# TATER ToT - Deciding When to Track or Target to Balance Spacecraft $\Delta V99$ and Tracking Time

Sai Chikine
*Aerospace Engineering*
*CU Boulder*
Boulder, USA
sai.chikine@gmail.com

***Abstract*—A key step of the spacecraft mission design process is assessing the $\Delta V99$ budget - the 99th percentile of $\Delta V$ accounting for all of the uncertainties involved in the mission, including maneuver execution errors and, critically, navigation errors. In turn, this process can help to set navigation performance requirements. In this work, a POMDP formulation is used to represent this scenario by modeling a problem where a spacecraft can choose to either gather tracking data to reduce its uncertainty or perform a maneuver with the aim of balancing $\Delta V99$ costs and tracking costs. POMCPOW is used to solve a coplanar, coaxial transfer problem from a low Earth orbit to a geosynchronous orbit. This POMDP formulation approach is entitled "TATER ToT" - Trained Agent That Expertly Ranks Tracking or Targeting. The POMCPOW results were found to be favorable to simple heuristic policies. This approach and results are promising and suggest a potentially powerful alternative to traditional methods of determining $\Delta V99$ budgets and navigation requirements.**

***Index Terms*—spacecraft mission design, POMDPs**

## I. Introduction

In initial spacecraft mission design, the nominal $\Delta V$ cost is obtained from the deterministic optimal reference trajectory. However, since designers are aware that maneuvers need to replanned on-orbit according to actual state estimates obtained in operations, real maneuver costs will differ from the nominal costs according to the actual state estimate's deviation from the reference, as well as other factors such as thrust magnitude and pointing errors. To account for this, designers compute a $\Delta V99$ value - the 99th percentile of $\Delta V$ when factoring in all modeled uncertainties. If the spacecraft's state estimate is too uncertain, states within its state distribution may be far enough from the mission reference trajectory that resulting maneuvers cause a very high $\Delta V99$ value. Conversely, to operate within a fixed mission $\Delta V$ budget, $\Delta V99$ values may be used to set requirements on navigation performance. Beyond Earth orbit where GPS is not available, setting tight navigation requirements becomes costly due to the high cost of deep space tracking - an hour of range and Doppler data using NASA's Deep Space Network (DSN) is in the neighborhood of $4000. Balancing fuel cost and navigation cost can involve a lengthy back-and-forth process between the mission design and navigation teams, e.g. sweeping through the parameter space of navigation $3\sigma$ values at various points in the mission until an acceptable $\Delta V99$ is reached. This method can be costly both in terms of human labor and cloud compute costs for very large, high fidelity Monte-Carlo simulations. This project rephrases this problem as an uncertain decision-making scenario: **when should a spacecraft choose to obtain more tracking data versus using its current state distribution to perform maneuver planning, to balance minimizing tracking time and minimizing $\Delta V99$?**

## II. Background

### A. $\Delta V99$ *and Mission Design*

The computation process to compute $\Delta V99$ figures usually consists of a coupled navigation and maneuver planning Monte-Carlo simulation involving two simulated spacecraft: the *truth spacecraft* and the *navigation spacecraft*. The truth spacecraft is used to generate simulated measurements, which are filtered to provide states with which to simulate the navigation spacecraft. The navigation spacecraft's state estimate is used to design a simulated maneuver. The simulated maneuver is then applied to the truth spacecraft with additional maneuver magnitude and pointing errors, and further measurements are simulated. In each individual Monte-Carlo run, a new truth initial state is drawn from the apriori initial state distribution, and maneuver execution errors are drawn each time a maneuver occurs, to yield a single $\Delta V$ value for that run. The $\Delta V99$ is then the 99th percentile $\Delta V$ over all Monte-Carlo runs. Critically, the process described assumes navigation performance: measurement noise values are dictated by the dish sizes, frequencies, and other parameters chosen before the simulation, and the measurement schedule is also assumed before the Monte-Carlo simulation begins. If these assumptions yield an excessive $\Delta V99$, more tracking time may be added to tighten up the state uncertainties and the entire process repeats to get a new $\Delta V99$. Since there are many sources of uncertainty, a very large number of runs must be performed to yield an accurate $\Delta V$ distribution, and each run is a very high-fidelity dynamics simulation. The size of this Monte-Carlo usually necessitates large amounts of cloud compute, making this stage of the mission design process much more expensive in computation costs than the rest of the whole reference trajectory generation cycle.

