# Particle Swarm Optimization Based Tracking Window Planning for Cislunar Orbiters Performing Autonomous Radiometric Navigation

Turan, E.; Speretta, S.; Gill, E.K.A.

**Publication date**
2023

**Document Version**
Final published version

**Published in**
Proceedings of the 74th International Astronautical Congress (IAC)

74th International Astronautical Congress (IAC), Baku, Azerbaijan, 2-6 October 2023.


IAC-23-C1.5.10

# Particle Swarm Optimization Based Tracking Window Planning for Cislunar Orbiters Performing Autonomous Radiometric Navigation

**Erdem Turan[a*], Stefano Speretta[b], Eberhard Gill[c]**

[a] *Delft University of Technology (TU Delft), Kluyverweg 1, 2629 HS Delft, The Netherlands,* e.turan@tudelft.nl
[b] *Delft University of Technology (TU Delft), Kluyverweg 1, 2629 HS Delft, The Netherlands,* s.speretta@tudelft.nl
[c] *Delft University of Technology (TU Delft), Kluyverweg 1, 2629 HS Delft, The Netherlands,* E.K.A.Gill@tudelft.nl
[*] Corresponding Author

**Abstract**

In recent years, there has been a growing interest in lunar missions, particularly with the growing role of small satellites facilitated by piggyback launch opportunities. Typically, ground-based radiometric tracking is the workhorse to establish the necessary navigation in these missions. However, this could be expensive, while small satellites development is expected to be at low cost. To address this challenge, autonomous navigation presents a potential solution. This study explores a satellite-to-satellite tracking-based on-board orbit determination method for a satellite formation in cislunar space. Several factors affect the performance of orbit determination, and one critical aspect is the timing of tracking windows. Basically, it is crucial to determine when to collect the most useful observations to optimize the outcome of the navigation filter. In some cases, there might be operational constraints such as inter-satellite distance due to the limited onboard power for ranging. This study investigates the optimization of satellite-to-satellite tracking windows by using particle swarm optimization. The findings of this work demonstrate that particle swarm optimization offers an accurate solution for tracking windows, taking into account constraints arising from the spacecraft itself or from other design choices. In summary, particle swarm optimization provides near-optimal tracking windows by minimizing the overall orbit determination error. The results presented have the potential to enhance the design of satellite formations performing autonomous on-board orbit determination and contribute to cost-effective mission planning solutions.
**Keywords:** Orbit determination, Navigation, Particle swarm optimization, Satellite-to-satellite tracking, Cislunar

## 1. Introduction

In recent years, there has been a growing interest in small satellites, particularly for lunar missions. This trend can be attributed to significant advancements in satellite technology, which have led to smaller and more cost-effective satellite designs. Furthermore, the increasing availability of piggyback launch opportunities has opened up more frequent access to lunar missions. In particular, almost 40% of all planned deep-space small satellite missions, proposed by universities, companies, and space agencies, fall into this category [1].

Typically, these missions are based on ground-based radiometric tracking for Orbit Determination (OD). However, this approach can be costly, which contrasts with the goal of cost-effective small satellite development. Autonomous on-board Orbit Determination (AOD), on the other hand, offers the potential for cost reduction, improved performance, and increased reliability. Moreover, there are specific mission scenarios, such as rendezvous, where autonomy becomes essential. In addition, not only navigation but also operations can benefit from autonomy, as the associated costs are not mission-dependent [2].

Until now, various AOD techniques have been explored and put into practice: among these, the Linked Autonomous Interplanetary Satellite Orbit Navigation (LiAISON) relies exclusively on satellite-to-satellite observations to estimate not only relative but also absolute spacecraft states when at least one of the satellites follows an orbit characterized by distinct parameters, including its size, shape, and orientation [3]. Previous research studies have demonstrated LiAISON capabilities over the years in lunar missions. The OD performance of the method depends on various factors: observation type, accuracy, precision, and, most importantly, relative geometry between spacecraft [3, 4, 5]. Basically, it is crucial to know when to collect the most useful observations to optimize the outcome of the navigation filter. Although it is better to equally distribute measurements over the full orbit, in some cases it is not possible to track each spacecraft all the time because of various problems. In some cases, the communication window must also be used for telemetry and payload data download, as navigation and data transfer occur at the same time and with reduced performance. In these cases, the problem turns into finding the best communication window. This navigation concept is currently being evaluated as part of the CAPSTONE mission, which relies on the communication link established between the Lu-

nar Reconnaissance Orbiter (LRO) and the CAPSTONE CubeSat orbiting the Moon within the Near Rectilinear Halo Orbit (NRHO) [6]. In [7], ground-based radio tracking performance has been investigated from a tracking perspective and has been shown that there are tracking windows that should be avoided. The study also points out that navigation solutions are sensitive to the location and number of tracking arcs and their duration. On the other hand, this study aims at providing a Particle Swarm Optimization (PSO)-based tracking window planning for orbiters performing Satellite-to-Satellite Tracking (SST)-based AOD (instead of simulating all different configurations). In this study, tracking windows are determined by minimizing the overall OD error.

In the upcoming sections, autonomous radiometric navigation will be presented first, and then, tracking windows will be discussed. Orbit determination models, including dynamical, measurement, and estimation models, will be introduced. The navigation simulation setup and results will then be presented. Ultimately, conclusions will be drawn.

## 2. Autonomous Radiometric Navigation

This paper investigates the LiAISON OD capabilities of satellite formations within the lunar vicinity, focusing on different time frames for satellite-to-satellite tracking. AOD, in this context, requires the accurate determination of absolute satellite positions and velocities without relying on ground-based measurements (see Figure 1). In a two-body problem, inter-satellite measurements do not provide the absolute state estimation, since only the relative but not absolute orientation of orbital planes can be determined. However, in multi-body dynamics, full-state estimation is possible through satellite-to-satellite tracking. LiAISON utilizes these inter-satellite measurements to estimate absolute spacecraft states, given that one of the Spacecraft (S/C) follows an orbit with distinctive characteristics in terms of size, shape, and orientation. The OD performance depends on several factors, including type, accuracy, precision, frequency of observations, relative satellite geometry, and more. In general, OD is most effective in two-S/C configuration where the satellites follow highly elliptical, non-coplanar orbits with significant separation and short orbital periods. For such applications, range observations outperform range-rate observations [4, 5]. Although systematic biases can impact performance, the navigation system can still yield an acceptable state estimation even when inter-satellite measurement errors are relatively high, around the order of 100 meters ($1\sigma$) ranging error, within the lunar vicinity.

