Introducing The Lunar Autonomous PNT System (LAPS) Simulator

Benjamin Hagenau
University of Colorado
1014 Adams Circle
Boulder CO 80303
beha7507@colorado.edu

Brian Peters
Ohio University
11373 Highland Park Rd.
Logan OH 43138
bp908214@ohio.edu

Roland Burton
Systems Engineer
HX5 LLC
NASA Ames Research Center
Moffett Field CA 94035
roland.burton@nasa.gov

Kelley Hashemi
Research Engineer
NASA Ames Research Center
Moffett Field CA 94035
kelley.e.hashemi@nasa.gov

Nicholas Cramer
DSA Project Manager
NASA Ames Research Center
Moffett Field CA 94035
nicholas.b.cramer@nasa.gov

Abstract— **In this paper we introduce a software simulator that has been used to develop an architecture for a low-cost Lunar Autonomous Position, Navigation and Time (PNT) System, LAPS. LAPS is a conceptual architecture for providing autonomous PNT services on and around the Moon using non-dedicated low-cost orbital and ground assets. The simulation tool has been developed to be flexible and is capable of modeling and analyzing the many different capabilities and configurations that the non-dedicated assets could support. The tool models the creation of an ad hoc swarm, the localization of this swarm and the subsequent provision of PNT services from this swarm. We present results from several studies of select configurations chosen to reflect existing and future real-world needs and capabilities.**

TABLE OF CONTENTS

1. INTRODUCTION

Over the next few decades there is expected to be a substantial increase in Lunar missions supporting and inspired by NASA's return to the moon. A large fraction of these missions will be low cost, utilizing ride-shares and CubeSat form factors including Lunar Flashlight[1], Lunar IceCube[2] and Luna H-Map[3].

These low cost missions will need navigation capabilities but will likely be unable to support the large power, mass and weight that existing navigation technologies entail, such as utilizing weak Global Navigation Satellite System (GNSS) signals or using the Deep Space Network (DSN). One option to provide a Lunar Position, Navigation and Time (PNT) service to these missions would be to create a dedicated Lunar GNSS constellation, similar to Earth GNSS systems. However, such a constellation would be expensive, with each member of the constellation carrying dedicated hardware, such as weak GNSS receivers and atomic clocks, necessary to provide precise navigation signals. Further, dedicated GNSS constellations are designed to provide a global 24/7 service that is under central control, and it is not clear that there will be enough lunar users to support the resources this would require. As an alternative, existing Lunar science and exploration assets could be used to create a low-cost, autonomous, ad-hoc and on demand mission-centric Lunar PNT swarm capable of providing PNT services to these low-cost lunar missions. This is the concept behind the Lunar Autonomous PNT System, LAPS.

The future concept of operations for LAPS is illustrated in Figure 1. In the first stage, a request is made by an *end user* to provide PNT services to a specific area on the lunar surface or orbit at a specific time. End users are assets that require PNT services but cannot help to provide them and may be ground based fixed or roving platforms, flyers or orbital vehicles. In the second stage the existing lunar assets that can usefully provide PNT services are selected and scheduled from the pool of existing non-dedicated and heterogeneous lunar science assets. Selection and scheduling is based on existing asset availability and trajectories as LAPS will neither control or maneuver the assets nor conflict with existing asset missions. The selected swarm needs to include at least one *anchor node* that may be ground or orbital otherwise the swarm localization algorithms will not converge. Anchor nodes have some ability to determine their own location independent of the LAPS system. In the third stage the selected assets localize themselves autonomously using mutual two-way crosslinks. This localization happens in a distributed manner whereby assets only localize with respect to their in-range neighbors. The distributed approach is a core feature of the LAPS architecture and fits with the provision of services by non-dedicated assets on an ad-hoc basis. In the fourth and final stage, the now localized swarm assets can provide PNT services by means of a broadcast signal to the requested areas or orbits.

By comparison, traditional Earth based GNSS systems do not include the first two stages and have significant differences in how stages three and four occur. The first two stages are not required in existing GNSS systems as PNT services are provided continuously from dedicated PNT assets, the

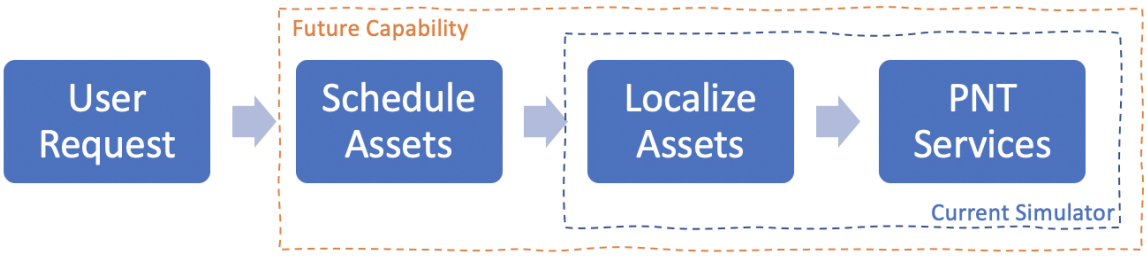


Figure 1: Concept of Operations of LAPS

GNSS constellation. In GNSS systems the third stage, asset localization, is done using ground-based radar precision orbit determination and is managed centrally from Earth-based operations centers. In comparison, LAPS uses decentralized localization carried out onboard the assets. The user localization algorithms used by GNSS in the final stage rely on the well defined orbits and homogeneity of the GNSS constellations. In comparison, LAPS user localization must account for the irregular orbits and heterogeneity of the swarm.

The existing LAPS Simulator models the third and fourth stages, and that is the focus of this paper. Eventually all three stages following a user request will be modeled.

The two most prevalent use cases for distributed swarm space systems are data collection and providing services. LAPS is part of the larger Distributed Spacecraft Autonomy (DSA) project[4], which includes demonstrations of both applications. A flight demonstration of distributed data collection on board the Starling1 mission[5] is scheduled for 2021, and the development of the LAPS Simulator will extend the flight software approaches used in the Starling1 demonstration to providing services. Further, the LAPS simulations will also demonstrate the ability of the chosen approaches to scale to large swarm sizes as part of the DSA Scalability Study project phase. The inclusion of LAPS within the DSA project means that the LAPS Simulator is more than just an academic exercise: the algorithms will become the basis for fight software that will be tested in a hardware-in-the-loop (HIL) simulation in late 2021.

This paper focuses on the algorithms that have been implemented in the LAPS Simulator so far and provides simulated results of these algorithms in test cases representative of the missions and environment that LAPS has been designed for.

2. Background

The state of the art designs for lunar positioning technology primarily utilize weak signal Global Positioning System (GPS)[6] or high earth orbit (HEO) GPS[7]. Simulation of these techniques has shown positioning errors of 100 m (3σ)[8], [9], [10] which will not meet many mission localization requirements without additional user Inertial Navigation System (INS) augmentation.

