Performance Evaluation of a Cartesian Move Algorithm for the LSMS Family of Cable-Driven Cranes

Javier Puig-Navarro* and Ryan Bowers†
Analytical Mechanics Associates, Hampton, VA, 23666

Dominic Bisio‡, Amelia Scott§, Matthew Vaughan¶, Joshua Moser‖, Jessica Friz**, Walter Waltz††, Julia Cline‡‡, and Danette Allen§§
NASA Langley Research Center, Hampton, VA, 23681

The Lightweight Surface Manipulation System, or LSMS, is a family of cable-actuated cranes, designed for surface operations on the Moon and Mars. Its structural design focuses on three fundamental traits for space environments: a high payload-to-mass ratio to reduce launch costs, compactness for space travel in reduced cargo bays, and self-erecting capabilities for operations before astronauts arrive. The resulting design confers nonlinear and switched dynamics to the LSMS, making these cranes an interesting challenge for autonomous and teleoperated applications. One of the protocols required for these applications is a Cartesian move algorithm. This algorithm is responsible for placing the end effector of the crane at a desired location, specified in Cartesian coordinates. This paper evaluates the performance of a Cartesian move algorithm developed for the LSMS at the NASA Langley Research Center. To this end, several payload manipulation tasks were performed with LSMS-L35, the smallest version within the LSMS family. The evaluated algorithm can place the end effector with accuracy, as shown in the hardware and simulation tests.

I. Nomenclature

LSMS	=	Lightweight Surface Manipulation System
θ_i	=	ith joint angle, with $i \in \{0, 1, 2, 3\}$
$\boldsymbol{\theta}$	=	vector of joint angles
ω_{c_i}	=	ith motor speed command, with $i \in \{0, A, B\}$
$\boldsymbol{\omega_c}$	=	vector of motor speed speed commands
$\boldsymbol{p_{goal}}$	=	goal position for the end effector
$\boldsymbol{p_{hook}}$	=	position of the end effector
$\boldsymbol{e_p}$	=	position error for the end effector
$\boldsymbol{e_{p_i}}$	=	ith coordinate of the position error, with $i \in \{x, y, z\}$
μ	=	mean
σ	=	standard deviation
$\{i\}$	=	ith reference frame, with $i \in \{\mathcal{G}, 0, 1, 2, 3\}$

*Aerospace Engineer Senior, MS 152, Autonomous Integrated Systems Research Branch, AIAA Senior Member.
†Associate Software Developer, MS 152, NASA Langley Research Center, Hampton, VA, 23681.
‡Robotics and Automated Systems Research Engineer, MS 152, Autonomous Integrated Systems Research Branch.
§Research Computer Scientist, MS 152, Autonomous Integrated Systems Research Branch.
¶Research Computer Scientist, MS 152, Autonomous Integrated Systems Research Branch.
‖Research Aerospace Engineer, MS 152, Autonomous Integrated Systems Research Branch, AIAA Member.
**Aerospace Engineer, MS 125B, Simulation Development and Analysis Branch.
††Research Computer Scientist, MS 152, Autonomous Integrated Systems Research Branch, AIAA Member.
‡‡Research Aerospace Engineer, MS 190, Structural Mechanics and Concepts Branch, AIAA Senior Member.
§§Senior Leader for Autonomous Systems and Autonomous Systems Capability Lead, NASA, AIAA Associate Fellow.

II. Introduction

The Lightweight Surface Manipulation System, or LSMS, is a series of scalable, long-reach, cable-actuated, robotic cranes. These robots are designed for construction applications on gravitationally rounded objects, such as the Moon and Mars. When equipped with a standard interface, the LSMS can conduct tool changes to perform a variety of tasks [1]. Some of these tasks include: spacecraft loading and off-loading, regolith manipulation for the construction of infrastructure, remote inspection of structures, and deployment of science payloads. To achieve this variety of tasks, the design of the LSMS considered a multitude of factors [2]. It is versatile to avoid launching multiple special-purpose platforms to the Moon [3] and lightweight to reduce launch costs. The LSMS has a high payload-to-mass ratio to maximize its utility in space and a compact form factor to fit in tight cargo bays. In addition, it can self-unfurl from its stowage configuration, which enables operations before astronauts arrive.

The LSMS family is composed of two prototypes. LSMS-L35 is the smallest platform with a reach of 2.25 m [4], shown in Fig. 1. LSMS-L1000 is the largest version with a reach of 7.00 m [5, 6]. The naming convention that succeeds the acronym LSMS specifies the celestial body the crane is designed to operate on and the maximum mass it can lift on the surface of that body. The letter L is used for lunar operations, while M for Martian operations. Thus, LSMS-L35 can safely lift a maximum of 35 kg on the lunar surface from the wrist, while LSMS-L1000 can lift a maximum of 1000 kg. Elbow lifts are also possible and often double the maximum mass specified for a wrist lift. The links and joints on LSMS cranes are named using an anthropomorphic metaphor. The waist joint is located at the base of the kingpost and is shown as point 0 in Fig. 1. The shoulder joint connects the kingpost and the arm link and is marked as point C. The elbow joint connects the arm with the forearm and is denoted as point G. Finally, the wrist joint connects the arm with the lifting link and is illustrated as point K in Fig. 1. The lifting link is a one-dimensional pendulum equipped with an end effector at point L in the shape of a two-pronged hook.

Recently, research and development efforts surrounding the LSMS have targeted capabilities that can be leveraged for the Artemis campaign and the Moon-to-Mars architecture. The Artemis campaign, currently scheduled for a crewed flight around the Moon in 2025, aims to advance new and existing technologies that will enable human missions to Mars [7, 8]. The Moon-to-Mars architecture defines NASA's strategy for crewed exploration of deep space. The architecture breaks NASA's strategy into ten areas. Those directly related to the LSMS efforts include *lunar infrastructure* and *Mars infrastructure*. The LSMS technologies can also be used to offload scientific payloads, aid astronaut operations on the Moon, and build a safe haven [9]. Thus, advancing the *science-enabling*, *operations*, and *transportation and habitation* areas within the Moon-to-Mars architecture. The latest developments have strengthened the autonomous and teleoperated capabilities of the LSMS. Some examples are the algorithm to reduce payload swinging at the end of a maneuver in [10], the joint trajectory tracking protocol in [11], and the definition of the safe operational envelope of the LSMS in [12].

This effort is part of the LSMS AutoNomy capabilities Development for surface Operations and Construction (LANDO) project at the NASA Langley Research Center. It aims to evaluate a novel capability designed for the LSMS family, a Cartesian move algorithm. This term is used in robotics to refer to a group of protocols responsible for moving the end effector of a manipulator to a particular location specified in Cartesian coordinates; see the section on Cartesian control in [13]. These algorithms are often used in manipulation tasks, such as grasping, scooping, moving, and inspecting objects. Their performance is important to reduce collision risks and increase task success rates in proximity operations. Cartesian move algorithms for manipulators use the Jacobian of the end effector position to map the position error to angular speed commands for the joints. However, the cable geometry shown in Fig. 1 makes this map more complex for the LSMS cranes. To determine if this new technology is feasible to perform payload manipulation tasks, this paper presents a preliminary study that analyzes the accuracy of the Cartesian move algorithm when placing the end effector at a target location.

The contributions of this paper are threefold: *i*) an precision study of the Cartesian move algorithm that uses LSMS-L35 as the test platform; *ii*) a set of recommendations on how and when to use the Cartesian move algorithm; and *iii*) an algorithm to be developed in the future for those applications that are not amenable to execution with the Cartesian move algorithm.

The rest of the paper is organized as follows. Section III provides an overview of the LSMS cranes with a particular focus on LSMS-L35, used in the hardware tests captured within this document. Section IV presents the structure, design criteria, and use cases for the Cartesian move algorithm. Section V describes the methodology used to evaluate the algorithm, as well as the hardware and simulation test results. Finally, Section VI closes the paper with a summary of the findings from the performance evaluation of the Cartesian move algorithm.