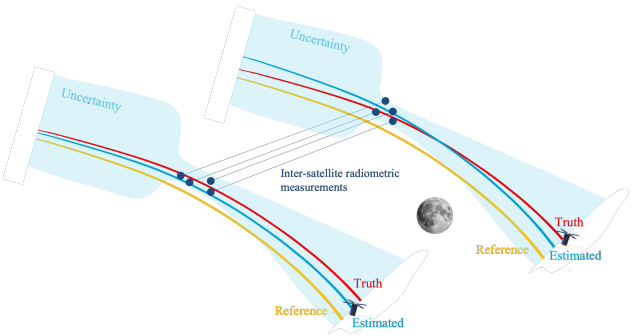


Fig. 1: Satellite-to-satellite tracking-based orbit determination.

## 3. Tracking Window

Tracking windows require critical considerations since they dictate when SST operations should take place. This timing plays a crucial role because tracking windows can produce different OD performance, potentially causing navigation requirements not being met and resulting in mission failures. It is worth noting that the best-placed tracking windows can align with nearly optimal OD results, which can translate to reduced propellant usage, simplifying mission operations, and prolonged mission durations. Grouping tracking windows, though, can also help reducing the overall set-up costs in tracking, like antenna pointing, actually lowering the overall navigation cost.

In essence, the key is to determine the most advantageous time windows for collecting observations that will maximize the effectiveness of the navigation filter. While it is generally ideal to equally distribute measurements across the entire orbit, practical constraints may block continuous tracking. These constraints can result from operational phases like Telemetry & Telecommand (TTC) and station-keeping periods. In some cases, it is feasible to modulate the ranging signal with telemetry or telecommand signals, effectively creating a single window for telemetry, telecommand, and tracking. However, conventional ranging methods require a specific amount of on-board power dedicated to the ranging signal, which could potentially limit the available power for telemetry. Consequently, in small satellites with limited on-board power employing SST, planning operations into different time frames becomes a necessity, as can be seen in Figure 2 (top).

In addition, there are alternative ranging techniques beyond conventional methods that streamline ranging operations, such as telemetry ranging [8] or teleme-

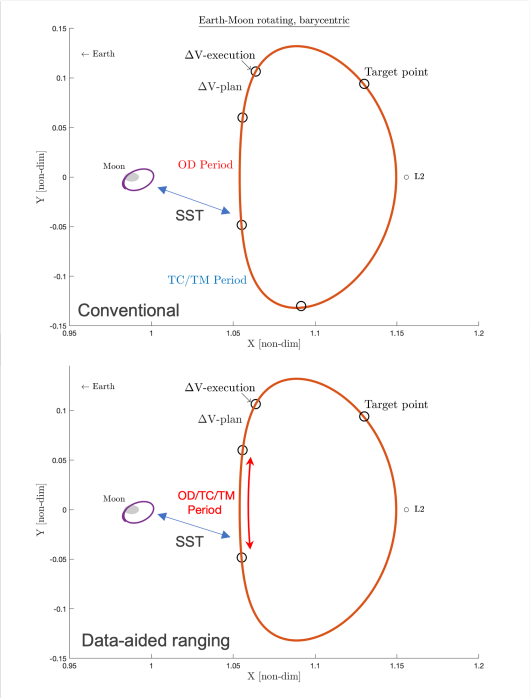


Fig. 2: Operational time windows.

try/telecommand ranging [9]. In such cases, it becomes possible to eliminate the ranging signal altogether, allowing the entire power capacity to be allocated for telemetry or telecommand transmissions. In these cases, the TTC and SST windows coincide, allowing all operations to be performed simultaneously as illustrated in Figure 2 (bottom). Furthermore, if necessary, the total duration of SST/TTC/OD can be extended, resulting in more navigation data and telemetry transmission. Even in this scenario, the challenge is to find the optimal communication window that yields the best OD performance.

## 4. Orbit Determination Models

This section introduces the models and methods employed for orbit determination and analysis in this paper. Subsections will present the dynamical, measurement, and estimation models.

### *4.1 Dynamical model*

In this study, the dynamical model takes the form of the Circular Restricted Three-body Problem (CRTBP). This assumes the presence of two massive bodies, Earth ($P_1$, with mass $m_1$) and the Moon ($P_2$, with mass $m_2$), both engaged in circular orbital motion around each other at a radius denoted as $r_{12}$, driven by their mutual gravitational force. A third body with mass $m_3$, with $m_3 \ll m_1, m_2$, exerts negligible influence on the motion of the primary bodies, $P_1$ and $P_2$. Additionally, a co-moving reference frame is employed, with its origin located at the barycenter of the two bodies. The positive x-axis points from the barycenter to $P_2$, the positive y-axis aligns with the $P_2$ velocity vector, and the z-axis is perpendicular to the orbital plane. The equations of motion governing the CRTBP are as follows [3]:

$$\ddot{x} = 2\dot{y} + x - (1-\mu)\frac{x+\mu}{r_1^3} - \mu\frac{x+\mu-1}{r_2^3} \quad [1]$$

$$\ddot{y} = (1 - \frac{1-\mu}{r_1^3} - \frac{\mu}{r_2^3})y - 2\dot{x} \quad [2]$$

$$\ddot{z} = (\frac{\mu-1}{r_1^3} - \frac{\mu}{r_2^3})z \quad [3]$$

where

$$r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$$

$$r_2 = \sqrt{(x+\mu-1)^2 + y^2 + z^2}$$

For the Earth-Moon system, the gravitational parameter $\mu$ is 0.01215, the normalized time $t^*$ 4.343 days, and the normalized length $l^*$ 384 747.96 km, respectively.

It is important to note that the method described in this paper can also be utilized for high-fidelity dynamics; however, employing PSO together with high-fidelity dynamics would require a considerable amount of computational resources.

### *4.2 Measurement model*

In this research, we focus on two-way pseudorange observations. The estimated state vector consists of the position and velocity components of the two spacecraft, resulting in a 12-dimensional state vector as follows:

$$\mathbf{X} = (\mathbf{r}_1, \mathbf{v}_1, \mathbf{r}_2, \mathbf{v}_2)^{\mathrm{T}} \quad [4]$$

where the subscript represents the S/C number.

