

Coordinated Motion Planning for On-Orbit Satellite Inspection using a Swarm of Small-Spacecraft

Benjamin Bernhard
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
bbernhar@nd.edu

Soon-Jo Chung
Graduate Aerospace Laboratories
California Institute of Technology
Pasadena, CA 91109
sjchung@caltech.edu

Changrak Choi
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
changrak.choi@jpl.nasa.gov

Amir Rahmani
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
amir.rahamani@jpl.nasa.gov

Fred Hadaegh
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
fred.y.hadaegh@jpl.nasa.gov

Abstract— This paper addresses the problem of how to plan optimal motion for a swarm of on-orbit servicing (OOS) small-spacecraft remotely inspecting a non-cooperative client spacecraft in Earth orbit. With the goal being to maximize the information gathered from the coordinated inspection, we present an integrated motion planning methodology that is a) fuel-efficient to ensure extended operation time and b) computationally-tractable to make possible on-board re-planning for improved exploration. Our method is decoupled into first offline selection of optimal orbits, followed by online coordinated attitude planning. In the orbit selection stage, we numerically evaluate the upper and lower bounds of the information gain for a discretized set of passive relative orbits (PRO). The algorithm then sequentially assigns orbits to each spacecraft using greedy heuristics. For the attitude planning stage, we propose a dynamic programming (DP) based attitude planner capable of addressing vehicle and sensor constraints such as attitude control system specifications, sensor field of view, sensing duration, and sensing angle. Finally, we validate the performance of the proposed algorithms through simulation of a design reference mission involving 3U CubeSats inspecting a satellite in low Earth orbit.



Figure 1: Spacecraft swarms can provide complete and frequent sensor coverage of a client satellite in Earth’s orbit, presenting significant advantages over single-spacecraft remote inspection. [S/C model credit: Northrop Grumman, EnduroSat]

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. PROBLEM STATEMENT	2
3. MOTION PLANNING SYSTEM OVERVIEW.....	3
4. SWARM ORBIT DESIGN	4
5. ATTITUDE PLANNING	6
6. SIMULATION RESULTS	7
7. CONCLUSION	11
REFERENCES	11
BIOGRAPHY	12

1. INTRODUCTION

On-orbit servicing (OOS) technology has been rapidly advancing in recent years as the growing commercial space sector has developed a market for services that extend the lifetime of satellites in Earth orbit. These services include the use of at least one servicing spacecraft for inspecting, repairing, refuelling, assembling, or upgrading a client (or target) satellite in Earth orbit [1]. Most on-orbit servicing concepts involve a single servicing spacecraft. The first demonstration of this technology was the DARPA Orbital Ex-

press in 2007. During this mission, the servicing spacecraft ASTRO (Autonomous Space Transfer and Robotic Orbiter) performed autonomous rendezvous, capture and berthing, propellant and electronics transfer, and visual inspection of the client satellite, NextSat [2]. Recent developments in OOS include a Northrop Grumman servicing spacecraft, called Mission Extension Vehicle (MEV), that launched in October of 2019. MEV is designed to dock with a client satellite and provide orbital station-keeping and attitude control [3]. More recently, NASA Johnson Space Center demonstrated single spacecraft remote inspection of the Cygnus spacecraft during the Seeker-1 technology demonstration in September of 2019 [4]. Likewise, many other single-spacecraft OOS capabilities are being developed and are detailed in [1, 5] and the references therein. However, while single-spacecraft OOS technology has been successful, utilizing a swarm (or team) of small-spacecraft has the potential to provide more capable and time-effective on-orbit services.

There are many OOS capabilities that spacecraft swarms may enable, such as autonomous on-orbit assembly [6, 7] or collaborative manipulation [8]. Yet, in this paper, we focus on the prospect of using a swarm of small-spacecraft for on-orbit inspection, an essential first step towards enabling future multi-spacecraft OOS capabilities. The benefits of using swarms for on-orbit inspection are analogous to the benefits associated with using multi-spacecraft systems for

Earth observation. Using multiple satellites allows for complete and sustainable distributed sensor coverage of the client with frequent updates. This removes the need to expend fuel for orbit transfers and provides a means for monitoring the client satellite during time-sensitive situations. Moreover, with the use of heterogeneous swarms, multiple types of data can be collected simultaneously at different locations of the client. To realize these benefits, we present a set of integrated algorithms for optimal orbit selection and real-time attitude planning that maximize the amount of scientific data collected by the swarm while minimizing control effort, ensuring collision avoidance, and reacting to new areas of interest on the target.