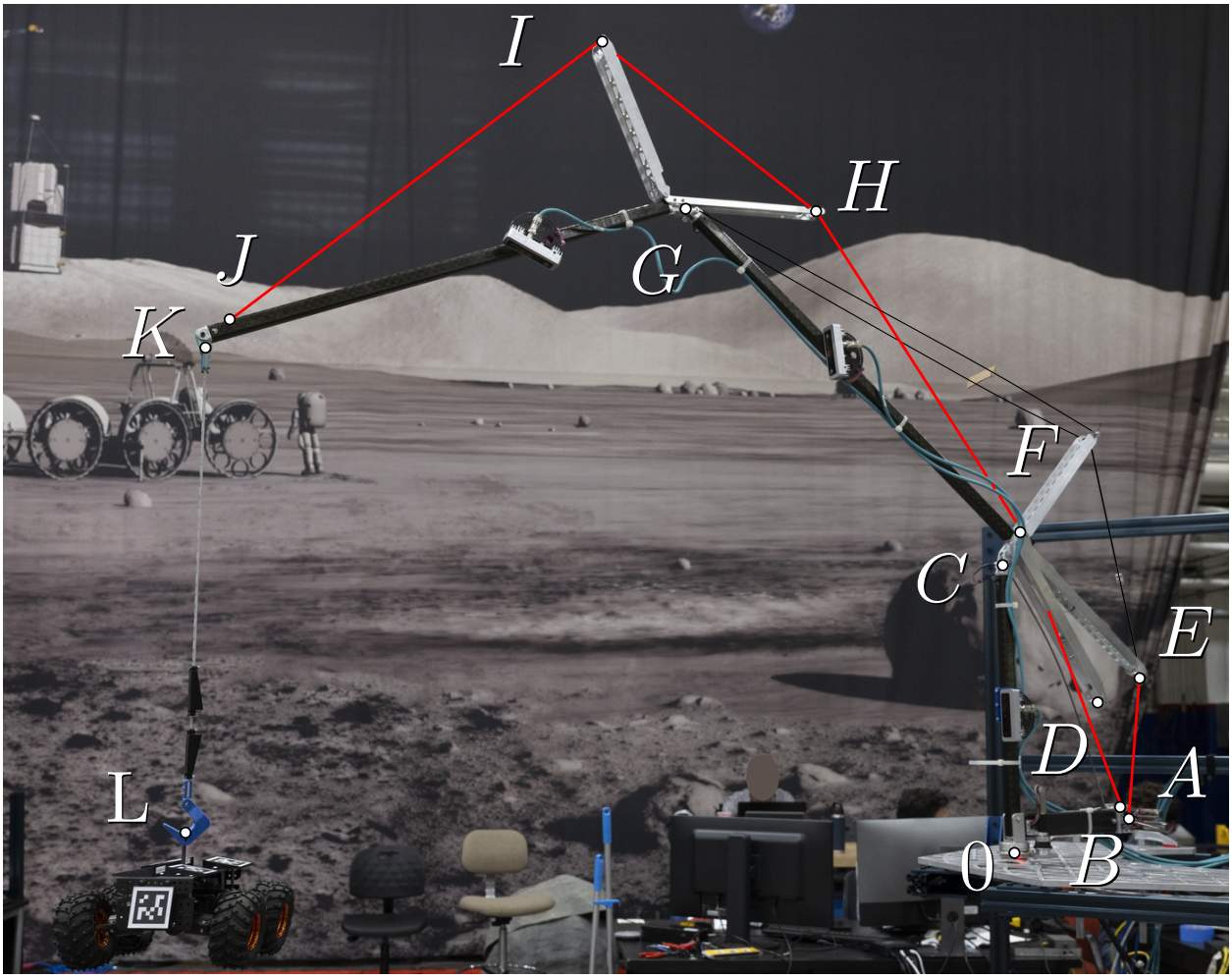


Fig. 1 LSMS-L35 using the Cartesian move algorithm to maintain the end effector at a specified location. The LSMS-L35 is shown in front of a lunar backdrop in a laboratory at the NASA Langley Research Center. Actuated cables are highlighted in red, while passive cables are highlighted in black for visibility. Relevant points on the LSMS are marked by a small circle and a capital letter. A and B mark the location of the motors, gearboxes, and cables drums that control the motion of the cables. Points C, G, and K denote the location of the shoulder, elbow, and wrist joints respectively. Points E and J highlight the attachment of the cables to the arm and forearm links. Point F shows the location of a pulley embedded in the structure of the LSMS. Pulley F routes the cable that controls the elbow joint. Finally, points D, H, and I illustrate the location of a pulley at the end of the spreaders, where the cables rest for certain crane configurations.

III. The Lightweight Surface Manipulation System

The LSMS is a family of cranes with three or more degrees of freedom, depending on the end effector installed. Fig. 2 shows a schematic of the LSMS with four degrees of freedom: the waist angle θ_0, the shoulder angle θ_1, the elbow angle θ_2, and the wrist angle θ_3. The waist angle is actuated by a gearmotor with gear ratio i_0, located at the base of the LSMS. The shoulder angle is controlled by a cable attached at point E and actuated by motor A, equipped with a gearbox with gear ratio i_A, and a cable drum with radius r_A. Similarly, the elbow angle is controlled by a cable attached at point J and actuated by motor B, equipped with a gearbox with gear ratio i_B, and a cable drum with radius r_B. Point F in Fig. 2 marks the location of a pulley embedded within the structure of the LSMS. This pulley routes the cable that controls the elbow angle. The position of pulley F depends on the shoulder angle, which creates a coupling between the shoulder and the elbow movement. See [11] for further details about the shoulder-elbow coupling and a proof of this phenomenon. Point D in Fig. 2 marks the location of a pulley at the end of the spreader. As the shoulder angle decreases, there exists an angle θ_1^* where the pulley D rests on the cable. Note the positive direction indicated by the arrow in Fig. 2. Similarly, there exist angles θ_2^* and θ_2^{**} where spreaders H and I become load bearing. The purpose of spreaders D, H, and I is twofold: *i*) efficiently distribute the forces and torques along the structure, and *ii*) extend the operational envelope of the LSMS, see [12]. These spreaders create discrete switches in the behavior of the LSMS as they rest on and separate from the cables, which adds complexity to the algorithms used for autonomous operations.

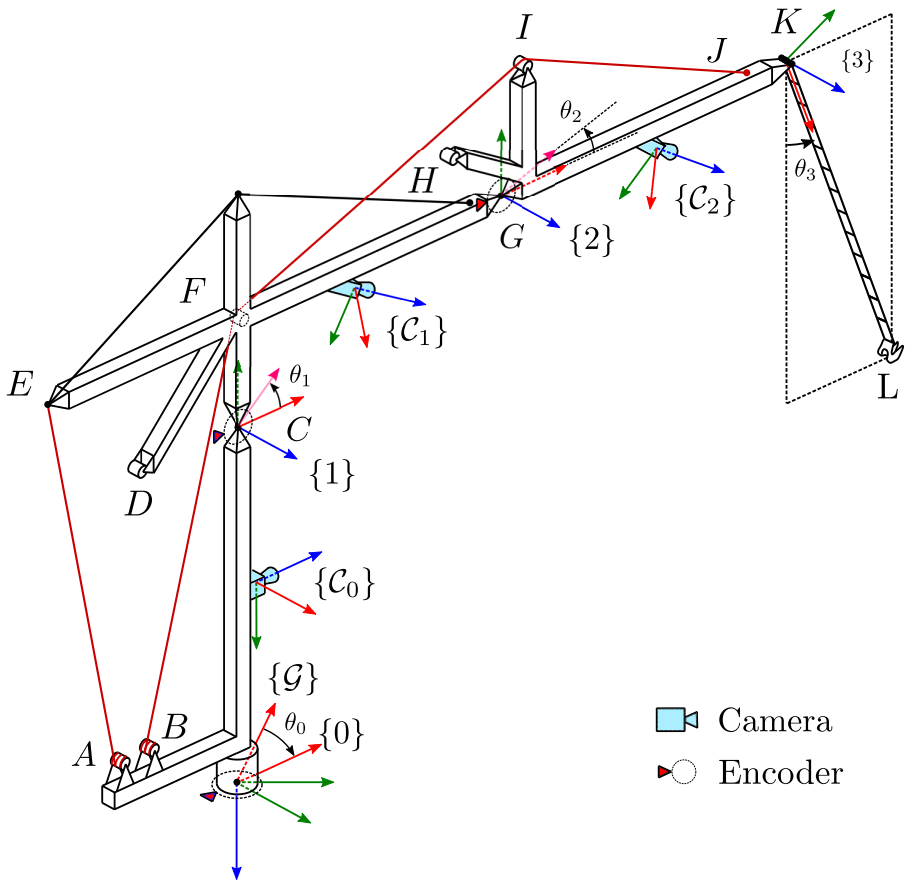
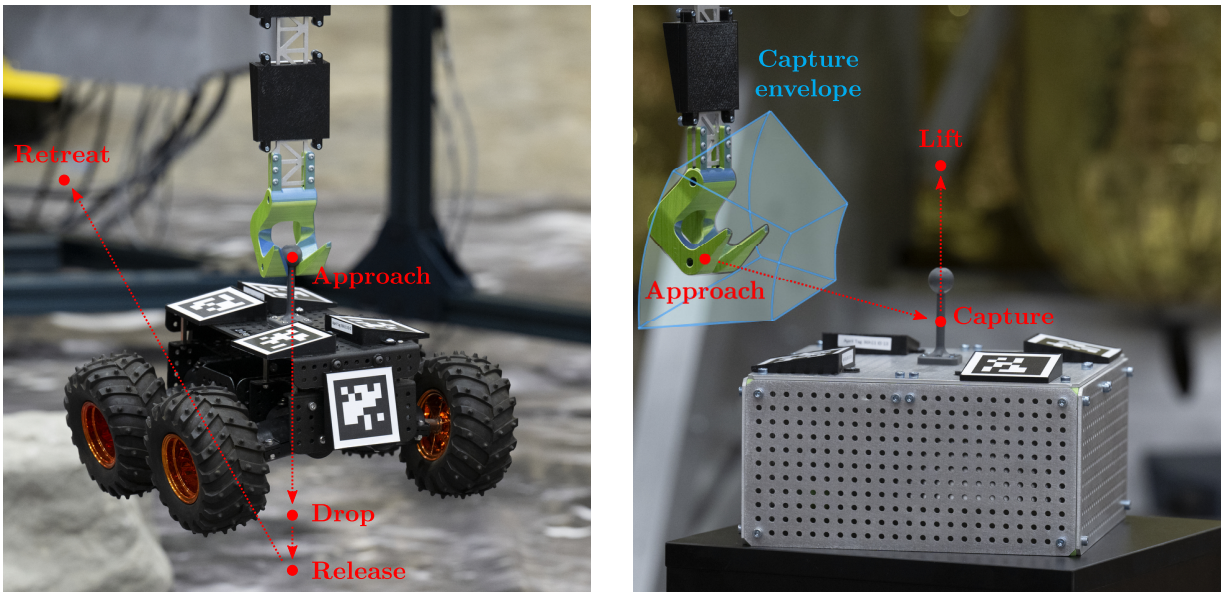


