movement actions of the spacecraft. It does so by commanding a Main Engine Control System (MECS) and Attitude Control System (ACS) to change modes, in a specified sequence, in order to move. The FMC system design and subcomponents are shown in Figure 6.

---

**Figure 6. Flight Management Computer System**

The command sequencing unit (CSU) is responsible for maintaining the active list of commands that are currently active. The active command in the CSU is referenced by the FMS during the mission; when a transition state is satisfied, the FMS instructs the CSU to transition to retrieve and activate the transition’s target command. The hop planner was provided a calculation to determine the parameters for the hop, including inclination angle and thrust firing. The final component of the FMS is the Flight Control System (FCS), which is responsible for implementing the control systems which govern thruster firings (main engine and attitude).

The list of FMS commands that are supported in the system are shown in Table 1. The update activity diagram for the FMS is shown in Figure 7. When a hop is executed, the FMS starts the process by instructing the planner to compute the details and parameters of the hop. The FMS then programs the hop commands into the CSU using the hop planner’s mission plan, and activates the first command state (executing the ‘on entry’ instructions). The FMS then continuously polls the active state, looking for transitions. When a transition occurs, the FMS activates the next state. The FMS commands are shown in Figure 8, detailing which modes are operational during the entire hop maneuver.
Table 1. FMS Command List and Associated FCS Modes

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
<th>MECS Mode</th>
<th>ACS Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Takeoff</td>
<td>Controlled takeoff from the lunar surface.</td>
<td>Altitude Attain/Altitude Hold</td>
<td>Lateral Velocity Cmd (zero)</td>
</tr>
<tr>
<td>Burn</td>
<td>Maintain attitude to the precision of the ACS while engaging the main engines.</td>
<td>BURN</td>
<td>Attitude Attain/Altitude Hold</td>
</tr>
<tr>
<td>Turn To</td>
<td>While coasting, turn to the specified orientation</td>
<td>OFF</td>
<td>Attitude Attain/Altitude Hold</td>
</tr>
<tr>
<td>Hover To</td>
<td>Maintain altitude and hover over a position</td>
<td>Altitude Attain/Altitude Hold</td>
<td>Lateral Position Command</td>
</tr>
</tbody>
</table>
Figure 8. Flight Management System Modes, Autopilot Modes
C. Attitude Control System and Main Engine Control System Design

The FCS is composed of two main control systems, the Main Engine Control System (MECS) and the Attitude Control System (ACS). The MECS and ACS are both cascaded mode-based control systems that support the main mission profile; the modes are shown in Figure 8. Each CSU command contains a list of instructions that are executed when the command becomes active (“on entry”) and when the command terminates (“on exit”). In addition, each command is associated with one or more MECS and ACS control modes. These control modes are listed in Table 2.

Table 2. ACS and MECS Modes List

<table>
<thead>
<tr>
<th>MECS Modes</th>
<th>ACS Modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertical Speed Command</td>
<td>Lateral Speed Command</td>
</tr>
<tr>
<td>Altitude Attain</td>
<td>Lateral Velocity Command</td>
</tr>
<tr>
<td>Altitude Hold</td>
<td>Attitude Command: Attain</td>
</tr>
<tr>
<td>Burn (Engine On)</td>
<td>Attitude Command: Hold</td>
</tr>
<tr>
<td>Engine Off</td>
<td>Disengage</td>
</tr>
<tr>
<td>Disengage</td>
<td>Pilot/Remote Control</td>
</tr>
<tr>
<td>Pilot/Remote Control</td>
<td></td>
</tr>
</tbody>
</table>

The attitude control system and main engine control system are shown in Figure 9 and in Figure 11. In this controller design, the MECS and ACS are largely decoupled; in order to affect lateral motion, the MECS maintains altitude while the ACS commands an angle that leans into the desired delta velocity direction. The normal MECS firing cycle will produce a horizontal acceleration depending on the attitude angle of the lander. This potentially could be made more efficient by coupling the MECS and the ACS in a coordinated maneuver that takes advantage of vertical accelerations, at the cost of added complexity.

Figure 9. Attitude Control System (ACS)

The ACS controller’s modes of operations (Table 2) represent loop closure at successively ‘outer’ loop blocks in the cascade seen in Figure 9. The ACS control system’s inner loop controller block, labeled “Attitude to Thruster”, is a phase-plane controller (as outlined in Ref. 2). The “Lateral Speed to Attitude” block and the “Position Error to Lateral Speed” blocks are proportional-derivative controllers.
Figure 10. Phase Plane Control Logic for ACS.
This diagram and details of the switching logic are described in Ref. 2.

The MECS controller’s modes of operations (shown in Table 2) are supported by the block diagram in Figure 11. The firing logic for altitude control are shown in the “Altitude Hold Controller” block. The vertical speed to main engine thruster controller block is implemented as a PID controller.

Figure 11. Main Engine Control System

D. Simulation Environment and Control Displays

In addition to the controller design and implementation, several graphical components and displays were created for simulation visualization, as well as graphical models for the various lander components generated from Solidworks design. Sensors such as the stereo vision pair imagers and the altitude radar were implemented in the simulation rendering display, although the actual simulated camera images sent to the vision system were generated in a separate process. Other instrument displays developed included a primary flight display, navigation display, data visualization graph displays, 3D scene visualization, and the mode control panel display. These displays comprise the operator’s control station window, shown in Figure 12. Graphic displays were modeled after the primary flight display of an MD-11 cockpit. The FMC model control panel allows operators to set FMC and autopilot modes, engage autonomous operations, or bypass the controller to allow direct pilot joystick commands into the simulation.
V. Conclusion

This paper outlined a conceptual mission design for the ALDER project, described a point design of a supporting spacecraft developed in support of the mission, and presented a general architecture that supports ALDER goals, integrating a higher-level reasoning layer above an autonomous flight control system for a representative Lunar lander vehicle. The unique requirements of autonomy and the specific maneuvering requirements were a key driver in the design of the flight management and control system for this conceptual Lunar lander vehicle. The success of the simulation studies demonstrated the ability of the flight control system to communicate with, respond to commands from, and support the objectives of the higher level reasoning system.

However, key deficiencies were uncovered during this integration exercise. The level of decision making provided by the higher level planning agents were insufficient for achieving optimal performance from the flight control system computer during mission execution. The authors hypothesize a solution could include a layer of optimization support to help bridge this gap and achieve better mission performance. This could be provided by a component dedicated to determining control laws to achieve optimal trajectories based on criteria and constraints established by the higher level reasoning system. A candidate system was investigated in Ref. 18. Identifying methodologies for more optimal integration between the flight computer and the higher level reasoning system remains an area for future research.

VI. References

9 “NASA Vision Workbench”, URL: http://ti.arc.nasa.gov/project/nasa-vision-workbench/ [cited 1 August 2009].