Additionally, human navigator and mission designer expertise is critical in this step to effectively heuristicaly determine the best times to obtain tracking data. For instance, in the 9:2 Near Rectilinear Halo Orbit (NRHO) currently in use by CAPSTONE and planned for NASA's Gateway, extensive navigation studies with very large Monte-Carlos were run to determine the best locations in the NRHO to obtain tracking data [1]. Furthermore, dynamical considerations may dictate where maneuvers are most effective: while a particular maneuver placement may yield a very low reference $\Delta V99$, dynamical sensitivity in that region may mean a much higher $\Delta V99$ due to the associated uncertainty growth, compared to a less optimal reference $\Delta V$ that could ultimately have a lower $\Delta V99$ simply because of less dynamical sensitivity.

### *B. Hohmann Transfers*

This scenario is ripe for reformulation as a decision making problem where the spacecraft - either simulated or perhaps actual - optimally decides whether it is more beneficial to obtain more tracking data, tightening its state distribution and lowering $\Delta V99$ before planning, or whether to simply plan and apply a maneuver based on its current state distribution, all while balancing fuel costs with tracking costs. In this project, a significantly simpler situation is posed that captures the broad concepts to serve as a preliminary investigation. Instead of a deep space or cislunar scenario, an orbital transfer between two coplanar circular orbits under a point-mass gravity model is considered where a spacecraft must transfer from an initial LEO parking orbit with radius $6478.0km$. to a geosynchronous orbit with radius $42164km$. With perfect state knowledge, the optimal solution to this problem is a Hohmann transfer, depicted in Figure 1.

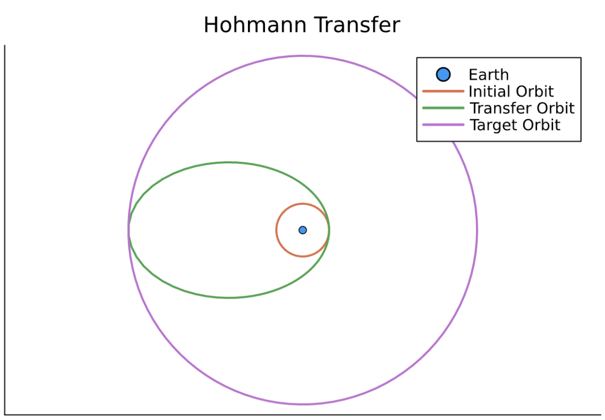


Fig. 1. Hohmann transfer between circular LEO and geosynchronous orbits.

A hohmann transfer between two circular orbits consists of two tangential maneuvers. In this case, since the initial orbit is smaller than the final orbit, the first maneuver boosts the initial orbit into a transfer orbit with an apoapse to the final orbit's radius, and the second maneuver boosts the transfer orbit's periapse up to the target orbit's radius, circularizing it. This process is is depicted below.

To compute the first $\Delta V$, we simply subtract the initial orbit's velocity from the transfer orbit's periapse (apoapse) velocity if the final orbit is larger (smaller) than the initial orbit. Similarly, to obtain the second $\Delta V$, we subtract the transfer orbit's apoapse (periapse) velocity from the target orbit's circular velocity if the target is larger (smaller) than the initial orbit. In the circular case, the maneuvers are calculated as follows:

$$\Delta V_1 = \sqrt{\frac{\mu}{r_1}}\left(\sqrt{\frac{2r_2}{r_1+r_2}}-1\right) \tag{1}$$

$$\Delta V_1 = \sqrt{\frac{\mu}{r_2}}\left(1-\sqrt{\frac{2r_1}{r_1+r_2}}\right) \tag{2}$$