Current GPS infrastructure makes little use of automated PNT algorithms. However, automated positioning techniques for other multi-unit systems, such as robot swarms, have received more attention and often rely on a distributed, extended Kalman filter (DEKF) to coordinate all available sensor measurements across the group[11], [12]. The DEKF structure implies that each node runs its own local Kalman filter that estimates its own position from its own sensor measurements. Additionally, each local Kalman filter incorporates relative position information from neighboring nodes when in range. At least one node in the network must have access to an independent source of position knowledge to permit observability and bounded uncertainty. DEKF techniques for satellite groups with crosslink capabilities have also been proposed[13].

Automated time synchronization among distributed nodes has also received significant attention in the context of mobile sensor networks. Various Kalman filter-based algorithms have been developed for synchronization of local clocks to a reference clock[14]. High accuracy, high-efficiency implementations of such algorithms has been demonstrated on underwater mobile sensor networks which also estimate and exploit Doppler shift[15]. Hardware-centric solutions for time-keeping in space that utilize high accuracy clocks are also under parallel investigation[10], but this type of solution is not pursued in LAPS.

Other NASA missions have begun to investigate the feasibility of a distributed solution to lunar PNT service and develop necessary technologies. NASAs LunaNet architecture will facilitate data transfer among LunaNet users and could be used to enable navigation message exchange[16]. NASAs Cislunar Autonomous Positioning System Technology Operations and Navigation Experiment (CAPSTONE) mission[17], scheduled to launch in 2021, intends to verify the dynamics of a lunar near rectilinear halo orbit useful for long-term infrastructure such as PNT assets. It will also evaluate autonomous navigation software through crosslink communication with the Lunar Reconnaissance Orbiter (LRO)[18].

3. The LAPS Simulator

The LAPS simulator currently models the final two stages of the LAPS architecture as illustrated in Figure 1. These stages are the localization of the swarm assets that will provide PNT services, and the localization of end users. The simulator also produces quality of service metrics for the PNT services that the localized swarm provides. The simulator has been designed to be modular, allowing various localization techniques and algorithms to be developed and tested without significant changes to the framework. The simulation currently supports distributed EKFs with pluggable dynamics models for asset localization and weighted least squares for user localization. The LAPS simulator also supports the use of a centralized EKF (CEKF) for swarm localization. CEKFs are not part of the LAPS concept and are used by the simulator to provide a theoretical performance ceiling to compare DEKF

methods against. PNT quality of service modeling adopts standard metrics from the GNSS community, including computing dilution of precision[19] (DOP). Crucially, all asset capabilities, including swarm clock accuracy, independent location self-knowledge (e.g. from DSN or weak-GNSS) and pseudorange timing measurement precision can be set independently for each asset, reflecting the key concept of utilizing non-dedicated assets. The simulator is predominantly implemented in MATLAB[1]. GMAT[2], an open source astrodynamics software program is used to propagate swarm assets and generate ephemeris. Ephemeris is exchanged between GMAT and MATLAB in the the CCSDS-OEM[20] format, allowing for alternative propagators to be used.

To date, the LAPS simulator has been used to develop and refine algorithms. However, as the DSA project moves toward the scalability study phase the simulator modules will be used as prototype for flight software modules (specifically, cFS[3] Apps) that will be used in the hardware-in-the-loop simulators scheduled for FY22.

The modular and pluggable architecture of the LAPS simulator means that it would be straightforward to use the simulator for modeling autonomous distributed PNT systems in non-Lunar environments. Indeed, only the dynamics model, a pluggable function, would need to modified to achieve this.

4. ALGORITHMS

In this section the algorithms that have been developed for use in LAPS for swarm asset localization and end-user localization are discussed.

Asset Localization

Inline with the distributed nature of the LAPS swarm, state estimation of swarm assets is primarily performed using a distributed extended Kalman filter (DEKF) implemented locally on each individual swarm asset. That is, the collective state information of the full swarm is distributed over all assets with each asset estimating its own state using an on board extended Kalman filter and inter-satellite link (ISL) measurements made with mutually visible assets. A centralized EKF is also implemented and is used to provide a baseline to compare distributed algorithms against.

Anchor nodes—Anchor nodes are swarm assets with an independent onboard position capability, such as weak-GNSS, DSN ranging or physical anchoring to the lunar surface. Anchor nodes are required because the absolute inertial states of the swarm assets are not observable using only inter-satellite link (ISL) pseudorange measurements in a Kalman filter when no asset state is perfectly known. Indeed, the swarm asset state estimate error is unbounded, and the solution will diverge with time. For this reason, the swarm needs to incorporate anchor nodes. Theoretically, the state estimate error and covariance are bounded when only one anchor node exists but the estimate error can become large if insufficient anchor nodes are included.

Measurements— The developed DEKF uses swarm asset to swarm asset pseudorange measurements made via inter-satellite links (ISL) and swarm asset to ground node pseudorange measurements to estimate the states of each swarm asset. Within the simulator at each measurement epoch, the true range is computed between each asset and all other assets that have a line-of-sight with the selected asset. Similarly, the exact range between each swarm asset and all visible ground assets is made assuming a configurable elevation cut-off angle to ground nodes, nominally 10°. The cut-off angle accounts for local terrain that may occlude the line of sight between ground nodes and swarm assets. Additive White Gaussian Noise (AWGN) is added to each true range to simulate the pseudorange measurement. Equation (1) is used to compute the simulated measured pseudorange measurements where w is a random sample from zero mean Gaussian distribution.

$$\hat{\rho} = \rho_{\text{true}} + w \tag{1}$$

The AWGN used is expected to bound the error of the pseudorange measurements but does not necessarily have the same error profile. The use of AWGN does enable development of Kalman filter algorithms which assumes that the measurement error is AWGN and provides a baseline performance the DEKF.

In the LAPS system, it is assumed that the measured pseudorange is derived from a two-way ISL communication by averaging the measured travel times in each direction, which reduces the impact of relative clock bias between the two swarm assets to only relativistic terms which are of the order of tens of nanoseconds. This two-way synchronization is modeled within the LAPS simulator and the AWGN noise, w, is sized to account for this small residual bias. Jitter is not explicitly considered and is assumed below other measurement noise.

Pseudorange measurements between a swarm asset and an anchor node are modeled as having half the covariance as that used for the inter-swarm ISL measurements. This is done under the assumption that the anchor nodes have a known position and no clock error.