The crosslink range measurement model, referred to as pseudorange, includes the geometric distance $R$, deterministic and stochastic clock errors, as well as systematic and random errors expressed as follows:

$$\rho = R + c\sum_{l=0}^{2} \Delta\tau_c^l + c\delta t_s + \rho_{\text{noise}} \quad [5]$$

where $\Delta\tau_c^l$ represents the deterministic clock errors, mainly: bias, drift, and aging for $l = 1, 2, 3$, respectively. Additionally, the stochastic clock error is given as $\delta t_s$. It is worth noting that $\rho_{\text{noise}}$ signifies the thermal noise that originates from the ranging system, which is typically the most significant source of error. These terms are combined in the equations denoted as $\rho_{\text{bias}}$ and $\rho_{\text{noise}}$, encompassing

sources of unmodeled statistical error. The range model is given as:

$$\rho = \sqrt{(\mathbf{r}_1 - \mathbf{r}_2) \cdot (\mathbf{r}_1 - \mathbf{r}_2)} + \rho_{\text{bias}} + \rho_{\text{noise}} \quad [6]$$

Light-time corrections are neglected, as the speed of light is greater than the S/C relative velocity and the relative distance is not large enough to introduce significant errors.

Several inter-satellite radiometric ranging methods exist, including conventional Pseudo Noise (PN) code, tone-based, telemetry or frame ranging [8, 10] but in this study, we will use telemetry ranging, as it allows the concurrent execution of all operations, including SST and TTC, within the same time window. The telemetry ranging jitter is given as follows [8]:

$$\sigma_\rho = \left(1 - \frac{2v_r}{c}\right)\left(\frac{4\,c\,T_{sd}^2}{\pi\,T\,E_S/N_0} + \frac{c}{8 f_{rc}}\sqrt{\frac{B_L}{(P_{RC}/N_0)}}\right) \quad [7]$$

where $T_{sd}$ is the telemetry symbol duration, $c$ the speed of light, $v_r$ relative velocity between satellites, $T_l$ the correlator integration time, $E_S/N_0$ the symbol-to-noise ratio, $f_{rc}$ the frequency of the ranging clock component, and $B_L$ one-sided loop noise bandwidth. It has been assumed that the onboard time-tagging is precise enough not to affect the overall system performance.

### *4.3 Estimation model*

In this study, the Extended Kalman Filter (EKF) serves as the estimation model: it operates by taking an initial state $\mathbf{X}_0$ and state covariance $\mathbf{P}_0$ at the initial time and subsequently processes inter-satellite range observations at all the measurement epochs [11]. The EKF is composed of two main steps: a prediction step and a correction step, as summarized in Algorithm 1, where $f$ is the system dynamics, $\mathbf{X}$ is the state vector, $\mathbf{h}$ is the measurement model, $\tilde{\omega}$ is the process noise, $\mathbf{v}$ is the measurement noise, $P$ is the covariance matrix, $K$ is the Kalman gain, $Q$ and $W$ are process and measurement noise covariance matrices, respectively.

In one of the cases given in this paper, we examine how measurement biases affect tracking windows: this investigation involves introducing a measurement bias as a *consider* parameter to be taken into account. In this approach, the state vector is estimated and the uncertainty of the measurement model (bias in this case) is included in the estimation error covariance matrix. This approach is called Consider Kalman Filter (CKF), and its implementation is detailed in Algorithm 2, where $C_k$ is the cross-covariance matrix, $B_0$ is the parameter covariance, $b_0$ is *a priori* the parameter error. It is worth noting that CKF turns into EKF in the case of zero parameter uncertainty.

**Algorithm 1:** Extended Kalman Filter (EKF)

*Given:* $P_0, \mathbf{X}_0, \tilde{\mathbf{y}}_k, Q, W_k$
**Model:**
$\dot{\mathbf{X}} = f(\mathbf{X}, \mathbf{u}, t) + \tilde{\omega}, \;\; \tilde{\omega} \sim N(\mathbf{0}, Q)$
$\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{X}_k) + \mathbf{v}_k, \;\; \mathbf{v}_k \sim N(\mathbf{0}, W)$
**Initialize:**
$\hat{\mathbf{X}}_0 = E\left\}\hat{\mathbf{X}}_0\right\}$
$P_0 = E\left\{\tilde{\mathbf{X}}_0\tilde{\mathbf{X}}_0^T\right\}$
**Propagation:**
$\dot{\hat{\mathbf{X}}} = f(\hat{\mathbf{X}}, \mathbf{u}, t)$
$\bar{P}_k = \Phi(t_k, t_{k-1})P_{k-1}\Phi^T(t_k, t_{k-1}) + Q$
$\dot{\Phi}(t_k, t_{k-1}) =$
$(\partial f(\mathbf{X}, \mathbf{u}, t)/\partial \mathbf{X})\,\Phi(t_k, t_{k-1}), \;\; \Phi(t_0, t_0) = I$
**Update:**
$K_k = \bar{P}_k\tilde{H}_k^T[\tilde{H}_k\bar{P}_k\tilde{H}_k^T + W_k]^{-1}, \;\; \tilde{H}_k =$
$\partial \mathbf{h}(\mathbf{X}_k)/\partial \mathbf{X}_k$
$\hat{\mathbf{X}}_k^+ = \hat{\mathbf{X}}_k^- + K_k[\tilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{X}}_k^-)]$
$P_k = [I - K_k\tilde{H}_k]\bar{P}_k$