Prior Work

For Earth-targeted remote sensing and other applications, distributed small-spacecraft systems are already being implemented to create unique sensing capabilities, as detailed in [9]. Furthermore, improved Earth remote sensing techniques using multiple spacecraft are being developed that take advantage of the agility and accuracy of new small-spacecraft attitude control systems (ACS). In [10], small-satellites are used in formation flight to make multi-angle observations of aerosols by directing multiple satellites to point at the same location at the same time. In [11] and [12], the authors develop an algorithm for scheduling attitude slew manoeuvres, such that the number of unique images taken by each satellite is maximized. These proposed algorithms are reliant upon modern ACS technology for small-satellites. As reported by NASA Ames Research Center in 2018, state-of-the-art ACS are small enough and accurate enough to be used in CubeSats to provide efficient slewing and pointing [13]. In a notable demonstration of the performance of these control systems, a 6U CubeSat called ASTERIA used the Blue Canyon Technologies flexIble Attitude Control Technology (XACT) to achieve pointing accuracy that drifted only 20 arcseconds over a 20-minute observation [14].

A swarm of distributed small-spacecraft presents the opportunity to provide enhanced OOS that is highly re-configurable, scalable, resilient to failure, and low-cost beyond its monolithic counterpart [15]. The multi-agent autonomy technologies required are maturing quickly, and a review paper on aerial swarm robotics [16] provides an overview of the state-of-the-art autonomy technologies that enable swarm operation. Furthermore, NASA is pushing towards augmenting OOS capabilities [17] with novel space system architectures such as SWIFT [18], a proposed architecture that would enable small-scale (100g) spacecraft to be flown as swarms.

However, the benefits of swarm operation come with the challenge of coordinating the motion of a large number of agents to collectively achieve a given objective [15]. Multi-agent motion planning is fundamentally a PSPACE-hard problem [19] that can quickly become NEXP when system dynamics are complex [20]. A recent survey paper [9, 21] that surveyed 39 multi-agent missions composed of small satellites shows that missions involving closely coordinated formation flying are limited to two or three satellites. This indicates the inherent difficulty and technology gap that exists in planning coordinated motion for a large swarm of spacecraft. Recent work [22–25] addresses this challenge by utilizing passive relative orbits (PROs) and sequential convex programming (SCP) to provide collision-free, re-configurable formation flying for a swarm of spacecraft. This paper presents related work that contributes to the technology gap with specific focus on coordinated inspection of a target object using a swarm of spacecraft.

The paper is organized as follows: The problem statement that defines the problem objective, definitions, and notation is given in Section 2. An overview of the coordinated motion planning architecture is presented in Section 3. Then, optimal orbit design for a swarm of spacecraft is detailed in Section 4, followed by a description of a dynamic programming based attitude planner in Section 5. Finally, we show numerical simulation results for the design reference mission in Section 6, with a conclusion in Section 7.

2. PROBLEM STATEMENT

Our objective is to develop a collaborative swarm motion planning methodology that autonomously plans optimal motion for a swarm (or team) of small-spacecraft performing an on-orbit inspection. The planned motion should maximize information collectively gathered from the coordinated inspection while minimizing the aggregated fuel cost and ensuring collision avoidance. The planner should be computationally efficient to make possible real-time, on-board computation and should be suitable for re-planning with updates from estimation. In this paper, we adopt the following definitions and assumptions for each component.

Definitions and Assumptions

Target Object: Let X_o be the state of the target object under inspection. We assume that the target object is undergoing pure rotation under its principal axis, and its motion $X_o(t)$ is known. Let M_o be the surface of the target object and m_i be a surface partition of the target object s.t. $m_i \subset M_o$, $\bigcup m_i = M_o$, $m_i \cap m_j = \emptyset$. We assume that geometry M_o is known prior, with its surface mesh given as partitions m_i .

Spacecraft: Let S be a swarm of on-orbit servicing spacecraft with cardinality $|S| = N_{sc}$. Let $s \in S$ be a spacecraft in the swarm, with x_s representing the state of the spacecraft. We assume that each spacecraft is homogeneous in size and actuation (but may differ in sensing capability). It is assumed that the spacecraft is capable of producing thrusting force in a desired direction, but total delta-v budget is heavily constrained. In contrast, we assume the spacecraft can orient itself at relatively low-cost to point a sensor in a desired direction using reaction wheels.