When either the initial or final orbits are elliptical instead of circular, the same process applies, but the appropriate periapse or apoapse velocities must be used for the initial/target orbits according to the problem. Though this is a simple process, enumerating all the branches to obtain the generalized algorithm for elliptical orbit Hohmann transfers is tedious (for instance, we can depart the initial orbit at either periapse or apoapse, and reach the final orbit at its periapse or apoapse), so we assume that we always depart the transfer orbit at its periapse and that the target orbit is circular. Under these assumptions, we denote the initial orbit's semimajor axis ($a$) and eccentricity ($e$), $a_1$, $e_1$, and the target orbit's $a$ and $e$ $a_2$, $e_2$. Then, the initial orbit's periapse radius is $r_{p,1} = a_1(1-e_1)$, and its apoapse radius is $r_{a,1} = a_1(1+e_1)$. This implies the transfer orbit's initial radius is $r_{p,1}$ and that its final radius is $r_t = a_2$ (the target orbit's radius). Using these values, we can calculate the transfer orbit's semimajor axis and eccentricity:

$$a_{transfer} = \frac{r_{transfer,initial} + r_{transfer,final}}{2} \tag{3}$$

$$\tag{4}$$

$$e_{transfer} = \frac{r_{transfer,final} - r_{transfer,initial}}{r_{transfer,final} + r_{transfer,initial}} \tag{5}$$

The first maneuver is then computed as

$$\Delta V_1 = \sqrt{\frac{2\mu}{r_{transfer,initial}} - \frac{\mu}{a_{transfer}}} - \sqrt{\frac{2\mu}{r_{p,1}} - \frac{\mu}{a_1}} \tag{6}$$

while the second is

$$\Delta V_2 = \sqrt{\frac{\mu}{r_t}} - \sqrt{\frac{2\mu}{r_{transfer,final}} - \frac{\mu}{a_{transfer}}} \tag{7}$$

### *C. State Errors*

When a set of maneuvers is planned using the spacecraft's knowledge of the state and applied to the actual state, the result will not be ideal, and corrections will need to be performed to reach the intended target. An exaggerated example of this is shown in Figure 2, where a maneuver is planned assuming an initial radius off by 600km from its true value. It reaches a different orbit than the target, and must perform another set of maneuvers to reach the true target. In actuality, this process may need to be performed multiple times.

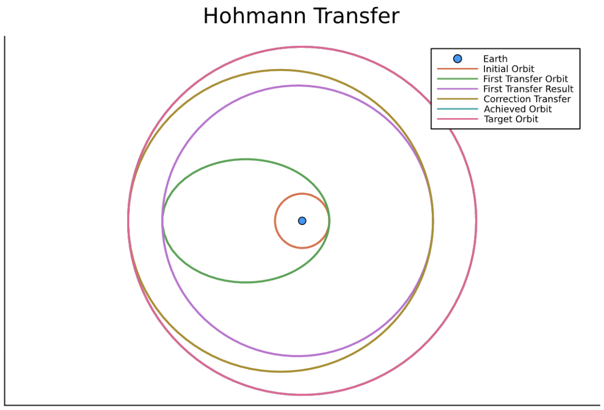


Fig. 2. Hohmann transfer with corrections.

## III. Problem Formulation

### A. General Problem Formulation

At a high level, the problem formulation here is a simplified representation of transferring from a low Earth orbit to a geosynchronous orbit to maximize a reward constructed from tracking time, $\Delta V99$ cost, and the operational costs associated with these actions. The state space $S$ represents the orbital state and is continuous, the observation space yields information about the spacecraft uncertainty, and is also continuous, and the action space $A$ is discrete, allowing a choice between tracking, waiting, or performing maneuvers.

While others have applied a POMDP formulation to sensor tasking for space applications [2] [3], to the author's knowledge, there is no existing work examining this approach to $\Delta V99$ analyses. Examining the $\Delta V99$ instead of the *expected* $\Delta V$ within the reward is similar to risk-aware methods which examine the tails of the Q function [5], but realistic representation of this setup requires the "target" action to have access to the belief $b$ to plan maneuvers based on the current state estimate, and the POMDPs.jl package does not support this framework, necessitating a slightly unorthodox approach. Instead of the belief being used in the discrete "target" action, the first element of the state representing the spacecraft orbit is assumed to be distributed normally about a fixed mean with a standard deviation $\sigma$. The POMDP state is augmented with this $\sigma$ value, and samples are drawn from this distribution during the planning process. The belief updater receives observations of this $\sigma$ value, providing a simplified effect similar to something like a Kalman filter which would be used in real spacecraft operations to update state estimates and reduce uncertainty.