**Algorithm 2:** Consider Kalman Filter (CKF)

*Given:* $P_0, B_0, C_0, \mathbf{X}_0, \mathbf{b}_k, \tilde{\mathbf{y}}_k, Q, W_k$
**Model:**
$\dot{\mathbf{X}} = f(\mathbf{X}, \mathbf{u}, t) + \tilde{\omega}, \;\; \tilde{\omega} \sim N(\mathbf{0}, Q)$
$\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{X}_k, \mathbf{b}_k) + \mathbf{v}_k, \;\; \mathbf{v}_k \sim N(\mathbf{0}, W)$
**Initialize:**
$\hat{\mathbf{X}}_0 = E\left\}\hat{\mathbf{X}}_0\right\}$
$\mathbf{b}_0 = E\,\}\mathbf{b}_0\}$
$P_0 = E\left\{\tilde{\mathbf{X}}_0\tilde{\mathbf{X}}_0^T\right\}$
$B_0 = E\left\{\mathbf{b}_0\mathbf{b}_0^T\right\}$
$C_0 = E\left\{\tilde{\mathbf{X}}_0\mathbf{b}_0^T\right\} = 0$
**Propagation:**
$\dot{\hat{\mathbf{X}}} = f(\hat{\mathbf{X}}, \mathbf{u}, t)$
$\bar{P}_k = \Phi(t_k, t_{k-1})P_{k-1}\Phi^T(t_k, t_{k-1}) + Q$
$\dot{\Phi}(t_k, t_{k-1}) =$
$(\partial f(\mathbf{X}, \mathbf{u}, t)/\partial \mathbf{X})\,\Phi(t_k, t_{k-1}), \;\; \Phi(t_0, t_0) = I$
$\bar{C}_k = \Phi(t_k, t_{k-1})C_{k-1}$
**Update:**
$K_k = (\bar{P}_k\tilde{H}_k^T + \bar{C}_kN_k^T)[\tilde{H}_k\bar{P}_k\tilde{H}_k^T +$
$N_k\bar{C}_k^T\tilde{H}_k^T + \tilde{H}_k\bar{C}_kN_k^T + N_kB_0N_k^T + W_k]^{-1}$
$\tilde{H}_k = \partial \mathbf{h}(\mathbf{X}_k, \mathbf{b}_k)/\partial \mathbf{X}_k, \;\; N_k =$
$\partial \mathbf{h}(\mathbf{X}_k, \mathbf{b}_k)/\partial \mathbf{b}_k$
$\hat{\mathbf{X}}_k^+ = \hat{\mathbf{X}}_k^- + K_k[\tilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{X}}_k^-, \mathbf{b}_k) - N_k\mathbf{b}_k]$
$P_k = [I - K_k\tilde{H}_k]\bar{P}_k - K_kN_k\bar{C}_k^T$
$C_k = \bar{C}_k - K_k(\tilde{H}_k\bar{C}_k + N_kB_0)$

## 5. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an optimization algorithm inspired by the collective behavior of animals [12]where a population of potential solutions, represented as particles, iteratively explores a search space. Each particle adjusts its position and velocity based on its own experience and that of its peers. PSO efficiently navigates complex, high-dimensional spaces to find near-optimal solutions. It is widely used in various fields, due to its simplicity and effectiveness in tackling optimization and search problems.

There are various OD performance analysis metrics given in scientific literature that can be derived through methods such as Monte Carlo simulations, Cramér-Rao Lower Bound (CRLB), or the observability analysis that includes observability indices, condition numbers, Root Mean Square (RMS) or Root-Sum-Square (RSS) position/velocity error or uncertainty. Any of these metrics can serve the same purpose as outlined in this research. However, for our study, we have adopted the approach given in [3], which is based on the length of the largest axis of the error ellipsoid:

$$\beta_i = 3\max(\sqrt{\lambda_j}) \quad [8]$$

where $\lambda_j$ for $j = 1, 2$ are the eigenvalues of $\mathbf{P}_{i_{3\times3}}$ (representing position or velocity components of S/C) and the average value of the two spacecraft system

$$\bar{\beta} = \frac{1}{n}\sum_{i=1}^{n}\beta_i, \qquad \beta_{ave} = \frac{1}{2}\sum_{j=1}^{2}\bar{\beta}_j \quad [9]$$

where $n$ is the number of $\beta$ values during the simulation and the system average $\beta_{ave}$ is given as $\beta_{ave}$.

In summary, PSO can be used to minimize the overall position and velocity uncertainty by finding near-optimal tracking windows. In this context, the optimization problem is given as follows:

$$\begin{aligned} \min \quad & \beta_{ave} = \frac{1}{2}\sum_{j=1}^{2}\frac{1}{n}\sum_{i=1}^{n}3\max(\sqrt{\lambda_j}) \\ \text{s.t.} \quad & 0 < T_{SST} < T_{end} \end{aligned} \quad [10]$$

where $T_{SST}$ represents SST windows (timing points) with the number $w = 1, 2, 3$ (e.g., three tracking windows), while $T_{end}$ is the simulation duration. It should be noted that, based on different scenarios, the constraints are varied throughout this study and more details are given in Section 6.

In brief, the OD simulation starts with an array containing the initial times of each tracking window determined by the PSO particles. Subsequently, the filter utilizes SST observations obtained during these tracking windows with the starting times defined by the particles and a fixed tracking duration. As the simulation progresses, $\beta_{ave}$ is assigned to the respective particle, and this process continues until the maximum number of particles is achieved in the first iteration. In each iteration, the global fitness value (which represents the minimum position and velocity uncertainty) is stored until the maximum iteration count is reached. This process can be seen in Algorithm 3.

**Algorithm 3:** Particle Swarm Optimization (PSO)

*Given:* $P_0, \mathbf{X}_0, Q, N_k, LB, UB$
**Minimize:**
$f(X)$, minimizing $\beta_{ave}$
subject to $0 < T_{SST} < T_{end}$,
$T_{SST} = [T_1, T_2, ..., T_l]$
(can be extended as $\rho < R_{req}$ or
$T_1 << T_0 + 3\,\text{days}$
**Initialize:** For each particle $i$ in a swarm
population size Ps
$X_i$ a random vector within [LB UB]
$V_i$ a random vector within [LB UB]
Evaluate the fitness f($X_i$)=$\beta_{ave,i}$
$pbest_i$ with a copy of $T_{SST,i}$
$gbest$ with a copy of $T_{SST,i}$ with the best fitness
**while** $k \geq N_k$ **do**
  $i \leftarrow i + 1$;
  Update $V_i^k$ and $X_i^k$ for each particle $i$ by
  $V_i^{k+1} = wV_i^k + c_1 rand_1(pbest_i - X_i^k +$
  $c_2 rand_2(gbest - X_i^k))$
  $X_i^{k+1} = X_i^k + V_i^{k+1}$
  Evaluate fitness $f(X_i^k)$
  $pbest_i \leftarrow X_i^k$ if $f(pbest_i) < f(X_i^k)$
  $gbest_i \leftarrow X_i^k$ if $f(gbest_i) < f(X_i^k)$
**end**

## 6. Simulations

This section presents the OD results using the tracking windows optimized with PSO. The simulation setup is presented first, and, thereafter, results are provided.