Filter State—Let the state being estimated by swarm asset i be defined as in Equation (2),

$${}^{i}x = \begin{bmatrix} {}^{i}r \\ {}^{i}v \end{bmatrix} = \begin{bmatrix} {}^{i}r \\ {}^{i}\dot{r} \end{bmatrix} \tag{2}$$

where ${}^{i}r$ and ${}^{i}v$ are the i th swarm asset's position and inertial velocity respectively. This contrasts with a centralized extended Kalman filter which estimates the states of all assets at once on a central processor. The DEKF used is implemented in a framework that simulates each asset performing estimation steps individually. At any given measurement epoch, each asset's latest state estimate is propagated to the current measurement time one at a time as though they perform this action individually. The local measurements for each asset at that epoch are used to update each asset one at a time in a similar fashion. Measurement collection and localized state estimation is performed as a Time Division Multiple Access System (TDMA) comprised of two frames: measurement and estimate. During the measurement frame each asset makes pseudorange measurements with all other visible assets and receives the current state estimate and state covariance of each visible asset. During the estimate frame, each asset estimates its own state using the information collected from the ISL made during the measurement frame and this process is cycled over the simulated time.

Filter Propagation—Each swarm asset propagates its own state using a 2-body propagator defined in a Lunar-centered

[1]https://www.mathworks.com/
[2]https://gmatcentral.org/
[3]https://cfs.gsfc.nasa.gov

inertial frame. This simple propagation model assumes that the Moon is a point mass and that the swarm asset only experiences lunar gravity force, ignoring all other perturbations. The single asset state vector dynamics for the 2-body problem is shown in Equation (3).

$$ {}^{i}\dot{x} = \begin{bmatrix} {}^{i}\dot{r} \\ {}^{i}\ddot{r} \end{bmatrix} = \begin{bmatrix} {}^{i}\dot{r} \\ -\frac{\mu}{\|{}^{i}r\|^3} {}^{i}r \end{bmatrix} \tag{3} $$

Where ${}^{i}\ddot{r}$ is the 2-body acceleration experienced by the i th orbital asset and μ is the Moon's gravitational parameter. At this point the i notation will be dropped and it is assumed that each state vector and state covariance is that of a single asset. With $(\,)^+$ and $(\,)^-$ denoting post-measurement update and pre-measurement update respectively, and $(\,)_k$ denoting the k th iteration of the filter, the estimate propagation step is iterated by Equation (4)

$$ x_{k+1}^{-} = \int_{t_k}^{t_{k+1}} \dot{x}\, dt \tag{4} $$

with x_k^+ as the initial condition. Within the simulation, integration is performed by MATLAB's *ode113()* function which employs a variable-step, variable-order Adams-Bashforth-Moulton solver of orders one to thirteen and is effective in numerically integrating non-stiff differential equations.

The Jacobian matrix that describes 2-body dynamics, denoted A, is defined as the derivative of the state dynamics with respect to the state and has an analytical solution as described in Equation (5).

$$ A = \frac{d\dot{x}}{dx} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ -\frac{\mu I_{3\times3}}{\|r\|^3} + \frac{3\mu r r^T}{\|r\|^5} & 0_{3\times3} \end{bmatrix} \tag{5} $$

The Jacobian matrix is used to compute the state transition matrix, Φ_k. Let the state transition matrix be linearized about some nominal trajectory and be defined such that some deviation from that nominal trajectory at epoch k can be propagated to time $k+1$ by $\delta x_{k+1} = \Phi_k \delta x_k$ to a first order. In the case of LAPS, the nominal trajectory is the estimated trajectory over the propagation time and therefore Φ_k can be used to propagate the state covariance between measurement epochs. Because Φ_k propagates state deviations from k to $k+1$, it is computed by propagating the six-by-six identity matrix from time k to $k+1$ using the differential equation, Equation (6), which is derived in Ref. [21] and has solution given in Equation (7), where Δt is the time between k and $k+1$.

$$ \dot{\Phi}_k = A\Phi_k \tag{6} $$

$$ \Phi_k = \exp\left(A\Delta t\right) \tag{7} $$

The state transition matrix is then used to propagate the state estimate covariance using Equation (8),

$$ P_{k+1}^{-} = \Phi_k {P_k}^{+} \Phi_k^T + Q \tag{8} $$

where Q is the analytical solution for the discrete time process noise covariance matrix for a Newtonian system. A rigorous derivation of the Q can be found in Ref. [21] and is not derived in detail here but is reproduced in Equations (9) to (12). It is important to note that this expression for Q assumes zero-order hold over the integration period and that the unmodeled accelerations take the form of AWGN. The zero-order hold assumption is valid for propagation times under about 1/100 th of the characteristic period for the system which in the test cases studies in this paper is the orbital period.

$$ Q_{11} = \begin{bmatrix} \frac{\Delta t^4}{3}\sigma_Q^2 & 0 & 0 \\ 0 & \frac{\Delta t^4}{3}\sigma_Q^2 & 0 \\ 0 & 0 & \frac{\Delta t^4}{3}\sigma_Q^2 \end{bmatrix} \tag{9} $$

$$ Q_{12} = Q_{21} = \begin{bmatrix} \frac{\Delta t^3}{2}\sigma_Q^2 & 0 & 0 \\ 0 & \frac{\Delta t^3}{2}\sigma_Q^2 & 0 \\ 0 & 0 & \frac{\Delta t^3}{2}\sigma_Q^2 \end{bmatrix} \tag{10} $$

$$ Q_{22} = \begin{bmatrix} \Delta t^2\sigma_Q^2 & 0 & 0 \\ 0 & \Delta t^2\sigma_Q^2 & 0 \\ 0 & 0 & \Delta t^2\sigma_Q^2 \end{bmatrix} \tag{11} $$

$$ Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \tag{12} $$

In Equations (9) to (12), Δt is the propagation time and σ_Q is the standard deviation describing the process noise of the unmodeled accelerations. This is a State Noise Compensation (SNC) model which implies that the process noise is modeled as is zero-mean, uncorrelated, Gaussian noise.

Filter Measurement Update— The increased measurement covariance matrix method is used in performing the measurement update in the implemented DEKF. This method is useful in keeping each asset responsible for only estimating its own state but also incorporating all the uncertainty that goes into the predicted measurement. This measurement update step used in the DEKF implementation is leveraged from Ref. [13].