### B. Proof-of-Concept: One-Step Problem Formulation

The problem is posed as a Partially Observable Markov Decision Process (POMDP). The spacecraft is assumed to begin in a circular orbit, characterized entirely by an initial scalar radius $r_i$, and its goal is to achieve a target final circular orbit with radius $r_f$. The actions that can be taken are either "track" or "target." The "track" action gives an observation that improves knowledge of the state distribution, while the "target" action plans maneuvers to reach the target orbit; the "target" action always transitions to the target (terminal) state.

Observations should function to improve the state estimate. In this problem formulation, when the "track" action is taken, the observation consists of a scalar $h\sigma \in \mathbb{R}$, $h \in (0,1)$, allowing the belief value of $\sigma$ to get smaller as more observations are taken, in turn implying the Hohmann transfer maneuvers are performed using a value $r_{i,plan}$ closer to the true value state $r_i$. When the "transfer" action is taken, the observation is simply $\sigma$, effectively meaning no new information is gained.

The one-step problem POMDP is summarized below

- $S : \{r \in \mathbb{R}, \sigma \in \mathbb{R}\}$
- $A : \{track, transfer\}$
- $O : \{\sigma_o \in \mathbb{R}\}$

The transitions are defined **deterministically** according to:

$$T : \begin{cases} s_{initial} \to s_{target} & a = target \\ s_{initial} \to [s_{1,initial}, 0.2 s_{2,initial}] & a = track \end{cases} \tag{8}$$

while the rewards are defined via:

$$R : \begin{cases} -\Delta V_{1,correction} - \Delta V_{2,correction} - 0.1 & a = target \\ -0.5 & a = track \end{cases} \tag{9}$$

and, finally, observations are defined by:

$$Z : o = \sigma' \tag{10}$$

### C. N-Step Transfer Problem Formulation

This formulation of the problem is similar to the one-step version, but the problem is not guaranteed to end at the target orbit after one "transfer" action. Instead, after a "transfer" action, the state is simply whatever orbit the applied maneuvers result in, and the state is now 3 dimensional, consisting of the semimajor axis, eccentricity, and standard deviation. From each state, another action may be taken. Additionally, a "wait" action is added, which doesn't alter the orbit, but has a 50% chance of increasing the uncertainty by a factor of 1.1, and a 50% chance of decreasing it by a factor of 0.8. This is a rough approximation of the possibility of an orbit's uncertainty naturally shrinking if in certain dynamics regions. The problem terminates when the state is within 5.0km and 1e-4 of the target orbit's semimajor axis and eccentricity, respectively, resulting in a terminal reward of $+5$. The "track" action has a negative reward in the range of $-0.5$, the "wait" action has a negative reward in the range of $-0.1$, and the "target" action has a negative reward equal to the 99th percentile of $\Delta V$, computed via 500 Monte-Carlo runs drawing from the state's $\sigma$ value, as well as a small additional negative reward of $-0.1$ to represent the operational cost of planning a maneuver.

The N-step problem POMDP is summarized below

- $S : \{a \in \mathbb{R}, e \in \mathbb{R}, \sigma \in \mathbb{R}\}$
- $A : \{track, transfer, wait\}$
- $O : \{\sigma_o \in \mathbb{R}\}$

The transitions are deterministic (except for the $\sigma$ transition in the "wait" action case), with state transitions due to maneuvers computed via Hohmann transfer maneuvers:

$$T: \begin{cases} a, e \rightarrow a', e' \text{ (Hohmann)}, \sigma \rightarrow 1.1\sigma & a = target \\ a, e \rightarrow a, e, \sigma \rightarrow h\sigma, h \in (0, 1) & a = track \\ a, e \rightarrow a, e, & \\ \quad p(\sigma \rightarrow 1.1\sigma) = 0.5, & \\ \quad p(\sigma \rightarrow 0.8\sigma) = 0.5 & a = wait \end{cases} \tag{11}$$

while the reward is defined via:

$$R: \begin{cases} -(\Delta V_1 - \Delta V_2)_{99} - 0.1 & a = target, \\ & s' \text{ not terminal} \\ -(\Delta V_1 - \Delta V_2)_{99} - 0.1 + 5.0 & a = target, \\ & s' \text{ terminal} \\ -0.5 & a = track \\ -0.1 & a = wait \end{cases} \tag{12}$$

where the $_{99}$ indicates that these quantities are the 99th percentile over a Monte Carlo sim. In each run, an initial $a$ was sampled according to the standard deviation $\sigma$ contained in the state, and maneuvers were planned based on this value. A large uncertainty will yield a high 99th percentile of $\Delta V$, while a smaller $\sigma$ will have a tighter spread of $\Delta V$ cost, meaning a lower 99th percentile cost.

Finally, the observations are deterministic according to:

$$Z : o = \sigma' \tag{13}$$

The discount factor $\gamma$ was set to $0.95$.

## IV. Solution Approach: POMCPOW

POMCPOW [4] is used here to solve the POMDP since it is capable of handling continuous state spaces. This algorithm was not implemented from scratch; the Julia package POMCPOW.jl was instead used to solve the orbital transfer POMDP. POMCPOW (Partially Observable Monte Carlo Planning with Observation Widening) is a POMDP solver algorithm capable of handling continuous state, observation, and action spaces. At a high level, it is a modification of POMCP utilizing *double progressive widening* - a method of gradually expanding the state and action nodes in the POMCP tree to avoid an infinitely wide shallow tree - and expanding weighted particle belief updates to avoid collapsing to a QMDP solution.

## V. Results

In all cases, the POMCPOW solver was used with a UCB (Upper Confidence Bound) selection criterion with $c = 25.0$.

### A. One-Step Transfer Problem

The one-step problem was used as a proof-of-concept testbed. In this case, the POMCPOW policy was able to achieve a marked improvement over a heuristic policy which always performs the "transfer" action, as shown in Table I.

| Policy | Average Reward |
|---|---|
| Transfer | -0.30 |
| POMCPOW | -0.16 |

TABLE I
One-Step Problem Results

### B. N-Step Transfer Problem

For the N-step transfer, POMCPOW was tested against two heuristic policies across a variety of problem parameters. The first heuristic always performs the "transfer" action (the Transfer Policy), while the second performs "track" until the standard deviation $\sigma < 2.0$, and then performs the "transfer" action (the TrackTransfer Policy).

The nominal case is defined via the parameters listed in Table II, and a histogram of rewards for the two heuristic policies as well as the POMCPOW policy is shown in Figure 3. Note that the large negative reward tail in the Transfer Policy is caused by a large negative reward of -100.0 assigned to cases resulting in excessively large $\Delta V$s which result in nonphysical behavior.

| Parameter | Value | Description |
|---|---|---|
| $\sigma_0$ | 50.0 | Initial SMA $\sigma$ |
| $\epsilon_r$ | 5.0 | Convergence criteria for SMA |
| $\epsilon_e$ | 1e-3 | Convergence criteria for ECC |
| $h$ | 0.2 | Reduction in $\sigma$ when tracking |
| $R_{track}$ | -0.5 | Reward for track action |
| $R_{transfer,fixed}$ | -0.1 | Reward for transfer action (in addition to $\Delta V$ |
| $R_{wait}$ | -0.1 | Reward for wait action |
| $R_{terminal}$ | 5.0 | Terminal reward |
| max_depth | 50 | POMCPOW max depth |
| $c$ | 25.0 | POMCPOW UCB parameter |

TABLE II
Nominal Problem Parameters

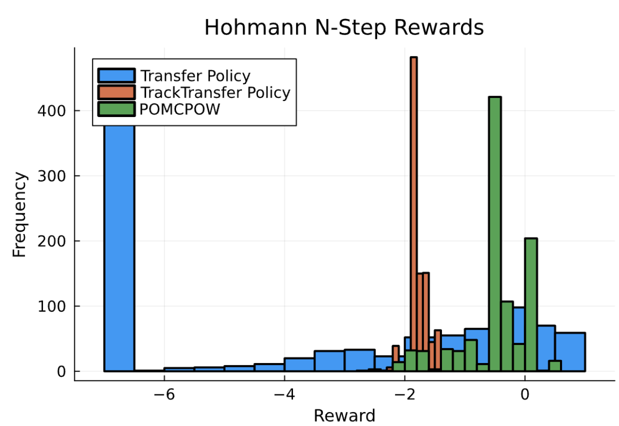


Fig. 3. Nominal reward histogram for heuristic and POMCPOW policies.