### *6.1 Simulation setup*

This study investigates AOD for cislunar formations using SST-based radiometric measurements, focusing on a cislunar satellite formation made by two or three satellites, as shown in Figure 3. Initial simulations will focus on a two-satellite formation that includes a satellite in an Earth-Moon L1 NRHO ($T = 7.83$ days, $C_j = 3.00$) and a L2 orbiter ($T = 14.14$ days, $C_j = 3.11$). Later simulations will also include a second orbiter around the Earth-Moon L1 point ($T = 12.10$ days, $C_j = 3.10$).

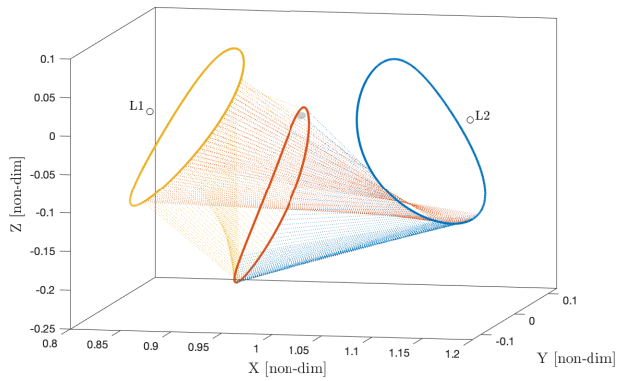


Fig. 3: A cislunar satellite formation formed by 3 S/C in a mesh topology. In orange the EM L1 NRHO, in blue the EM L2 Halo orbiter, and in yellow the EM L1 Halo orbiter.

Table 1: Radiometric parameters

| **Inter-Satellite Link Budget** | **Value** |
|---|---|
| Frequency, $f$ | 2200 MHz/2100MHz |
| TX power, $P_t$ | 3 dBW |
| TX path losses, $L_t$ | 1 dB |
| TX antenna gain, $G_t$ | 6.5 dBi |
| Polarisation loss, $L_p$ | 0.5 dB |
| Data rate | 5 kbps |
| Required $E_b/N_0$ | 2.5 dB |
| Link Margin | 3 dB |
| **Inter-Satellite Radiometric Measurement Parameters** | **Value** |
| Symbol rate, $1/T_s d$ | 5 ksps |
| Correletor integration time, $T_l$ | 1 s |
| Symbol-to-noise ratio, $E_s/N_0$ | -1 dB |
| Modulation | BPSK |
| Range clock frequency $f_r c$ | 1 MHz |
| Range clock to noise spectral density, $P_r c/N_0$ | 25 dBHz |
| Tracking Loop Bandwidth, $B_L$ | 1 Hz |
| Ranging jitter, $\sigma_\rho$, see Eqn.7 | 10 m |

The study generates true and estimated trajectories using the ODE113 solver in Matlab, with a simulation duration, $T_m$, set to $21\,\mathrm{days}$ (1.5 times the longest orbital period in the formation). Telemetry ranging has been assumed as the inter-satellite ranging method, and measurement errors ($1\sigma$ error of $10\,\mathrm{m}$) have been considered without any bias (see Table 1 for further details). However, a $10\,\mathrm{m}$ bias is considered in the CKF case.

The initial state errors for each state parameter are set to $500\,\mathrm{m}$ for position and $5\,\mathrm{mm/s}$ for velocity, while the initial state uncertainty ($1\sigma$) is set to $1\,\mathrm{km}$ for position and $1\,\mathrm{cm/s}$ for velocity. Each tracking window, as a reference setting, spans a continuous $2\,\mathrm{days}$ period with measurements taken every 180 seconds. To put this into context, an example taken from ground-based conventional PN tracking (assuming the T2B code) involves approximately $175\,\mathrm{s}$ for signal acquisition, $0.25\,\mathrm{s}$ to $500\,\mathrm{s}$ for signal correlation (including ambiguity solving) and user-defined measurement intervals on the major tone (e.g., $0.5\,\mathrm{s}$) [10]. In practice, additional measurements can be conducted, and observations can be averaged after signal acquisition to enhance accuracy. Given the total tracking window duration and the overall duration of the ranging sequence, a measurement cadence of $180\,\mathrm{s}$ is considered an appropriate choice.

It is important to note that the satellites within the formation have different orbital periods and, consequently, the measurement geometry continually changes over time, making it impossible to plan SST when one orbiter passes a specific point in its orbit, and the other is at a completely different location. For example, the first orbiter may be in a high-velocity region in the NRHO, while the second orbiter is in a low-velocity region in the EM L2 Halo orbit. Due to this variability, the orbit was not divided into separate segments, and the different combinations have not been simulated. Instead, PSO is employed to seek a near-optimal solution, providing the best averaged OD uncertainty. To achieve this objective, we explored six distinct scenarios to identify the most favorable tracking windows under different conditions:

- Case-A: Four tracking arcs.
- Case-B: Four tracking arcs, each within four equally divided time windows.
- Case-C: A single tracking arc initiated after the convergence period.
- Case-D: A single tracking arc planned when the inter-satellite distance is less than $90\,000\,\mathrm{km}$.
- Case-E: Three tracking arcs that incorporate measurement bias as a consider parameter.
- Case-F: Three tracking arcs for a satellite formation consisting of three spacecraft in a mesh topology, where all satellites are linked with each other.

Simulation results are presented in the next sections, starting from the nominal scenario and then continuing by progressively adding more constraints.

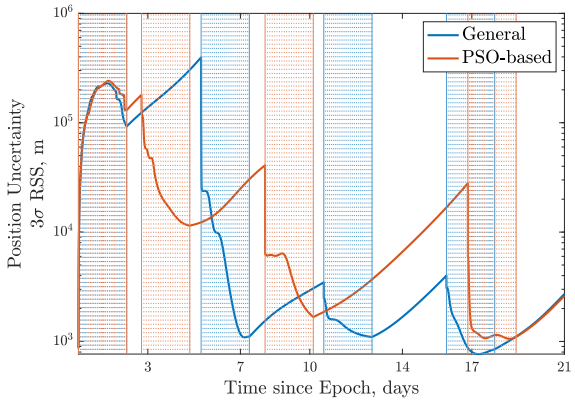


Fig. 4: Comparison of PSO-based and equally distributed tracking windows (EML1 Halo/EML2 Halo case). Vertical lines represent SST points for each solution.