The nonlinear measurement function used to predict the pseudorange is defined in Equation (13) where asset i is predicting the range between itself and asset j to update its own state. Recall that during the measurement frame, each asset relays its current state estimate and estimate covariance with surrounding swam assets that are in view.

$$ {}^{ij}\hat{\rho} = h\left({}^{i}r, {}^{j}r\right) = \sqrt{({}^{i}r - {}^{j}r)^T({}^{i}r - {}^{j}r)} \tag{13} $$

In Equation (13), ${}^{i}r$ and ${}^{j}r$ are the estimated position coordinates of the i th and j th swarm assets respectively. The measurement sensitivity matrix, H, used to map deviations in the state to deviations in measurement is defined such that $\delta\rho = H\delta x$ to a first order. The measurement sensitivity matrix for a measurement made between the i th and j th swarm assets being used to update the i th swarm asset's state is stated in Equation (14)

$$ {}^{i}H = \frac{\delta h\left({}^{i}r, {}^{j}r\right)}{\delta {}^{i}x} = \frac{\left({}^{i}r - {}^{j}r\right)}{{}^{ij}\hat{\rho}} \tag{14} $$

Note that the measurement sensitivity matrix for the j th swarm asset is the negated measurement sensitivity matrix of the i th swarm asset, as described in Equation (15)

$$ {}^{j}H = -{}^{i}H \tag{15} $$

As ${}^{i}H$ is defined as the derivative of h with respect the i th swarm asset's state, there is assumed to be no uncertainty mapped into the predicted measurement from the position of

the j th swarm asset. In reality, this is not the case because the j th swarm asset's position is not perfectly known and is also being estimated. To accommodate this uncertainty without having the i th swarm asset estimate the state of the j th swarm asset as well as its own, which would decrease the scalability of the DEKF by increasing the computational load of each swarm asset, the measurement covariance is inflated by the state estimate covariance matrix of the j th asset. The augmented measurement covariance matrix used by the i th asset is stated in Equation (16)

$$ {}^{i}R_{\text{full}} = R_s + {}^{j}H^{j}P^{j}H^{T} \tag{16} $$

where R_s is the measurement covariance matrix and ${}^{i}R_{\text{full}}$ is the augmented measurement covariance used by the i th swarm asset to update its own state. The augmented measurement covariance matrix accounts for the uncertainties in the measurement caused by pseudorange error such as clock bias and latency as well as the uncertainty in the measurement that comes from the fact that the position of the other asset is not perfectly known. Equation (16) implies that the i th swarm asset requires knowledge of the j th swarm asset's covariance estimate. It is assumed that this information is exchanged during the two-way pseudorange measurement, along with the asset's states.

User Localization

The user localization algorithms used in LAPS are heavily influenced by those used in existing terrestrial GNSS systems[19] and are based on the least-squares approach. However, whereas in GNSS systems all GNSS assets providing service are assumed to have the same self-knowledge, in LAPS each swarm asset will have varying degrees of certainty of self-knowledge as expressed in the extended Kalman filter's covariance matrix. LAPS incorporates this heterogeneity by including a weighting matrix in the least-squares computation that is derived from the geometry and swarm asset covariance matrix. As such, in addition to the state estimates it is assumed that the navigation message also includes each asset's state covariance matrix at each epoch.

Measurements—At each epoch, pseudorange measurements are made from the user's true position to each visible PNT asset. Simulated error in this measurement manifests as a bias, t_k (in meters), added to the true ranges between the user and each asset, determined by the user receiver clock error τ_k (in seconds), multiplied by the speed of light, c_0, as in Equation (17).

$$ t_k = (\tau_k + N)\, c_0 \tag{17} $$

The range bias t_k is composed from the slow moving and estimated user clock bias τ_k (typically on the order of 10 milliseconds) and a smaller magnitude noise component, N (typically on the order of 10 nanoseconds). The simulated pseudorange measurement is then given by Equation (18),

$$ \hat{\rho}_k = \rho_{\text{true}} + t_k \tag{18} $$

where ρ_s^k is the simulated user pseudorange measurement, and ρ_{true} is the true range between the user and a swarm asset.

Weighting Matrix—The weight assigned to an asset in the least squares regression is determined by the projection of the asset's state covariance matrix, P, on to the unit vector from the asset to the user's estimated position, x, as show in Eq. (19).

$$ w_i = \frac{1}{x^T P_i x} \tag{19} $$

where x is the asset-to-user unit vector, and P is the asset's position state covariance matrix. The weights for each asset are then used to form the weight matrix W as shown in Eq. (20).

$$ W = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_n \end{bmatrix} \tag{20} $$

Herein lies an important distinction from common GNSS methodology, where it is assumed that the information provided to the user regarding the state of the PNT asset is highly accurate, meaning GNSS least squares weighting models are typically built on knowledge of the error present in the user's measurements caused by atmospheric effects, signal noise, etc. For LAPS, it is assumed that the user will have access to the swarm assets' state covariances, which can be employed to improve the localization performance by decreasing an asset's contribution to the solution if its state is known less precisely. The reliance on covariance information means that the navigation message broadcast by swarm assets to users also needs to include this information, in addition to the usual swarm asset position information. The weight matrix is applied in the least squares regression as shown in Eq. (21).

$$ \delta x_k = \left(A^T W A\right)^{-1} A^T W \delta\rho_s \tag{21} $$

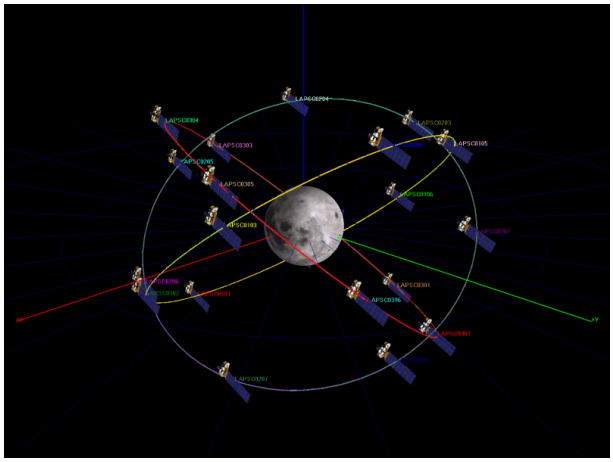
where δx_k is the change in the estimated user state vector, A is the system design matrix, and $\delta\rho_s$ is the vector of measurement residuals. Note that as the unit vector x is dependent on the user's estimated location and the weight matrix is dependent on x, the weight matrix estimates must be updated in each iteration of the least squares problem and converge along with the user's position estimate.

5. TEST CASES

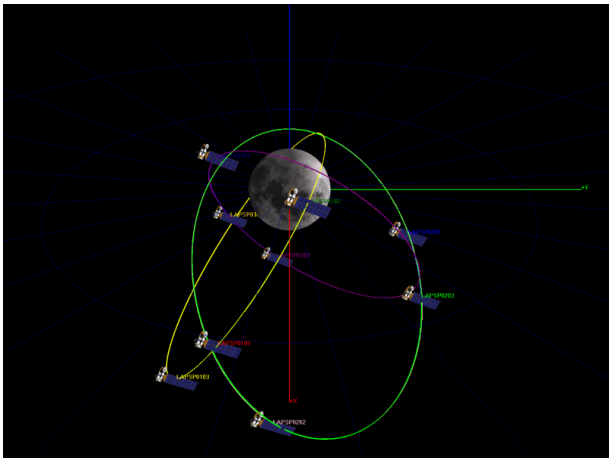
To test the performance of the algorithms and simulator, results from two canonical test cases are presented in this paper. Test cases One and Two use frozen constellations that provide low-latitude[22] and polar[23] coverage respectively, and are detailed in Table 1. Frozen orbits were chosen to be representative of future low cost missions with limited fuel budgets. Test Case One, the large low-latitude constellation, is designed to represent a swarm that could provide global coverage if all assets were available and can provide upper bounds to service availability. In contrast, Test Case Two, the polar coverage constellation, represents a smaller swarm that is still able to provide continuous PNT service to the lunar south pole. Such a constellation could support missions such as VIPER[24]. The two test case constellations are illustrated in Figure 2.