Additional cases were run with various changes to the nominal problem parameters, and these results are listed in Table III.

In all of these cases, the POMCPOW policy was able to achieve better results than the heuristic policies. Although this problem formulation is still quite a long way off from

| | Average Reward For Policy | | |
|---|---|---|---|
| Setup Description | Transfer | TrackTransfer | POMCPOW |
| Nominal | -3.76 | -1.77 | -0.38 |
| $h = 0.5$ | -3.75 | -1.77 | -1.30 |
| $R_{track} = -0.1$ | -3.69 | -1.78 | 0.42 |
| $R_{track} = -1.0$ | -3.82 | -1.77 | -1.11 |
| $\epsilon_r = 1.0$ | -6.13 | -1.85 | -0.99 |

TABLE III
N-STEP PROBLEM AVERAGE REWARDS

a sufficiently high-fidelity model, these results are promising. This decision-making approach combined with a robust solver yields good results, and requires much less human analysis time than the typical approach, which would be equivalent to sweeping through various heuristic policies.

## VI. CONCLUSION

A POMDP was formulated to represent the mission design problem of deciding whether a spacecraft should gather tracking data to reduce its state uncertainty or immediately plan a maneuver to reduce $\Delta V99$ while avoiding excessive expensive tracking time. In the simplified model used here, a spacecraft must transfer from a low Earth orbit to a coplanar, coaxial, geosynchronous orbit using a series of Hohmann transfer maneuvers. At each step, the spacecraft can either transfer, track, or wait. POMCPOW proved effective in most cases compared to heuristic policies, but struggled in some scenarios when the cost of getting tracks was high. The results seem to be promising and suggest that (with extensive improvements) this decision-making approach could be a fruitful alternative to the traditional mission design method of using $\Delta V99$ studies to determine fuel budget and navigation requirements.

## VII. FUTURE WORK

The problem formulation presented here was drastically simplified to fit within the allotted time frame and be computationally feasible. The fidelity of the problem setup should be significantly enhanced to more accurately represent reality. This could be done in multiple stages. Firstly, uncertainty evolution over time via dynamical proccesses should be implemented, which would make the uncertainty information significantly more realistic. Next, an integrator should be used to propagate orbits based on dynamics equations instead of using two-body point-mass assumptions to transition between apsis states. Actions could then be taken every hour or minute instead of being constrained to occur at apses. The result of these enhancements would be a much more informative analysis of the effectiveness of this approach to the mission design process. Finally, the scenario could be changed to the cislunar regime to assess how results change with chaotic dynamics.

## VIII. CONTRIBUTIONS AND RELEASE

Sai Chikine performed all of the tasks described here. The authors grant permission for this report to be posted publicly.

## REFERENCES


[1] E.W. Kayser, M.R. Thompson, M. Bolliger, N.P. Re, D.C. Davis, M.L. McGuire, S.L. McCarty, "Navigation and Mission Design for Low-Thrust Insertion into Near Rectilinear Halo Orbits," Presented at AIAA SCITECH 2022 Forum, California, USA, Jan. 2022

[2] S.J. Fedeler, M.J. Holzinger, W. Whitacre, "Optimality and Application of Tree Search Methods for POMDP-based Sensor Tasking," AMOS 2020 Conf.

[3] P.M. Siew, R. Linares, "Optimal Tasking of Ground-Based Sensors for Space Situational Awareness Using Deep Reinforcement Learning," in Sensors, vol. 20, pp. 7847, 2022

[4] Z. Sunberg, M. Kochenderfer, "Online algorithms for POMDPs with continuous state, action, and observation spaces," arXiv:1709.06196 [cs.AI], Sep. 2018

[5] T. Theate, D. Ernst, "Risk-Sensitive Policy with Distributional Reinforcement Learning" arXiv:2212.14743 [cs.LG], Dec. 2022