Fig. 2 Schematic of LSMS. The legend contains the symbols used to represent the sensors on the LSMS. Passive cables are highlighted in black, whereas actuated cables are depicted in red. Reference frames $\{\mathcal{G}\}$, $\{0\}$, $\{1\}$, $\{2\}$, and $\{3\}$ are the ground, waist, shoulder, elbow, and wrist frames. These reference frames are defined so all rotations occur about their z axes. Reference frames $\{C_0\}$, $\{C_1\}$, and $\{C_2\}$ mark the location and orientation of the camera sensors on the kingpost, arm, and forearm links. All reference frames are color coded using red, blue, and green for the x, y, and z axes, respectively. Relevant points in the structure are labeled using capital letters. In this configuration spreader I rests on the cable, while spreaders D and H do not rest on their cables. The spreader configuration changes with θ_1 and θ_2.

The LSMS is equipped with three magnetic encoders located at the waist, shoulder, and elbow joints. See reference frames {0}, {1}, and {2} in Fig. 2. The wrist joint is neither actuated nor instrumented and is marked by reference frame {3}. The absence of an actuator lets the lifting link rotate freely around the wrist. The lack of an encoder requires that the Cartesian move algorithm infer the wrist angle to estimate the position of the hook, marked as point L on the lifting link. For the purposes of this paper the lifting link has a fixed length. Alternatively, a variable-length lifting link or a reeling cable can be installed. The angles provided by the encoders are used by the Cartesian move algorithm in real time as control feedback.

The LSMS is equipped with three additional sensors: a wide-angle camera installed on the kingpost, at reference frame $\{C_0\}$; and two narrow-angle cameras on the arm and the forearm, denoted by reference frames $\{C_1\}$ and $\{C_2\}$. While camera information is not used as real-time feedback for the Cartesian move algorithm, it is used to determine the location of various points of interest in the workspace. A relevant example is the location of the pick-up fixture on the payloads.

Fig. 3 shows the two payloads used to test the Cartesian move algorithm. Both payloads have a pick-up fixture that consists of a slender rod topped with a small sphere. The end effector of the LSMS is also visible in the figure, a two-pronged hook whose base matches the shape of the sphere in the pick-up fixture. Fig. 3a illustrates how the sphere is nestled at the base of the hook when lifting the payload. The fiducials on the rover and the box are used by the perception system to identify and estimate the position and orientation of the payloads. These values are leveraged by the robot to define multiple target locations for the end effector. Once computed, these target locations are ingested by the Cartesian move algorithm. The following section describes the structure and use cases for the Cartesian move algorithm in detail.



(a) Rover being dropped-off by LSMS-L35. **(b) Box being picked-up by LSMS-L35.**

Fig. 3 Payloads used to test the algorithms that enable autonomous operations on the LSMS.

IV. The Cartesian Move Algorithm

The objective of the Cartesian move algorithm is to ensure that the end effector converges to and remains at a specified location. This is particularly relevant for tasks like payload pick-up and drop-off. To this end, the Cartesian move algorithm takes in a constant goal position $\boldsymbol{p}_{goal}$, as shown in Fig. 4. In control terms, this location is the reference position for the end effector. Internally, the algorithm compares $\boldsymbol{p}_{goal}$ with the estimated location of the end effector $\boldsymbol{p}_{\boldsymbol{hook}}$ and computes the speed commands $\boldsymbol{\omega_c} := [\omega_{c_0}, \omega_{c_A}, \omega_{c_B}]^\top$ for motors 0, A, and B. These commands bring the end effector to the goal position. To estimate the position of the end effector, the algorithm combines the encoder feedback $\boldsymbol{\theta} := [\theta_0, \theta_1, \theta_2]^\top$ with the forward kinematics of the LSMS, whose dimensions are obtained from a CAD model. In this process, the algorithm assumes that the lifting link remains vertical to estimate the wrist angle θ_3. This assumption may not be realistic for aggressive maneuvers. However, the LSMS is designed to move slowly.

The Cartesian move algorithm has a nonlinear proportional structure. Its design achieves a linear motion from the initial location of the end effector to the goal position under ideal conditions. That is, in the absence of modeling errors, unmodeled dynamics, and external disturbances, the end effector should trace a straight segment from its initial location to the goal location when placed within some neighborhood of the goal location. The algorithm also incorporates a saturation scheme for the motor speed commands that protects the actuators and preserves the linear motion.

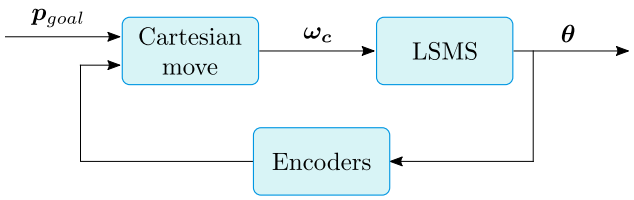


Fig. 4 Schematic of the Cartesian move algorithm.

The Cartesian move algorithm is designed for close-proximity payload manipulation tasks, such as pick-up and drop-off operations. To understand the use of this algorithm, it is helpful to break down the pick-up procedure in three phases:

i) *Approach*: uses a joint trajectory generation algorithm to plan a time-parameterized sequence of joint angles that move the end effector from an arbitrary location to a point in the capture envelope of the payload*. See Fig. 3b. This trajectory avoids obstacles in the workspace, whose geometry can be produced by the perception system on the LSMS, and respects the physical limits of the crane. To execute the trajectory, the LSMS uses the trajectory tracking algorithm proposed in [11].

ii) *Capture*: once the end effector is within the capture envelope of the payload, the perception system produces a goal position. This point is located halfway between the base of the pick-up fixture and the sphere atop the pick-up fixture, as shown in Fig. 3b. Then, the goal position is handed to the Cartesian move algorithm, which drives the end effector to the capture location.

iii) *Lift*: once the end effector has the pick-up fixture between its prongs, the perception system generates a new goal location. This point is located above the sphere atop the pick-up fixture. The lift location is depicted in Fig. 3b. Then, the new goal position is handed to the Cartesian move algorithm, driving the end effector upward and lifting the payload.

A similar pattern, but in reverse and slightly more complicated, occurs for payload drop-off tasks:

i) *Approach*: uses a joint trajectory generation algorithm to plan a time-parameterized sequence of joint angles that move the end effector from an arbitrary location to a point above the desired drop-off location. This drop-off location must be reachable and may be specified by the onboard autonomous software or an operator. In this case, the trajectory generation algorithm shall consider that the payload is attached to the end effector to avoid running into obstacles. Again, the joint trajectory is executed using the algorithm described in [11].

ii) *Drop*: once the payload is above its drop-off location, the perception system generates a drop position that lets the payload rest on its supporting surface, while the sphere atop the pick-up fixture is still nestled within the end effector, as illustrated in Fig. 3a. Then, the goal location is handed to the Cartesian move algorithm, which lowers the payload until it touches its supporting surface.

iii) *Release*: once the payload is on the surface, the perception system computes a new goal location. This point is halfway between the sphere and the base of the of the pick-up fixture. Then, the new goal is handed to the Cartesian move algorithm, which further lowers the end effector to the release location; see Fig. 3a.

iv) *Retreat*: once the end effector no longer carries the weight of the payload, the perception system computes a new goal location. This point ensures that the end effector does not graze the top surface of the payload as it retreats. It also guarantees that the prongs of the end effector stay clear of the sphere atop the pick-up fixture. Finally, the new goal position is handed to the Cartesian move algorithm, which moves the end effector away from the payload.