Table 1: Test Cases

Metric	Test Case One	Test Case Two
Coverage Area	Low Latitudes	South Pole
No. of Planes	3	3
No. of Assets	21	9
Inclination (deg)	39.71	56.2
Eccentricity	0.001	0.6
Semi-Major-Axis (km)	7298.6	6541.4



(a) Test Case One: Low Latitude



(b) Test Case Two: South Polar

Figure 2: Illustration Test Case Swarm Orbits

6. RESULTS

Results from running the asset localization and user localization algorithms for the two test cases are described in this section.

Asset Localization

This first results section describes the performance of the distributed extended Kalman filter and provides a comparison to the CEKF that demonstrates the benefits and draw backs of utilizing a DEKF.

DEKF Results: Test Case One—The DEKF implemented in this section uses the 2-body propagation and increased measurement covariance update steps described in Section 4. The variance of the inter-satellite pseudoranges was assumed to be $10\,\mathrm{m}^2$ and pseudoranges were measured every $100\,\mathrm{s}$. Twenty-two evenly distributed ground nodes were used as anchor nodes.

Figure 3 displays the position estimate error for the seventh asset in the third plane in that asset's orbit frame. The x-axis is in the orbit radial direction, the y-axis is in the along-track direction, and the z-axis is in the orbit normal direction. These results are representative of each asset.

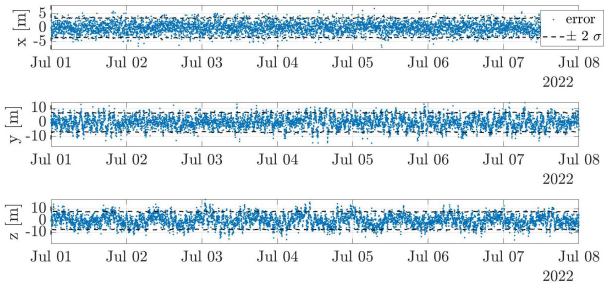


Figure 3: Position error and covariance estimates for one swarm asset (Test Case One)

Insight can be drawn from plotting errors in the swarm asset's local orbital frame. The radial direction, the x-axis, has the lowest error of the three directions because it is dominantly updated by measurements made between the swarm asset and lunar ground nodes which have reduced error compared to ISL measurements. The along-track direction, the y-axis, has a more random, and slightly lower error than that of the z-axis because the along track direction position is consistently being updated by the assets ahead and behind it in the same plane. The orbital normal direction, the z-axis, is updated dominantly by swarm assets in other planes and even then these measurements are not as directly in the orbital normal direction than as the other measurements are in their respective directions because the planes all have an inclination of $40\,^\circ$.

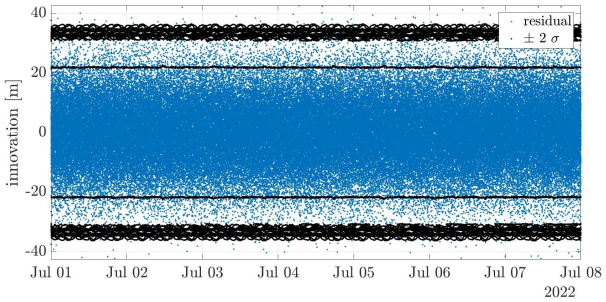


Figure 4: Measurement residuals (innovation) for one swarm asset (Test Case One)

The measurement residual, or innovation, for the same asset is shown in Figure 4 for every measurement made by that asset. In Figure 4 the residuals and their $2\,\sigma$ bounds are shown superimposed over the seven-day period to illustrate the performance of the filter with all measurements for a single asset. The innovation is defined as the difference in the predicted measurement computed after the propagation step and before the measurement update step and the actual measurement for that epoch. The mean innovation for this asset is 7.20×10^{-5} m and appears to approach zero as more measurements are considered in the average. The inner covariance bounds correspond to measurements made between this representative asset and the ground nodes. While the outer $2\,\sigma$ bounds correspond to the measurements made via ISL.

DEKF Results: Case Two—Figures 5 and 6 show the position error and covariance estimates and the measurement residuals

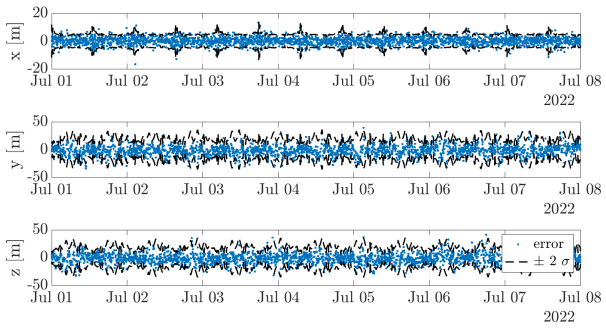


Figure 5: Position error and covariance estimates for one swarm asset (Test Case Two)

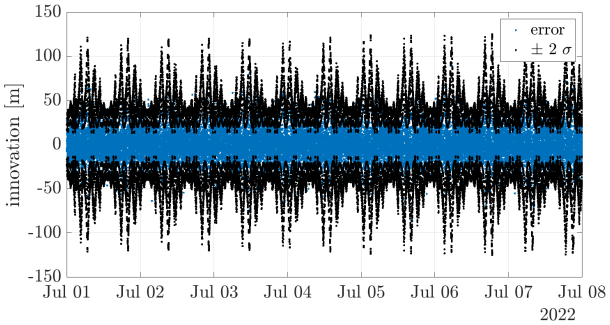


Figure 6: Measurement residuals (innovation) for one swarm asset (Test Case Two)

for Test Case Two. The results for Test Case Two differ in two key ways: the overall position errors and estimated variances are higher; and there is a cyclical pattern to the both estimated variances. The larger magnitude or filter errors, covariance and residuals are expected given the much reduced number of assets in the swarm. The cyclical patterns seen in measurement residuals are coincident with the orbit anomaly and reflective of the highly elliptic orbits used in Test Case Two.

DEKF-CEKF Comparison—The CEKF algorithm is implemented in the LAPS simulator to baseline the performance of DEKF approaches. The CEKF estimates the states of all assets simultaneously on a central processor using all measurements at a given epoch while the DEKF has each orbital asset estimate its own state using local measurements based on available ISLs.