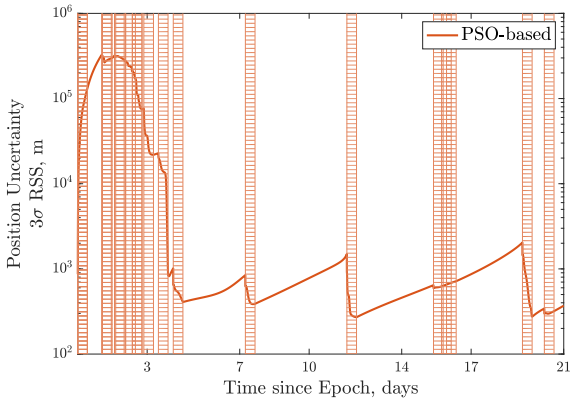


Fig. 5: PSO-based SST windows and corresponding position uncertainty (EML1 NRHO/EML2 Halo case)

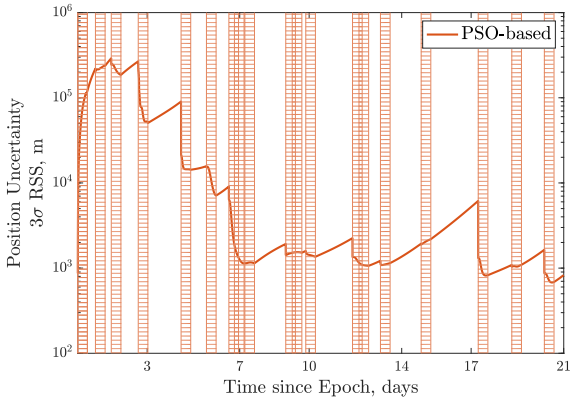


Fig. 6: PSO-based SST windows and corresponding position uncertainty (EML1 Halo/EML2 Halo case).

### *6.2 Case-A: Nominal*

This presents the standard scenario in which the goal is to identify SST windows (a total of $N$) providing the best overall OD accuracy. Here, we impose straightforward limitations, including tracking arcs, which must fall within the range of 0 to $T_{end}$. We have selected four tracking windows, each comprising 1000 measurements taken at intervals of 180 seconds, resulting in approximately 2.08 days of continuous tracking. A further constraint was imposed to avoid two tracking windows to overlap: the PSO normally avoids this and would not provide optimal results, but the constraint limits the search space and reduces the simulation time. To sum up, these constraints can be outlined as follows:

$$\begin{aligned} \min \quad & \beta_{ave} \\ \text{s.t.} \quad & 0 < T_{SST} < T_{end} \\ & T_{SST} = [T_1, T_2, T_3, T_4] \\ & T_l = 2.08\,\text{days} \\ & T_{l+1} - T_l = 2.08\,\text{days} \end{aligned} \tag{11}$$

Figure 4 presents the comparison between equally distributed tracking windows and the tracking windows provided by the PSO technique: the colored vertical lines in figure represent SST points for each approach. Basically, PSO pushes the very first two tracking windows into the very initial phase of the mission, as expected, due to the high initial uncertainty in the system. On the other hand, distributing tracking arcs couldn't provide an accurate solution, since the first tracking session was not long enough to capture useful information. This results in around $6\,\text{km}$ ($3\sigma$) increase in overall position uncertainty.

The previous case was good for illustration purposes since there were only four tracking sessions. However, almost 2 days of continuous tracking might be too long for small satellite missions. For this reason, in the second scenario, each tracking session is reduced to 10 hours ($T_l = 10\,\text{hours}$), representing 200 measurements once every 180 s. Figure 5 and 6 illustrate what would happen in this scenario, showing a trend for the first figure, EML1 NRHO/EML2 Halo: pushing tracking windows into the very early phase of a mission to reduce the initial position uncertainty and distributing fewer tracking sessions at a later stage, since convergence had already been achieved. The second one, EML1 Halo/EML2 Halo, shows a trend of distributing tracking sessions over the full mission.

### *6.3 Case-B: Equally divided time windows*

In the previous case, the tracking windows were distributed over the full mission duration; however, there might be the need to have a tracking window at specific intervals due to mission constraints. This then requires one

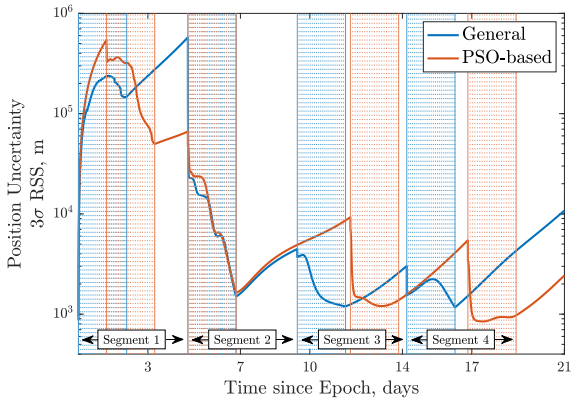


Fig. 7: Comparison of PSO-based and equally distributed tracking windows (EML1 Halo/EML2 Halo case).

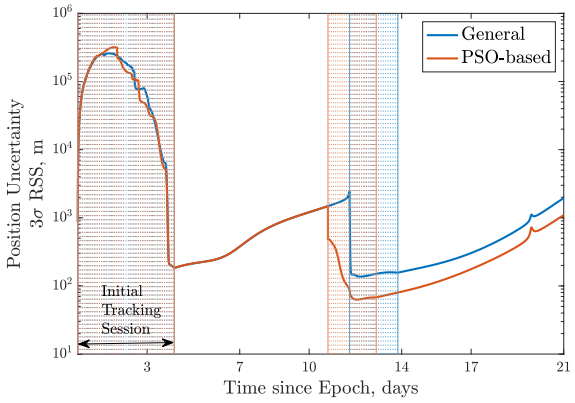


Fig. 8: Comparison of PSO-based and nominal (midpoint) tracking windows (After convergence, EML1 NRHO/EML2 Halo case).

to divide the full mission span into segments and place a tracking arc into each individual part, and the PSO can help find the best tracking windows for each corresponding segment. We can formulate the problem as follows:

$$\begin{aligned} \min \quad & \beta_{ave} \\ \text{s.t.} \quad & 0 < T_{SST} < T_{end} \\ & T_{SST} = [T_1, T_2, T_3, T_4] \\ & LB = [0, 4.7, 9.4, 14.2](\text{days}) \\ & UB = [4.7, 9.4, 14.2, 18.9](\text{days}) \\ & T_l = 2.08\,\text{days} \\ & T_{l+1} - T_l = 2.08\,\text{days} \end{aligned} \quad [12]$$

Figure 7 illustrates the corresponding result: it is interesting to see in this case that the PSO skipped the very first days and placed the first tracking arc around day 2. This means that the relative geometry between satellites (thus, the measurement geometry) is better after around day 2 than the very first days, compensating for the increase in state uncertainty. Overall, around $4.5\,\text{km}$ $(3\sigma)$ positional improvement has been observed ($26\,\text{m}$ after the second tracking window).