The task break-down above intends to give context for the use and relevance of the Cartesian move algorithm. This effort does not evaluate the performance of these tasks, which rely on many other protocols. Further details about the Cartesian move algorithm are also beyond the scope of this paper. Future publications will include the mathematical

*The capture envelope is a region in the workspace that ensures the pick-up fixture is captured when the end effector is placed within and the Cartesian move algorithm is engaged. Fig. 3b includes an approximate overlay of the capture envelope onto the picture.

definition of the algorithm, and theoretical performance guarantees. The following section presents a preliminary study used to determine whether the Cartesian move algorithm is an adequate approach for payload manipulation tasks.

V. Performance Evaluation

To evaluate the performance of the Cartesian move algorithm, define the end effector position error

$$\boldsymbol{e_p} \coloneqq \boldsymbol{p_{hook}} - \boldsymbol{p_{goal}}, \tag{1}$$

where $\boldsymbol{p_{hook}}$, $\boldsymbol{p_{goal}} \in \Omega_r$; $\Omega_r \subset \mathbb{R}^3$ is the reachable workspace of the LSMS; and the goal position is constant, that is $\dot{\boldsymbol{p}}_{\boldsymbol{goal}} = \boldsymbol{0}$. Recall that the goal of the Cartesian move algorithm is to drive $\boldsymbol{e_p}$ to zero

$$\|\boldsymbol{e_p}\| \xrightarrow{t\to\infty} 0. \tag{2}$$

Recall also that the Cartesian move algorithm is designed to make the end effector trace a straight segment from its initial location to the goal position under ideal conditions, which require the absence of modelling errors, unmodeled dynamics, and external disturbances. In addition, the team selected three thresholds to characterize the performance the Cartesian move algorithm under relaxed assumptions on errors, classifying the test results as:

i) *Desired*: for errors that are smaller than or equal to 10^{-3} m.
ii) *Minimum*: for errors that are greater than 10^{-3} m but smaller than or equal to $5 \cdot 10^{-3}$ m.
iii) *Unsatisfactory*: for errors above $5 \cdot 10^{-3}$ m.

These thresholds are compared with maximum and average errors in the subsequent subsections. The threshold values are selected considering a multitude of factors: the set of expected payload manipulation tasks, the geometry of the end effector and the pick-up fixture, and other sources of error that contribute to the success or failure of the manipulation tasks such as the perception algorithm.

With this in mind, this section is partitioned into four subsections. Section V.A describes the methodology used to test the algorithm. Section V.B presents hardware test results on LSMS-L35, performed in the laboratory under conditions that do not meet the definition of ideal assumptions given above. Section V.C shows simulation test results that do replicate ideal assumptions. Finally, section V.D provides a discussion about both the hardware and simulation test results, as well as the adequacy of the Cartesian move algorithm for payload manipulation tasks.

A. Methodology

To evaluate the performance of the Cartesian move algorithm and other systems within the autonomy stack, ten consecutive tests were performed with LSMS-L35. Each test included an autonomous pick-up task and an autonomous drop-off task. LSMS-L35 is equipped with monitoring software to determine if a pick-up task fails or succeeds. This information is leveraged to autonomously reattempt the pick-up task a maximum of three times. After the third failed attempt, the task is marked as failed and the test is terminated. The drop-off task does not employ monitoring software to determine if the task fails or succeeds. As a result, autonomous reattempts at the drop-off tasks have not been implemented at the time of this report. Instead, the LSMS operator marks the task as failed and terminates the test after the first failed attempt. This is a temporary solution implemented given the time constraints of the project.

Each test had a fixed set of planned human interactions:

i) *Payload placement in the workspace*: performed by a team member. To this end, the team used two pedestals and the ground to place the payloads. The pedestals had different heights, and their location was often changed within the workspace. The autonomy stack was equipped with an algorithm to determine whether a payload was reachable within the space visible to LSMS-L35. If the initial location of the payload was not reachable, then a team member would reposition the payload.
ii) *Payload selection*: performed by the LSMS operator, who chose between the rover or the box through a user interface. The two payloads were always present in the workspace during tests.
iii) *Selection of the payload drop-off location*: performed by the LSMS operator through a user interface. If the initial drop-off location was not within reach of LSMS-L35, the operator was prompted to select a new and reachable location.

All other human interventions led to the task being marked as failed and the test terminated, even if the intervention did not require physical access to the hardware. As a result, each test could engage the Cartesian move algorithm a different number of times, depending on the tasks completed and the number of reattempts. The following signals were recorded

during the tests: hook and goal positions with respect to the ground frame $\{\mathcal{G}\}$, position error, norm of the position error, joint angles, and motor speed commands. In total, 49 different data records where the Cartesian move algorithm was engaged were collected.

The goal location was generated by the perception algorithm. To decouple the performance of this algorithm from the perception algorithm, we assume that the goal location accurately represents the desired end effector location. This paper compares the goal location with the estimate of the end effector location within the Cartesian move algorithm. However, on multiple occasions, the goal position does not coincide with the actual desired location for the end effector. For example, this often holds for the sphere atop the pick-up fixture, due to measurement errors in the perception algorithm. This aspect can lead to scenarios where the Cartesian move algorithm accurately moves the end effector to the goal location, but the given goal location does not succeed in meeting the task requirements. Thus, we note that the success of the Cartesian move algorithm is necessary but not sufficient for the successful completion of a task. The performance evaluation of the perception algorithm is beyond the scope of this paper. The performance evaluation of the task success is also beyond the scope of this paper, but it depends both on the perception and Cartesian move algorithms. The following section presents the hardware test results with LSMS-L35.

B. Hardware Test Results

Table 1 presents the summary statistics for the 49 data records obtained during the hardware tests with LSMS-L35. The columns contain the x, y, and z coordinates of the position error, as well as the Euclidean norm of the position error. The maximum, average μ, and standard deviation σ are computed using the last sample available across all data records; see Table 2 in the Appendix. The last sample available for each data record corresponds to the last time step the Cartesian move algorithm was executed.

Table 1 Summary statistics for the hardware tests on LSMS-L35.

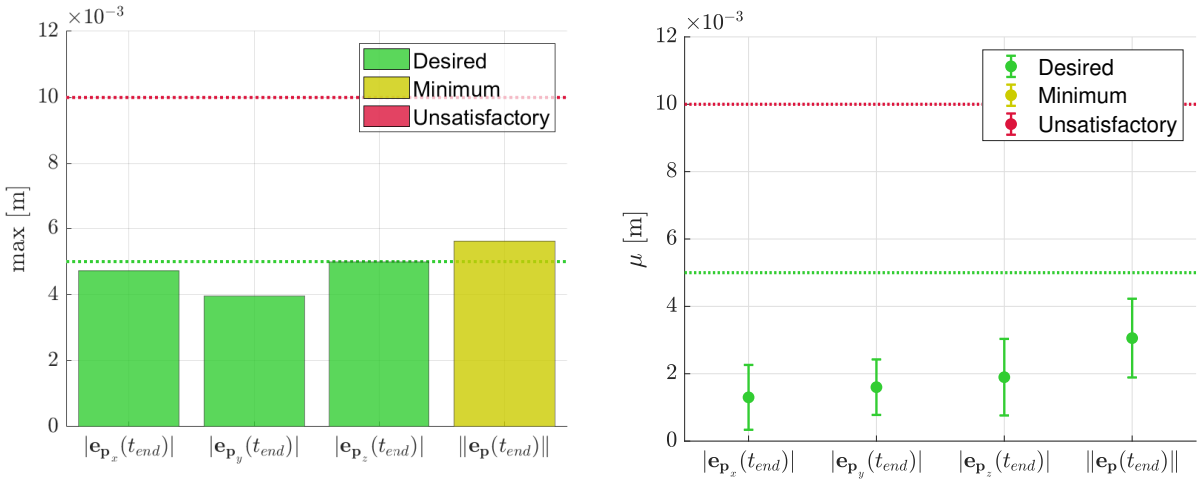
	$\lvert \boldsymbol{e}_{\boldsymbol{p}_x}(t_{end}) \rvert$	$\lvert \boldsymbol{e}_{\boldsymbol{p}_y}(t_{end}) \rvert$	$\lvert \boldsymbol{e}_{\boldsymbol{p}_z}(t_{end}) \rvert$	$\lVert \boldsymbol{e}_{\boldsymbol{p}}(t_{end}) \rVert$
max [m]	0.0047	0.0040	0.0050	0.0056
μ [m]	0.0013	0.0016	0.0019	0.0031
σ [m]	0.0009	0.0008	0.0011	0.0012

Fig. 5 illustrates the results in Table 1. Fig. 5a shows the maximum absolute value of the various contributions to the position error, as well as maximum Euclidean norm of the position error across all records evaluated at the end of each run. The x, y, and z contributions are colored green to highlight that the desired outcome is met. The Euclidean norm of the position error is colored yellow, denoting that the minimum outcome is satisfied. Fig. 5b shows the average absolute value of the different contributions to the position error, as well as the average Euclidean norm of the position error. The error bars represent one standard deviation. All markers are colored green, indicating that the desired outcome is satisfied

Fig. 6 provides some granularity about the distribution of the different contributions to the position error. The histograms show a count of the data records whose position errors at the end of the run fall within the specified bins. The width of the bins is 10^{-3} m. Note that Figures 6a through 6c no longer use the absolute value, to identify biases in certain directions. Fig. 6d shows the results for the Euclidean norm of the position error. In most scenarios, errors fall within 10^{-3} and $3 \cdot 10^{-3}$ m. Note also that only three samples out of 49 fall beyond the desired outcome of $5 \cdot 10^{-3}$ m.