Figure 7 shows the position estimate accuracy performance difference between the CEKF and the DEKF for a representative swarm asset. The left column in Figure 7 displays the root-mean-square (RMS) position estimate error for the CEKF and the left-hand column shows that of the DEKF. The top row contains results simulated using the scenario described previously in this section and the bottom row is the same scenario modified to have eight lunar ground nodes and only simulated over a day. The dashed lines are the mean estimate error while the blue points are RMS error at each epoch.

Table 2 summarizes the position errors for the cases with eight and twenty anchor nodes. As can seen, the performance of the DEKF is much more sensitive to the number of anchor nodes than the CEKF is. It is also important to note the shape of the error profiles. Despite the number of ground nodes, the results for the CEKF have a more desirable shape because they are consistently zero mean and randomly distributed. The DEKF has a similar form with 22 ground nodes but as the number of ground nodes is decreased to 8, the shape of the error becomes less random and therefore not as desirable.

Table 2: CEKF and DEKF Performance Dependence on Number of Anchor Nodes

Number of Anchor Nodes	**22**	**8**
CEKF Mean Error (m)	2.7	3.6
CEKF Max. Error (m)	12.3	14.1
DEKF Mean Error (m)	3.2	6.7
DEKF Max. Error (m)	13.2	25.3

In summary, CEKF performance can be closely matched by the DEKF by increasing the number of anchor nodes modeled. However, even with 22 anchor nodes, the CEKF error is still lower than that of the DEKF. As the number of anchor nodes decreases, the performance of the DEKF degrades significantly while that of the CEKF only decreases by a small margin. In addition to mean errors, the form of the DEKF error becomes significantly less desirable as the number of anchor nodes is decreased. Recall that the anchor nodes are assumed to have no error in their positions for this demonstration so the DEKF's dependence on the anchor nodes indicates that as a higher fidelity model is simulated, introducing error in the ground node positions, the CEKF's performance margin over the DEKF will likely increase. However, the performance of the DEKF is still close to that of the CEKF and is a viable option for replacing the CEKF. More tuning to adjust for differing numbers of anchor nodes should be performed before drawing more conclusions about the DEKF's performance.

Figure 8 compares the computation time to perform the propagation and measurement update steps for the CEKF and DEKF filters. To best reflect onboard computational burdens, in the case of the CEKF, the mean time for a single propagation step and a single measurement update step is shown. For the DEKF, the mean propagation and measurement update step time for a single asset to estimate its own state is averaged over all assets, over all epochs.

As expected, as the number of assets increases in each plane, the CEKF computation time required increases significantly for both the measurement and propagation steps. However, the time required by the DEKF remains nearly the same and quickly becomes orders of magnitude smaller than the CEKF computation time as the constellation grows. The DEKF not only greatly increases the scalability of the LAPS system by dispersing the computational load required to estimate the states of the orbital assets throughout the constellation, it is also operationally more practical than having all assets report their inertial measurements to a single processor which then updates those assets after performing estimation. The observed results imply that the DEKF facilitates a significantly more scalable system than the CEKF for only a minor loss in estimate accuracy depending on the number of anchor nodes modeled.

User Localization

In order to provide a broad view of the localization performance of a given LAPS configuration, a grid of 200 user positions was defined over the lunar surface as shown in Figure 9. In these performance studies it is assumed that all swarm assets are providing service all the time.

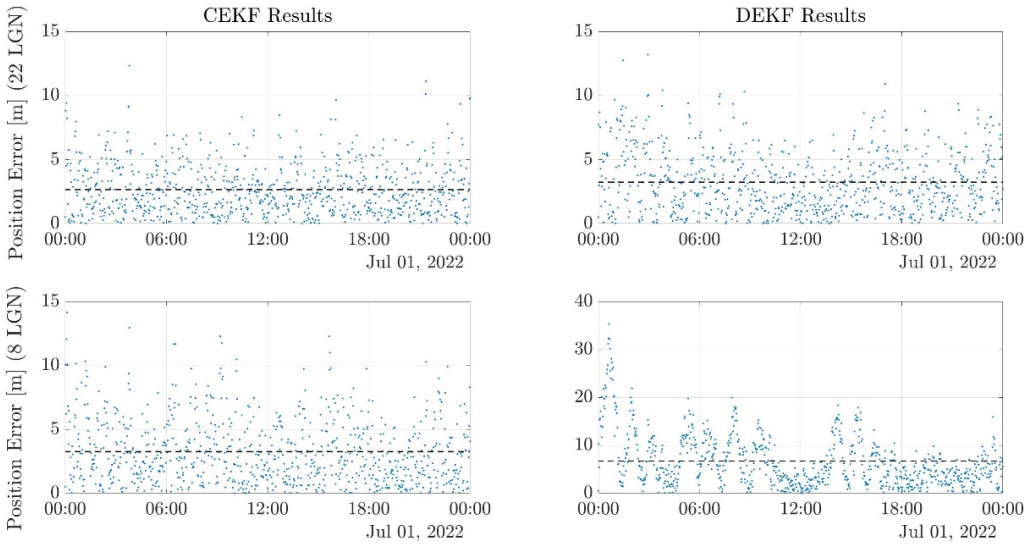


Figure 7: DEKF, CEKF Estimate RMS Error Comparison

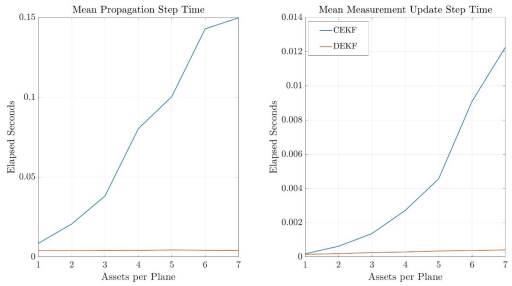


Figure 8: Comparison of Computational Time between DEKF and CEKF for Propagation and Measurement Update Steps

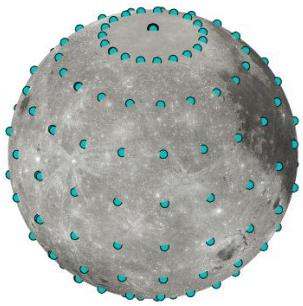


Figure 9: Simulated User Positions on Lunar Surface

The LAPS simulation iterates through each of these sample positions and estimates the user's position at each time step. Heat maps are produced which span the lunar surface and color each location according to the median of the magnitude of the position estimate error over the entire runtime. From these heat maps, inefficiencies in the LAPS constellations can be identified, and locations with poor localization performance can be evaluated in-depth.

Case One—The results in this subsection are produced using the circular low inclination swarm constellation. User clock error was set to 1.5 milliseconds plus a noise component with a standard deviation of 1 nanosecond. At each sampled location, a visibility cutoff of 15 ° was imposed to mask any low-elevation satellites that might be obstructed by terrain, or in a real implementation might introduce significant multipath error. The results from the localized swarm simulations as presented in the previous subsection were used to provide swarm asset covariances.