### *6.4 Case-C: After the convergence period*

Until now, previous simulations have indicated the advantages of having tracking windows during the initial phases of missions, primarily due to the presence of high initial uncertainties. From an operational perspective, there will be an initial OD performed either from ground-based sources or from another external source, which provides an initial state to the on-board filter. After the convergence period, the PSO can seek the near-optimal tracking window: a single tracking window starting after day 4, determined by running a separate simulation. The optimization problem can be defined as:

$$\begin{aligned} \min \quad & \beta_{ave} \\ \text{s.t.} \quad & 0 < T_{SST} < T_{end} \\ & T_{SST} = T_1 \\ & LB = 4\,\text{days} \\ & UB = 18.92\,\text{days} \\ & T_l = 2.08\,\text{days} \end{aligned} \quad [13]$$

The comparison between a single tracking window found by PSO and a midpoint selected tracking window is shown in Figure 8 where the PSO shifts the tracking window a bit earlier than the midpoint of the mission, resulting in a better overall state uncertainty. In case the number of tracking windows is increased to ten, while each tracking session is reduced to 10 hours, the same pattern can be seen as given in Figure 9 ($\approx$125 m $(3\sigma)$ positional improvement with PSO).

### *6.5 Case-D: constrained distance*

Another operational constraint can be related to the communications system: inter-satellite distance can be limited due to the onboard power that can be used for ranging. For such cases, tracking windows may be restricted to short time intervals only when the distance is lower than a certain value. In this case, we have checked the mean inter-satellite distance ($90\,000\,\text{km}$) and set it as a constraint: Figure 10 illustrates the corresponding results,

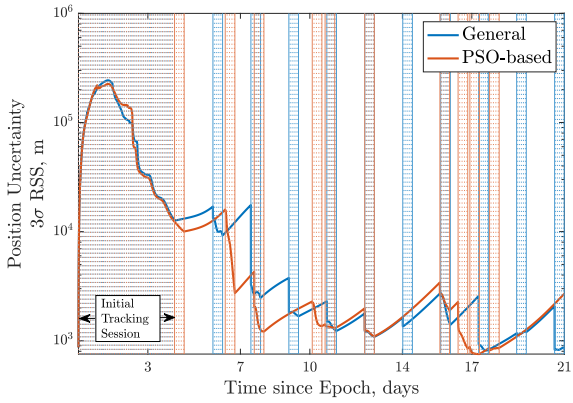


Fig. 9: Comparison of PSO-based and equally distributed tracking windows (After the initial tracking session, EML1 Halo/EML2 Halo case).

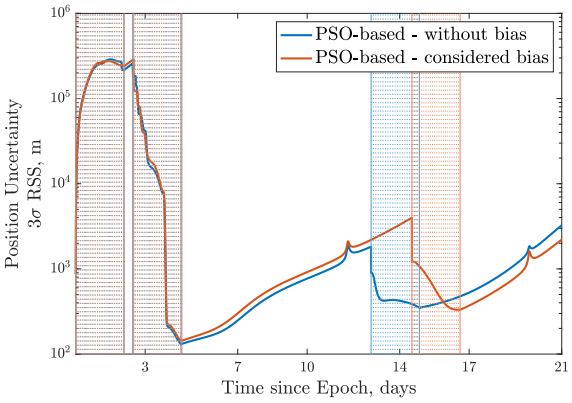


Fig. 11: PSO-based tracking window planning under the measurement bias constraint.

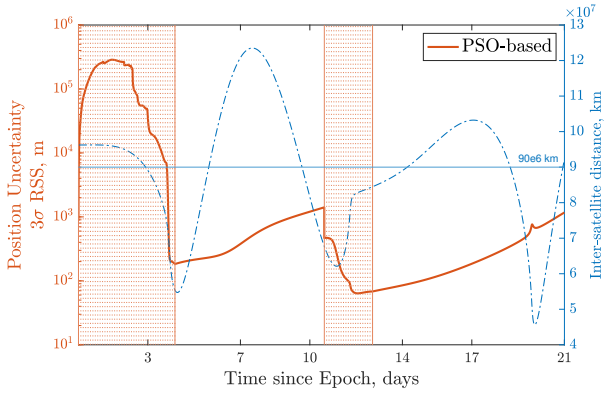


Fig. 10: PSO-based tracking window planning under the distance constraint.

while the following equations represent the problem:

$$\begin{aligned} \min \quad & \beta_{ave} \\ \text{s.t.} \quad & 0 < T_{SST} < T_{end} \\ & T_{SST} = [T_1, T_2, T_3] \\ & \rho \leq 90\,000\,\text{km} \\ & T_l = 2.08\,\text{days} \\ & T_{l+1} - T_l = 2.08\,\text{days} \end{aligned} \quad [14]$$

### 6.6 *Case-E: Considered bias*

It is well known that systematic bias has a significant impact on measurement quality, consequently influencing the OD performance. There are various approaches to handling systematic biases, and one of them is implemented in this study as CKF. The filter, in this case, takes into account the impact of measurement bias, which has been assumed $10\,\text{m}$. The optimization problem is the same as given in Eqn. 11, the only difference being the covariance matrix in the filter.

Figure 12 presents cases where both PSO-based tracking windows. However, there is a distinction between them: the former takes into account measurement bias and incorporates it into the filter, while the latter does not have any bias in measurements. As can be seen, PSO provides completely different solutions, highlighting the idea that systematic biases can affect the timing of tracking windows. It is worth noting that the average OD uncertainty is around $9773\,\text{m}$ $(3\sigma)$ for the considered bias case, while it is around $9665\,\text{m}$ $(3\sigma)$ for the no-bias scenario, as expected. Using the tracking window of the no-bias case instead of the tracking window of the considered case brings the average solution of $9933\,\text{m}$ $(3\sigma)$ when measurements are affected by a constant bias. This demonstrates that measurement biases can impact OD uncertainty under specific geometries, leading PSO to recommend a different time window to minimize overall OD uncertainty.