Fig. 7 shows the relevant signals collected during data record 6 to visualize the behavior of the Cartesian move algorithm. Fig. 7a illustrates the evolution of the hook and goal positions over time. The end effector converges to the goal location without overshooting the target. Fig. 7b presents the motor speed commands generated by the Cartesian move algorithm. The speed command for motor B is saturated for almost 5 s at the beginning of the record, showing that the saturation scheme engages to protect the actuators. This saturation scheme was specifically designed for this Cartesian move algorithm. The motors include a gearbox with a high gear ratio, leading to much lower angular speeds on the cable drums. The values in Fig. 7b represent the motor speed commands before the gearbox.

Fig. 7c shows the evolution of the position error over time. As expected, the position errors converge to a small neighborhood around zero. The z error shows a smaller rate of convergence than that of the x and y errors. The tuning gains of the Cartesian move algorithm were chosen to achieve the same convergence rate. The result is likely caused by modelling errors or unmodeled dynamics, such as the elasticity of the cables. Fig. 7d displays the evolution of the norm of the position error over time. As expected, the norm converges to a small neighborhood around zero.



(a) Maximum errors. **(b) Average errors.**

Fig. 5 The summary statistics for the hardware test results with LSMS-L35. The desired, minimum, and unsatisfactory outcomes are color coded in green, yellow, and red, respectively. All parameters meet the desired outcome, except the maximum norm of the position error evaluated at t_{end}, which only meets the minimum outcome.

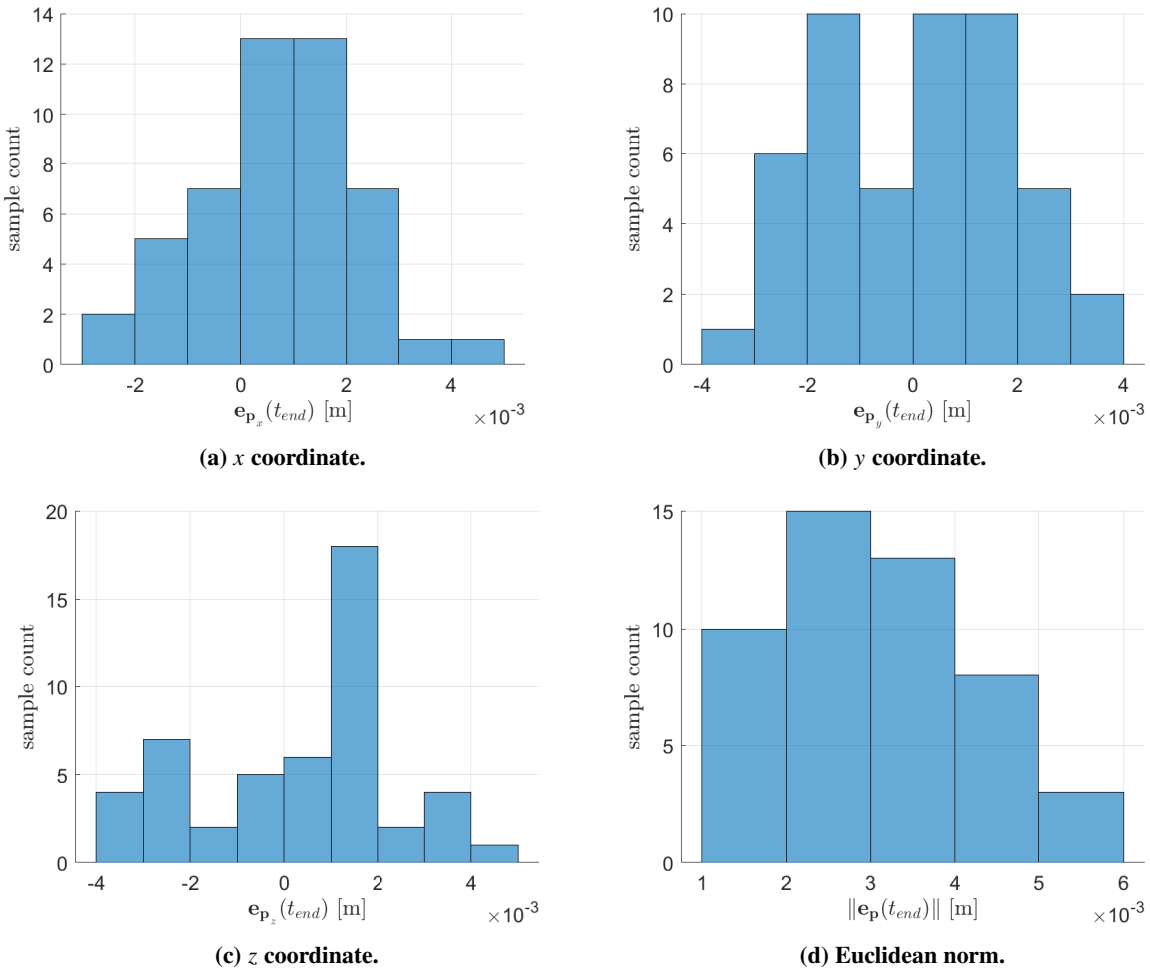
Fig. 7e shows the evolution of the angles for the waist, shoulder and elbow joints. The joints experience a smooth and well-behaved motion. The waist motion is minimal compared to the shoulder and elbow motions, since the start and end of the retreat are both within the plane of symmetry of the LSMS. Finally, Fig. 7f depicts the trace of the end effector in the workspace as it moves towards the goal location. Notice that the end effector does not trace a straight segment. The z error takes longer to correct than the x and y errors, as inferred from Fig. 7c. This indicates a departure from ideal conditions during the hardware test. This departure is expected as real systems have noise, modelling errors, unmodeled dynamics, and external disturbances, such as forces that may appear when the end effector is in contact with the pick-up fixture. Despite these uncertainties, the Cartesian move algorithm achieves its goal with precision and brings the end effector to its goal location.

C. Simulation Test Results

Fig. 8 presents the same signals as Fig. 7, but for a simulation that includes an identical implementation of the Cartesian move algorithm, the same tuning gains, the same initial position for the end effector, and an identical goal location. While in Fig. 7 the motor commands were sent to LSMS-L35 and the joint angles were collected from the actual sensors, in Fig. 8 the motor commands were sent to an ideal model of LSMS-L35 and the joint angles were collected from ideal models of the encoders. In other words, no noise or sensor biases, no unmodeled dynamics such as the underlying actuator dynamics, and no external disturbances are present.

The signals in Fig. 8 show similar trends as those in Fig. 7. Compare Figures 7a, 7c, and 7d with Figures 8a, 8c, and 7d. As expected, the position errors now converge to zero, not just a small neighborhood around zero. The rate of convergence of the position errors is slightly faster than for the hardware tests. This is likely caused by the absence of the underlying motor dynamics, which tend to introduce a time delay and small tracking errors that can slow down convergence. The largest difference between Figures 7c and 8c is in the z error. Three possible contributors to this trend, besides the underlying motor dynamics, may be: *i*) modelling errors, *ii*) unmodeled dynamics like the elasticity of the cables that control the shoulder and the elbow, and *iii*) external disturbances such as contact forces between the pick-up fixture and the end effector.

The speed commands in Fig. 8b also exhibit saturation, but for a shorter period of time than in Fig. 7b. The joint angles in Fig. 8e are smooth and well-behaved, but move slightly faster than those in Fig. 7e. Finally, Fig. 8f shows that the trace of the end effector is a straight segment. This confirms that the algorithm behaves as expected under ideal conditions even when the saturation scheme engages. The saturation protocol seen here was specifically designed for this algorithm to preserve the linear motion of the end effector. The simulation results increase confidence in a correct



(a) x coordinate. **(b) y coordinate.**

(c) z coordinate. **(d) Euclidean norm.**

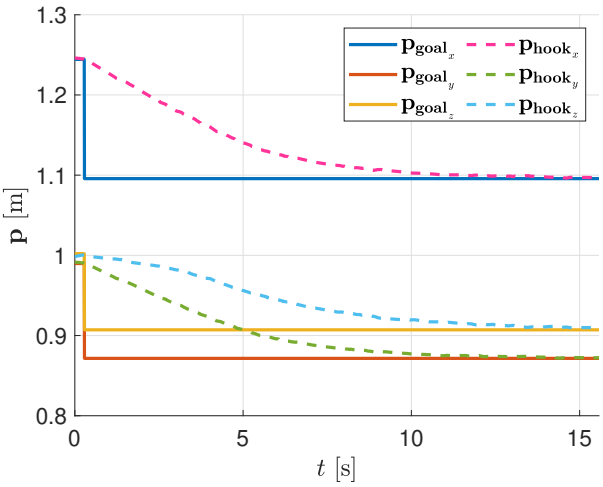
Fig. 6 Histograms for the end effector position errors.

implementation of the algorithm, both with and without the saturation scheme engaged. The following subsection addresses the key takeaways of these results.