Figure 10 shows the heatmap of median position error estimates for a twenty eight day simulation. This simulation time period was selected to ensure full geometric diversity over a full lunar day.

As expected, the performance of the low latitude swarm constellation degrades as the user's latitude approaches the polar regions and the asset visibility becomes limited, but this configuration is still able to provide adequate performance over most of the lunar surface. Some particular epochs can be seen to exhibit particularly high position error, but these spikes in error are typically not persistent and occur at epochs where the dilution of precision (DOP) is high due to poor geometry of visible assets.

In addition to overall heat maps, the LAPS simulation time histories at individual locations can be generated, such as shown in Figures 11 and 12.

Case Two—Figure 13 shows the 28-day heatmap showing median position error for latitudes below 50 ° S (above this latitude the polar swarm constellation is not able to provide the minimum four satellite visibility for consistent service). In this region, typical median position errors are between 10 m and 15 m. Figure 14 shows a time history for a user at the Lunar south pole. The frequent time spent with only four satellites in view shows that Case Two is close to a minimum

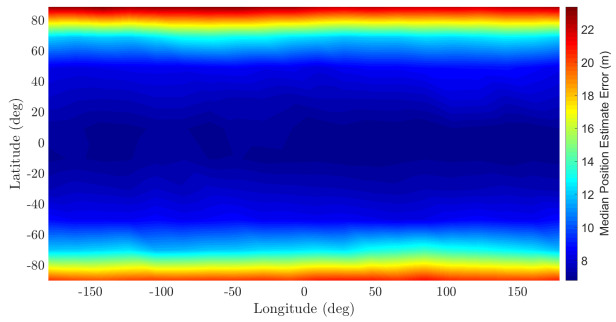


Figure 10: Heat Map of User Position Estimate Error Over 28-day Period (Lunar Day)

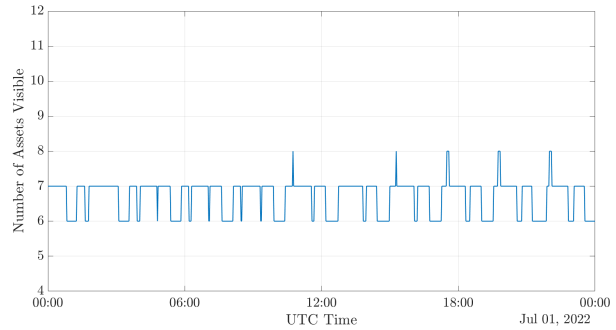


Figure 11: Time History of Number of Assets in View For User at 20 ° N, 90 ° W

viable swarm size to provide consistent service at a pole.

7. Conclusions

In this paper a simulator that is being used to develop the LAPS concept has been presented. Early results from simulator test cases have already provided key insights into how PNT performance is influenced by swarm configuration, including swarm orbits and the number of anchor nodes utilized. Performance with a small swarm constellation suitable for south polar coverage has been studied, and with reasonable assumptions about hardware capabilities, a localization performance of around 10 m has been predicted.

Over the next several months the capabilities of the LAPS simulator will be expanded to encompass all phases of the LAPS concept of operations, including modeling the distributed resource scheduling stage and adding swarm networking models. Eventually, the LAPS simulator will provide the basis for a hardware in the loop demonstration of a Lunar PNT swarm as part of the DSA Project.

In the future it is envisioned that the LAPS simulator will be used to model PNT systems in other regions, such as Mars or even in Earth orbit.

Acknowledgments

Funding for the work presented in this paper has come from the Distributed Spacecraft Autonomy (DSA) Project, part of the NASA Game on Development Program, the ISRDS-3 contract and the Ames Center Innovation Fund.

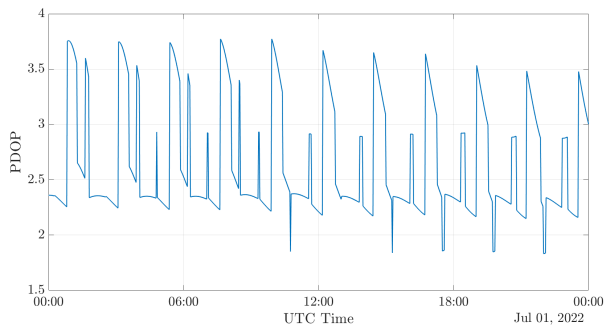


Figure 12: Time History of PDOP of Assets in View For User at 20 ° N, 90 ° W

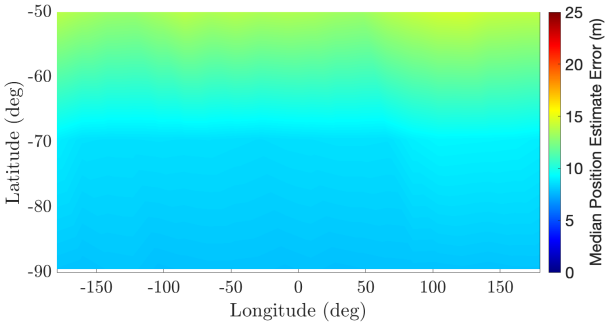


Figure 13: Southern Latitude Heat Map of User Position Estimate Error over 28-day Period (Lunar Day)

References

[1] B. A. Cohen, P. O. Hayne, B. T. Greenhagen, and D. A. Paige, "Lunar flashlight: Exploration and science at the moon with a 6u cubesat," *AGUFM*, vol. 2015, pp. EP52B–07, 2015.

[2] P. E. Clark, B. Malphrus, K. Brown, T. Hurford, C. Brambora, R. MacDowall, D. Folta, M. Tsay, C. Brandon, and L. I. C. Team, "Lunar ice cube: Searching for lunar volatiles with a lunar cubesat orbiter," in *DPS*, 2016, pp. 223–03.

[3] H. Kerner, C. Hardgrove, J. Bell, R. Amzler, A. Babuscia, M. Beasley, Z. Burnham, and K.-M. Cheung, "The lunar polar hydrogen mapper (lunah-map) cubesat mission," in *2016 Small Satellite Conference*. Utah State University, 2016.

[4] D. Cellucci, N. Cramer, and J. Frank, "Distributed spacecraft autonomy," in *2020 AIAA Ascend Conference*. AIAA, 2020.

[5] H. Sanchez, D. McIntosh, H. Cannon, C. Pires, J. Sullivan, S. D'Amico, and B. O'Connor, "Starling1: Swarm technology demonstration," in *2018 Small Satellite Conference*. Utah State University, 2018.

[6] P. A. Stadter, D. J. Duven, B. L. Kantsiper, P. J. Sharer, E. J. Finnegan, and G. L. Weaver, "A weak-signal gps architecture for lunar navigation and communication systems," in *2008 IEEE Aerospace Conference*, 2008, pp. 1–11.