### 6.7 *Case-F: Network topology*

Adding a third satellite (in a mesh topology) to the cislunar satellite formation (see Figure 3 for the full configuration) leads to an optimization problem equivalent to the one expressed in Eqn.11, with the sole distinction being the size of the estimated state vector (18 instead of 12) and the increased number of inter-satellite links (number of measurements). The PSO-driven SST windows are illustrated in Figure 12, where it is interesting to see that the final tracking window is long enough to maintain a stable uncertainty. In this case, conducting just a few hours of tracking would be sufficient, but it is worth noting that OD performance is more accurate accurate (average $2141\,\text{m}$, $3\sigma$) in the mesh topology due to the higher amount of

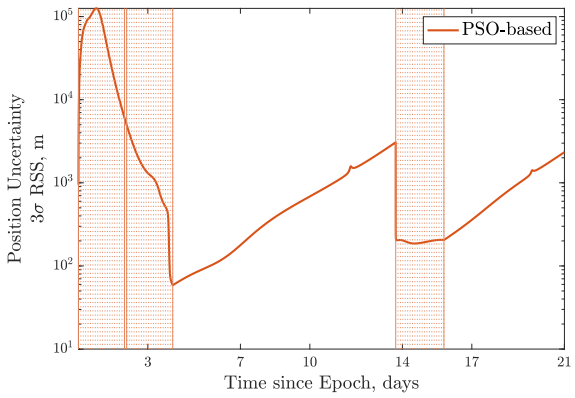


Fig. 12: PSO-based tracking window planning for a satellite formation formed by 3 S/C in a mesh topology.

links, allowing for more relaxed tracking durations.

In this section, it has been demonstrated that PSO can provide an accurate solution for SST windows for cislunar satellite formations performing on-board OD. This algorithm can be used in advance on the ground to plan TTC windows. Although the given approach is applied only to the EM L1 Halo/NRHO/L2 Halo case, this method is applicable to missions in different orbital regimes. It is worth noting that there are several limitations that can be further investigated in the future. In this study, the tracking durations are assumed to be fixed. However, these durations can also be optimized to have much more compact planning. Additionally, two-way ranging operations are assumed in this study. However, one-way ranging operations can be investigated by estimating clock parameters along with spacecraft states by minimizing the overall OD uncertainty.

## 7. Conclusions

This study introduced a PSO-based approach for planning satellite-to-satellite tracking windows in cislunar orbits, particularly for satellite formations performing autonomous on-board orbit determination. The algorithm was tested with various operational constraints, demonstrating that PSO effectively addresses constraints related to spacecraft characteristics and design choices. Even in scenarios involving multiple satellites within a formation, the proposed algorithm was able to provide accurate solutions, simplifying operational complexities. Furthermore, this technique can readily be adapted for other cases, such as minimizing station-keeping maneuver costs through the identification of near-optimal SST windows. The primary benefit of this algorithm lies in its ability to offer a straightforward solution without the need to account for various SST configurations. It has been highlighted in this study that uncertainties in measurements, such as biases, can impact tracking window selection when near-optimal solutions are required. Additionally, the algorithm can easily integrate communication power constraints (inter-satellite distances), simplifying complex implementations. The presented results offer valuable information to refine the design of satellite formations performing autonomous on-board OD and obtaining cost-effective solutions for mission planning.

## References

[1] E. Turan, S. Speretta, and E. Gill, "Autonomous navigation for deep space small satellites: Scientific and technological advances," *Acta Astronautica*, vol. 193, pp. 56–74, Apr. 2022.

[2] S. Speretta, F. Topputo, J. Biggs, P. D. Lizia, M. Massari, K. Mani, D. D. Tos, S. Ceccherini, V. Franzese, A. Cervone, P. Sundaramoorthy, R. Noomen, S. Mestry, A. do Carmo Cipriano, A. Ivanov, D. Labate, L. Tommasi, A. Jochemsen, J. Gailis, R. Furfaro, V. Reddy, J. Vennekens, and R. Walker, "LUMIO: achieving autonomous operations for lunar exploration with a CubeSat," in *2018 SpaceOps Conference*, American Institute of Aeronautics and Astronautics, May 2018.

[3] K. A. Hill, *Autonomous Navigation in Libration Point Orbits*. PhD thesis, University of Colorado, 2007.

[4] K. Hill, J. Parker, G. Born, and N. Demandante, "A Lunar L2 Navigation, Communication, and Gravity Mission," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, American Institute of Aeronautics and Astronautics, Aug. 2006.

[5] E. Turan, S. Speretta, and E. Gill, "Performance analysis of crosslink radiometric measurement based autonomous orbit determination for cislunar small satellite formations," *Advances in Space Research*, vol. 72, pp. 2710–2732, Oct. 2023.

[6] M. R. Thompson, A. Forsman, S. Chikine, B. C. Peters, T. Ely, D. Sorensen, J. Parker, and B. Cheetham, "Cislunar navigation technology demonstrations on the CAPSTONE mission," in *The International Technical Meeting of the The Institute of Navigation*, Institute of Navigation, Feb. 2022.

[7] M. Bolliger, M. R. Thompson, N. P. Ré, C. Ott, D. C. Davis, and N. Parrish, "Ground-based navigation trades for operations in gateway's near rectilinear halo orbit," in *31st AAS/AIAA Space Flight Mechanics Meeting*, 2021.

[8] K. Andrews, J. Hamkins, S. Shambayati, and V. Vilnrotter, "Telemetry-based ranging," in *2010 IEEE Aerospace Conference*, pp. 1–16, IEEE, 2010.

[9] V. Vilnrotter and J. Hamkins, "Telecommand/telemetry ranging for deep-space applications," in *2019 IEEE Aerospace Conference*, pp. 1–10, IEEE, 2019.

[10] The Consultative Committee for Space Data Systems, "Pseudo-Noise (PN) Ranging Systems," tech. rep., CCSDS Secretariat , 2014.

[11] B. Tapley, B. E. Schutz, and G. H. Born, *Statistical Orbit Determination*. Elsevier Academic Press, Burlington, USA, 2004.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE, 1995.