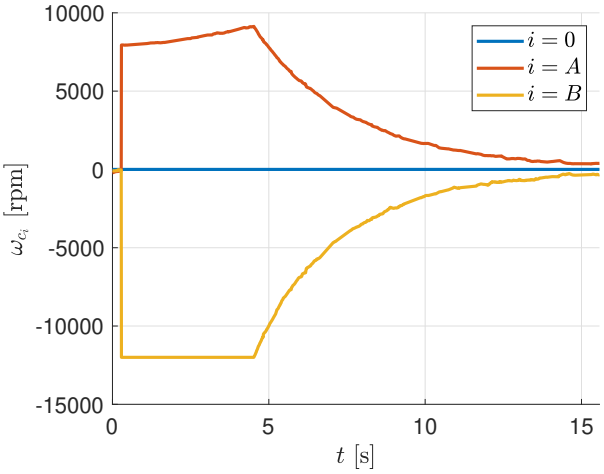
D. Discussion

The selection of the payload placement and payload drop-off locations were not systematic, which may have infused biases in the results presented. The locations chosen by the LSMS operator did not span the entire workspace of the LSMS. However, the expended level of effort was deemed adequate by the team for a preliminary study on the performance of the Cartesian move algorithm.

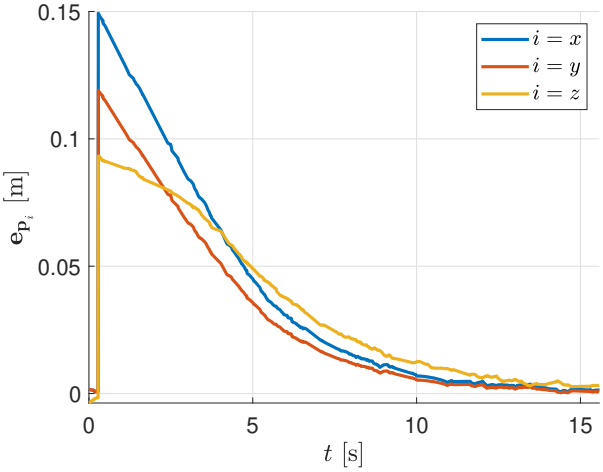
The results presented show that the Cartesian move algorithm is an accurate mechanism to place the end effector at a desired location, specified in Cartesian coordinates. This claim is limited to scenarios where end effector is placed relatively close to the goal location, since the maximum initial distance between the end effector and the goal location across all data records is 0.41 m. More importantly, the algorithm is designed for proximity operations, since it does not provide control over the path of the end effector. For applications that require precise control on the path of the end effector, a more sophisticated algorithm, such as Cartesian trajectory tracking or path-following algorithms, would be required. If such an algorithm is developed, the capture phase within the pick-up task and the retrieve phase within the drop-off task may benefit from an additional level of control along the path. This is likely to reduce the number of events where the end effector grazes the top of the payloads, which can sometimes occur due to the motion of the end effector; shown in Fig. 7f.



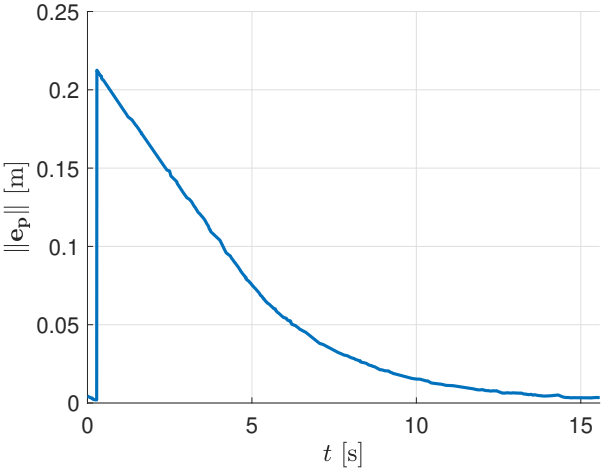
(a) Hook and goal positions.



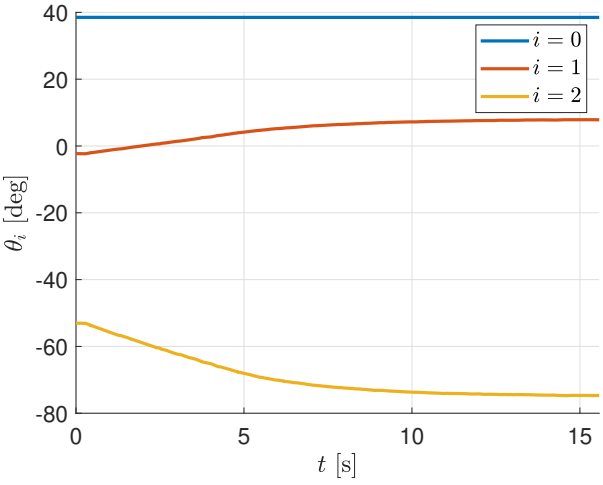
(b) Motor speed commands.



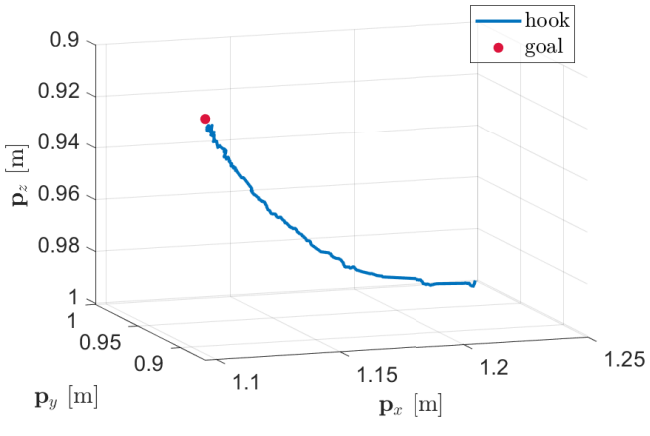
(c) Position error.



(d) Norm of the position error.

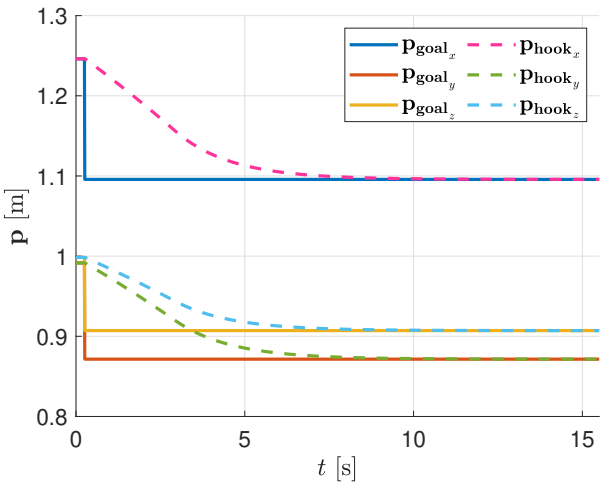


(e) Joint angles.

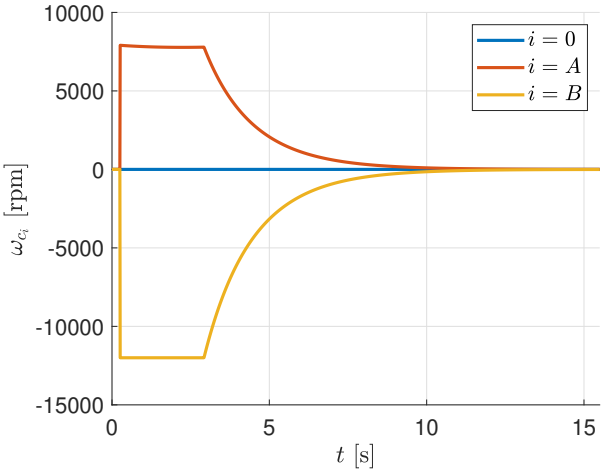


(f) Hook and goal positions in the 3D workspace.