[7] B. Ashman, F. H. Bauer, J. Parker, and J. Donaldson, *GPS Operations in High Earth Orbit: Recent Experiences and Future Opportunities*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2018-2568

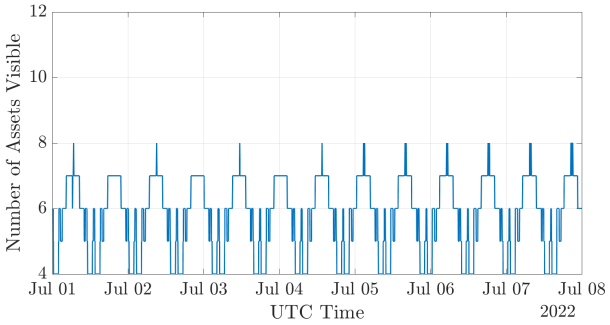


Figure 14: Time History of Number of Assets in View For User at 89.5 ° S, 179.5 ° W (Lunar South Pole)

[8] M. Tian, M. Wang, P. Wang, P. Shi, V. Capuano, J. Leclère, C. Botteron, P. Pierre-André, and P.-A. Farine, "Cross-band aided acquisition on heo orbit," 10 2014.

[9] V. Capuano, C. Botteron, J. Leclère, J. Tian, Y. Wang, and P.-A. Farine, "Feasibility study of gnss as navigation system to reach the moon," *Acta Astronautica*, vol. 116, pp. 186 – 201, 2015.

[10] Z. Wang, "Review of chip-scale atomic clocks based on coherent population trapping," *Chinese Physics B*, vol. 23, no. 3, p. 030601, mar 2014. [Online]. Available: https://doi.org/10.1088/1674-1056/23/3/030601

[11] S. I. Roumeliotis and G. A. Bekey, "Distributed multi-robot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[12] G. Battistelli and L. Chisci, "Stability of consensus extended kalman filter for distributed state estimation," *Automatica*, vol. 68, pp. 169 – 178, 2016. [Online]. Available: https://doi.org/10.1016/j.automatica.2016.01.071

[13] Y. Wen, J. Zhu, Y. Gong, Q. Wang, and X. He, "Distributed orbit determination for global navigation satellite system with inter-satellite link," *Sensors*, vol. 19, p. 1031, 02 2019.

[14] F. Kirsch and M. Vossiek, "Distributed kalman filter for precise and robust clock synchronization in wireless networks," in *2009 IEEE Radio and Wireless Symposium*, 2009, pp. 482–485.

[15] J. Liu, Z. Wang, M. Zuba, Z. Peng, J. Cui, and S. Zhou, "Da-sync: A doppler-assisted time-synchronization scheme for mobile underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 582–595, 2014.

[16] D. Israel, L. V. Cooper, K. Mauldin, and K. Schauer, "Lunanet: A flexible and extensible lunar exploration communications and navigation infrastructure and the inclusion of smallsat platforms," in *2020 Small Satellite Conference*. Utah State University, 2020.

[17] T. Gardner and B. Cheetham, "Leaving no capstone unturned: How a cubesat pathfinder will enable the lunar gateway ecosystem," in *2020 Small Satellite Conference*. Utah State University, 2020.

[18] G. Chin, S. Brylow, M. Foote, J. Garvin, J. Kasper, J. Keller, M. Litvak, I. Mitrofanov, D. Paige, K. Raney *et al.*, "Lunar reconnaissance orbiter overview: Theáinstrument suite and mission," *Space Science Reviews*, vol. 129, no. 4, pp. 391–419, 2007.

[19] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2011. [Online]. Available: https://books.google.com/books?id=5WJOywAACAAJ

[20] *Orbital Data Messages*, Blue Book Series, Consulative Committee for Space Data Systems (CCSDS) Recommendation for Space Data System Standard CCSDS 502.0-B-2, Nov. 2009. [Online]. Available: https://public.ccsds.org/Pubs/502x0b2c1.pdf

[21] B. D. Tapley, B. E. Schutz, and G. H. Born, "Chapter 6 - consider covariance analysis," in *Statistical Orbit Determination*, B. D. Tapley, B. E. Schutz, and G. H. Born, Eds. Burlington: Academic Press, 2004, pp. 387 – 438.

[22] F. Pereira and D. Selva, "Exploring the design space of lunar gnss in frozen orbit conditions," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 444–451.

[23] T. Ely and E. Lieb, "Constellations of elliptical inclined lunar orbits providing polar and global coverage," *The Journal of the Astronautical Sciences*, vol. 54, 03 2006.

[24] A. Colaprete, D. Andrews, W. Bluethmann, R. C. Elphic, B. Bussey, J. Trimble, K. Zacny, and J. E. Captain, "An overview of the volatiles investigating polar exploration rover (viper) mission," *AGUFM*, vol. 2019, pp. P34B–03, 2019.

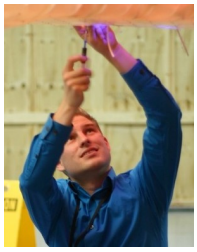
BIOGRAPHY

Benjamin Hagenau *is currently pursuing a B.S. in aerospace engineering sciences and an M.S. in astrodynamics and satellite navigation through a concurrent program at the University of Colorado at Boulder. He completed an internship at NASA Ames Research Center in 2020 where he was involved in the development of LAPS. Before this role he worked for short periods of time at Space Exploration Technologies and NASA Johnson Space Center. He is currently finishing his degrees.*

Brian Peters *received his B.S in Electrical Engineering from Ohio University in 2019 and is currently pursuing his M.S. in Electrical Engineering at Ohio University. He completed an internship at NASA Ames Research Center in 2020 where he was involved with the development of LAPS. He is currently a research assistant at Ohio University's Avionics Engineering Center.*

Roland Burton *received his M.Eng. in Aeronautical Engineering from Imperial College and his Ph.D in Aeronautics & Astronautics from Stanford University. He leads the development of LAPS, part of the DSA scalability studies. Before his current role, he has worked for OneWeb, Millennium Engineering Services, Field Street Capital Management and Lehman Brothers.*

Kelley Hashemi *is an engineer in the Intelligent Systems Division at NASA Ames and serves as a technical lead for autonomous systems research. Her current technical work addresses modeling, estimation, and control techniques for heterogeneous groups in both aeronautics and space applications. She holds an M.S. and Ph.D. in Aerospace Engineering from the University of Texas at Austin and received her B.S. from MIT.*



Nick Cramer *is the project manager of Distributed Spacecraft Autonomy at NASA Ames Research Center in the Intelligent Systems division. His areas of research span, distributed systems, intelligent structures, perception systems, and broad applications of autonomy. Nick occasionally teaches a UAV design class at San Jose State and received his Ph.D. in Computer Engineering with a focus in robotics from the University of California, at Santa Cruz.*