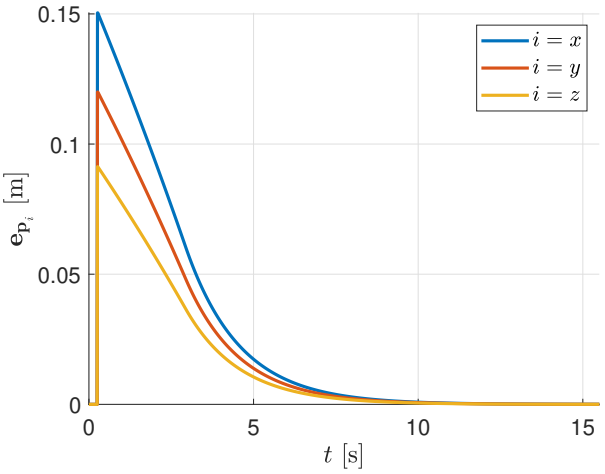
Fig. 7 Hardware results in data record 6, a retreat phase within a drop-off task.



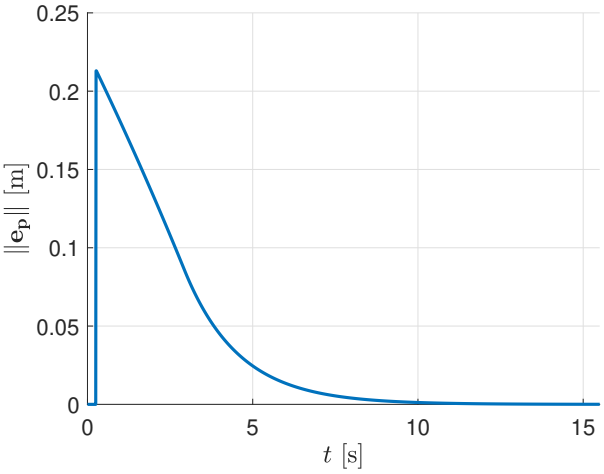
(a) Hook and goal positions.



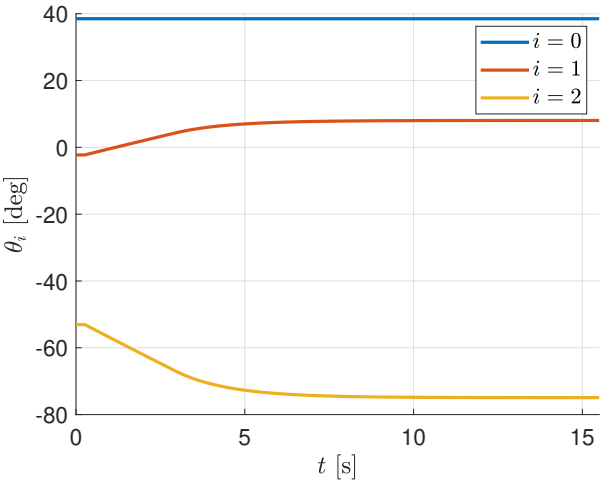
(b) Motor speed commands.



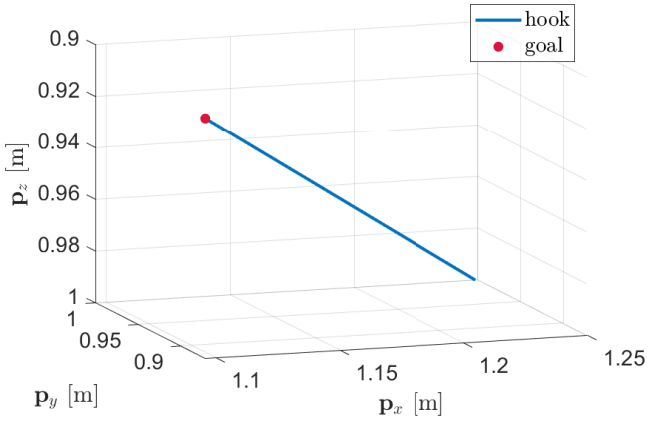
(c) Position error.



(d) Norm of the position error.



(e) Joint angles.



(f) Hook and goal positions in the 3D workspace.

Fig. 8 Simulation with the initial and final conditions in data record 6, a retreat phase within a drop-off task.

VI. Concluding Remarks

This paper provides a brief overview of the LSMS family of cable-driven cranes designed at NASA Langley Research center for surface operations on the Moon and Mars. A short description of a Cartesian move algorithm, designed specifically for the LSMS, follows. The algorithm generates motor speed commands to move the end effector into a reachable goal location, specified in Cartesian coordinates.

The algorithm is a key technology to perform autonomous and teleoperated payload manipulation tasks, as demonstrated by the preliminary assessment of the performance of the Cartesian move algorithm. The study includes a set of ten tests with payload pick-up and drop-off tasks where the Cartesian move algorithm is leveraged to perform different parts of the tasks. In total, 49 different instances were recorded.

The algorithm met the desired outcomes in 46 of the records, and minimum outcomes in 3 of the records. The Cartesian move algorithm is proving to be an adequate approach for motion control tasks that specify a goal position for the end effector with little or no requirements on the path it takes to arrive at the goal position. It can reliably place the end effector with accuracy at the goal location. Applications that impose requirements on the path that the end effector must take to arrive at the goal position should employ Cartesian trajectory tracking or path-following algorithms. These algorithms are not yet available for the LSMS and will need to be developed in future work.

Appendix

Table 2 End effector position data for all data records expressed with respect to the ground frame.

Record	Initial position $p_{hook}(t_{ini})$ [m]			Final position $p_{hook}(t_{end})$ [m]			Desired position p_{goal} [m]			Final position error $e_p(t_{end})$ [m]		
	x	y	z	x	y	z	x	y	z	x	y	z
1	1.4755	0.2547	0.1604	1.7702	0.3178	0.2385	1.7673	0.3186	0.2353	0.0029	-0.0009	0.0032
2	1.7856	0.2940	0.1334	1.7870	0.3250	0.0492	1.7868	0.3254	0.0479	0.0002	-0.0004	0.0013
3	1.7688	0.3169	0.2246	1.7766	0.3006	0.0611	1.7770	0.2998	0.0597	-0.0004	0.0008	0.0014
4	1.2536	0.9973	0.5850	1.2513	0.9966	0.9882	1.2522	0.9960	0.9871	-0.0008	0.0006	0.0011
5	1.2526	0.9973	0.9855	1.2461	0.9915	0.9983	1.2443	0.9898	1.0021	0.0017	0.0017	-0.0038
6	1.2461	0.9915	0.9984	1.0970	0.8723	0.9103	1.0957	0.8715	0.9071	0.0013	0.0008	0.0032
7	1.5284	-1.0712	0.2119	1.5286	-1.0686	0.0976	1.5298	-1.0669	0.0957	-0.0012	-0.0017	0.0019
8	1.5290	-1.0691	0.2343	1.5372	-1.0732	0.0912	1.5369	-1.0741	0.0908	0.0004	0.0009	0.0004
9	1.1146	-0.9896	0.5871	1.1145	-0.9860	0.9889	1.1142	-0.9892	0.9871	0.0003	0.0032	0.0018
10	1.1149	-0.9874	0.9887	1.1081	-0.9846	0.9998	1.1067	-0.9826	1.0021	0.0014	-0.0020	-0.0023
11	1.1081	-0.9846	0.9998	0.9666	-0.8587	0.9108	0.9647	-0.8565	0.9071	0.0020	-0.0023	0.0037
12	1.5366	-1.1270	0.2271	1.5258	-1.1456	0.1394	1.5260	-1.1452	0.1384	-0.0002	-0.0004	0.0010
13	1.5555	-1.1041	0.2197	1.5633	-1.1058	0.1209	1.5649	-1.1041	0.1191	-0.0015	-0.0017	0.0017
14	1.5490	-1.0974	0.2498	1.5576	-1.1037	0.1234	1.5568	-1.1055	0.1215	0.0007	0.0018	0.0019
15	1.5343	0.4594	0.6614	1.5342	0.4584	1.0142	1.5333	0.4571	1.0157	0.0009	0.0013	-0.0014
16	1.4641	0.4368	1.0148	1.3420	0.4002	0.9373	1.3417	0.4000	0.9357	0.0003	0.0002	0.0016
17	1.2922	-0.9708	0.2610	1.5541	-1.1105	0.2900	1.5494	-1.1126	0.2882	0.0047	0.0020	0.0018
18	1.3128	-0.9385	0.2235	1.5550	-1.1108	0.2856	1.5529	-1.1131	0.2857	0.0020	0.0022	-0.0001
19	1.5526	-1.1124	0.2818	1.5396	-1.1535	0.1612	1.5390	-1.1549	0.1597	0.0007	0.0014	0.0014
20	1.5550	-1.1108	0.2856	1.5599	-1.1171	0.1249	1.5589	-1.1190	0.1221	0.0011	0.0019	0.0028
21	1.5296	-0.6021	0.1538	1.8099	-0.6987	0.2450	1.8120	-0.6974	0.2430	-0.0021	-0.0013	0.0019
22	1.8109	-0.6999	0.2326	1.8093	-0.7167	0.0542	1.8088	-0.7179	0.0531	0.0005	0.0011	0.0012
23	1.5926	0.1762	0.5861	1.5905	0.1777	0.9872	1.5903	0.1756	0.9871	0.0001	0.0021	0.0001
24	1.5695	0.1725	0.9959	1.3941	0.1544	0.9085	1.3915	0.1536	0.9071	0.0025	0.0007	0.0014
25	1.3813	0.4461	0.1489	1.6532	0.5166	0.2390	1.6535	0.5165	0.2373	-0.0003	0.0001	0.0017
26	1.3694	0.4258	0.1683	1.6528	0.5149	0.2361	1.6522	0.5180	0.2370	0.0006	-0.0031	-0.0009
27	1.6532	0.5166	0.2390	1.6783	0.5395	0.0514	1.6784	0.5413	0.0504	-0.0001	-0.0018	0.0009
28	1.6527	0.5149	0.2365	1.6657	0.5159	0.0666	1.6659	0.5141	0.0666	-0.0002	0.0017	-0.0000
29	1.5914	-0.1757	0.5909	1.5929	-0.1775	0.9842	1.5903	-0.1756	0.9871	0.0026	-0.0019	-0.0029
30	1.5928	-0.1775	0.9843	1.5823	-0.1766	0.9992	1.5804	-0.1745	1.0021	0.0019	-0.0021	-0.0029
31	1.5813	-0.1764	1.0006	1.3929	-0.1529	0.9090	1.3915	-0.1536	0.9071	0.0013	0.0008	0.0019
32	1.4004	0.4590	0.1836	1.6545	0.5289	0.2694	1.6555	0.5249	0.2686	-0.0010	0.0040	0.0008
33	1.3642	0.4404	0.1954	1.6561	0.5290	0.2728	1.6560	0.5310	0.2696	0.0001	-0.0019	0.0033
34	1.6545	0.5289	0.2694	1.6891	0.5460	0.0809	1.6884	0.5485	0.0835	0.0006	-0.0025	-0.0026
35	1.6551	0.5284	0.2398	1.6478	0.5255	0.1003	1.6495	0.5234	0.0953	-0.0016	0.0020	0.0050
36	1.5817	-0.2568	0.6113	1.5811	-0.2575	1.0127	1.5794	-0.2558	1.0157	0.0016	-0.0017	-0.0029
37	1.5816	-0.2575	1.0129	1.5726	-0.2563	1.0273	1.5695	-0.2542	1.0307	0.0031	-0.0021	-0.0034
38	1.5726	-0.2563	1.0274	1.3835	-0.2246	0.9372	1.3820	-0.2238	0.9357	0.0015	-0.0008	0.0015
39	1.4109	0.4603	0.2019	1.6395	0.5245	0.2755	1.6408	0.5225	0.2757	-0.0013	0.0020	-0.0002
40	1.3682	0.4380	0.2025	1.6348	0.5242	0.2721	1.6368	0.5256	0.2728	-0.0021	-0.0014	-0.0006
41	1.6395	0.5245	0.2755	1.6909	0.5427	0.0987	1.6897	0.5447	0.0978	0.0011	-0.0020	0.0009
42	1.6348	0.5242	0.2720	1.6381	0.5204	0.0970	1.6385	0.5175	0.0983	-0.0004	0.0029	-0.0013
43	1.5223	-0.4944	0.6156	1.5233	-0.4969	1.0132	1.5217	-0.4944	1.0157	0.0016	-0.0025	-0.0025
44	1.5222	-0.4963	1.0147	1.5142	-0.4936	1.0273	1.5122	-0.4913	1.0307	0.0020	-0.0022	-0.0034
45	1.5141	-0.4935	1.0274	1.3326	-0.4334	0.9374	1.3315	-0.4326	0.9357	0.0011	-0.0008	0.0017
46	1.7464	-0.4192	0.2168	1.7451	-0.4132	0.1135	1.7446	-0.4150	0.1132	0.0005	0.0018	0.0003
47	1.5261	0.4517	0.5842	1.5263	0.4558	0.9845	1.5237	0.4542	0.9871	0.0026	0.0015	-0.0026
48	1.5258	0.4556	0.9850	1.5166	0.4524	0.9983	1.5142	0.4514	1.0021	0.0024	0.0010	-0.0038
49	1.5165	0.4523	0.9985	1.3339	0.3975	0.9093	1.3321	0.3971	0.9071	0.0018	0.0004	0.0022

Acknowledgments

The authors would like to thank Natalia Alexandrov for her constructive comments and corrections, and William Doggett for his advice on the intersection of structures, robotics, in-space assembly and manufacturing.

References

[1] Dorsey, J., Jones, T., Doggett, W., King, B., Mercer, C., Brady, J., Berry, F., Anderson, E., and Ganoe, G., "Recent Developments in the Design, Capabilities and Autonomous Operations of a Lightweight Surface Manipulation System and Test-bed," *AIAA SPACE Conference & Exposition*, Long Beach, CA, USA, 2011, pp. 1–20. https://doi.org/10.2514/6.2011-7266.

[2] Doggett, W., Dorsey, J., Collins, T., King, B., and Mikulas, M., "A Versatile Lifting Device for Lunar Surface Payload Handling, Inspection & Regolith Transport Operations," *American Institue of Physics Conference Proceedings*, Vol. 969, No. 1, 2008, pp. 792–808. https://doi.org/10.1063/1.2845045.

[3] Dorsey, J., Jones, T., Doggett, W., King, B., Mikulas, M., and Roithmayr, C., "Developments to Increase the Performance, Operational Versatility and Automation of a Lunar Surface Manipulation System," *AIAA SPACE Conference & Exposition*, Pasadena, CA, USA, 2009, pp. 1–15. https://doi.org/10.2514/6.2009-6795.

[4] Wong, I. M., and Cline, J. E., "LSMS–L35, Miniature Crane for Payload Off-loading and Manipulation: Development, and Application," *ASCEND*, Las Vegas, NV, USA, 2022, pp. 1–17. https://doi.org/10.2514/6.2022-4237.

[5] Jones, T. C., "A Protoflight Lightweight Surface Manipulation System to Enable High-Load, Long Reach Lunar Surface Operations," *ASCEND*, Las Vegas, NV, USA, 2021. https://doi.org/10.2514/6.2021-4167.

[6] Dorsey, J., Mikulas, M., and Doggett, W., "Preliminary Structural Design Considerations and Mass Efficiencies for Lunar Surface Manipulator Concepts," *AIAA SPACE Conference & Exposition*, San Diego, CA, USA, 2008, pp. 1–17. https://doi.org/10.2514/6.2008-7916.

[7] "Artemis Plan, NASA's Lunar Exploration Program Overview," Tech. rep., National Aeronautics and Space Administration, September 2020. URL https://www.nasa.gov/wp-content/uploads/2020/12/artemis_plan-20200921.pdf, visited on 12/04/2024.

[8] "NASA Strategic Plan 2022," Tech. rep., National Aeronautics and Space Administration, March 2022. URL https://www.nasa.gov/wp-content/uploads/2018/01/2022_nasa_strategic_plan_0.pdf, visited on 12/04/2024.

[9] Grande, M. L., Moses, R. W., Cosgrove, P. A., Mueller, R. P., Prater, T. J., and Blanchard, A. J., "Protecting Crew and Surface Systems with a Long-Duration Lunar Safe Haven," *ASCEND*, Las Vegas, NV, USA, 2021, pp. 1–14. https://doi.org/10.2514/6.2021-4070.

[10] Doggett, W., Roithmayr, C., Dorsey, J., Jones, T., King, B., Mikulas, M., Shen, H., and Seywald, H., "Automation of a Versatile Crane (the LSMS) for Lunar Outpost Construction, Maintenance and Inspection," *AIAA SPACE Conference & Exposition*, Pasadena, CA, USA, 2009. https://doi.org/10.2514/6.2009-6546.

[11] Puig-Navarro, J., Bisio, D. R., Pye, J. E., Granov, Y., Moser, J. N., Friz, J. S., Waltz, W. J., Cline, J. E., and Allen, B. D., "A Trajectory Tracking Algorithm for the LSMS Family of Cable-Driven Cranes," *Robotics: Science and Systems (RSS)*, Delft, The Netherlands, 2024, pp. 1–12.

[12] Puig-Navarro, J., Bisio, D. R., Scott, A. E., Cline, J. E., and Allen, B. D., "Safe Operational Envelope for the LSMS Family of Cable-Driven Cranes," *AIAA Aviation Forum and ASCEND*, Las Vegas, NV, USA, 2024, pp. 1–25.

[13] Lewis, F. L., Dawson, D. M., and Abdallah, C. T., *Robot Manipulator Control: Theory and Practice*, 2nd ed., Marcel Dekker, Inc., New York, NY, USA, 2003.