Attitude Control System: Let ω_{slew} be the slew rate achievable by the attitude control system (ACS), and τ_{settle} be the time it takes for the ACS to settle at a given pointing direction after a slew maneuver. The design of the attitude control system itself is outside the scope of this work, so we make the simplifying assumption that ω_{slew} and τ_{settle} are fixed parameters.

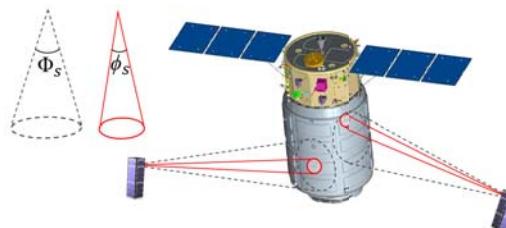


Figure 2: A spacecraft’s pointing direction is bounded by the maximum off-nadir angle to account for sensor distortion at large angles of incidence.

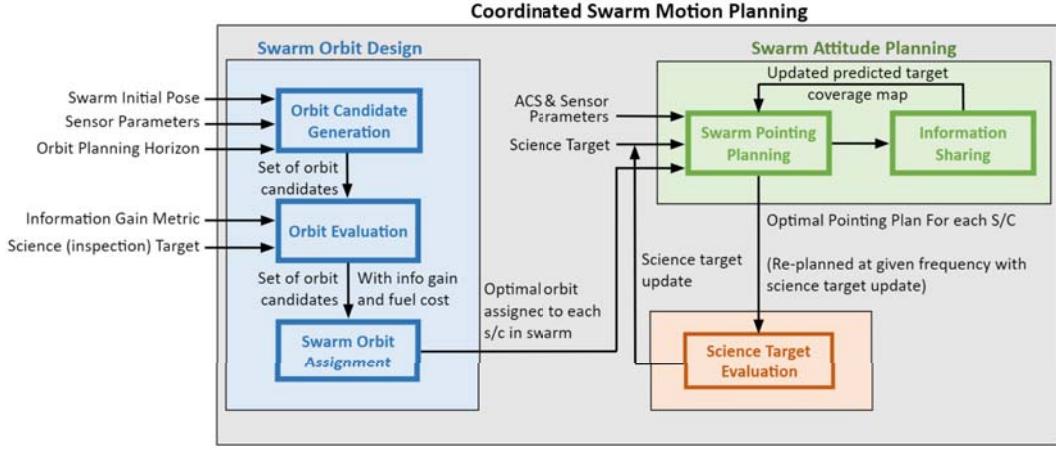


Figure 3: The proposed motion planning approach is decomposed into 1) a swarm orbit design stage that selects optimal orbits for a team of spacecraft, and 2) an optimal attitude planning stage that coordinates sensor pointing of multiple spacecraft to maximize the collective information gain from the inspection.

Sensing: Let d_s be the maximum sensing distance and T_{sense} be the minimum time required for sensing to be effective for the sensor on the spacecraft $s \in S$. Let ϕ_s define the conical field of view (FoV) of the sensor and θ_s be the maximum allowed deviation angle from the surface normal. This establishes a conical FoV boundary with angle Φ_s that is illustrated in Fig. 2. For each sensor, we assume d_s , T_{sense} , ϕ_s , and θ_s are specified based on the desired sensing resolution and sensor characteristics. Last, we assume that the sensor is fixed relative to the spacecraft.

Coverage: Let $C(m_i) \in \{0, 1\}$ be coverage index of the target object's i -th surface partition. $C(m_i) = 1$ if the surface partition m_i is *effectively viewed* by a spacecraft's sensor for more than T_{sense} consecutively and $C(m_i) = 0$ otherwise. A surface partition m_i is *effectively viewed* if m_i is within the FoV of the sensor, and the angle between the sensor pointing direction and surface normal of m_i is within θ_s .

Science Target: Let $S(m_i) \in \{0, 1\}$ be the science index of i -th surface partition m_i of the target object surface M_o . $S(m_i) = 1$ if the surface partition m_i is categorized as a science target of interest and $S(m_i) = 0$ otherwise. Science targets represent areas of the target object that hold higher information value and need to be prioritized for inspection (e.g., a valve that is suspected to be malfunctioning). We assume science targets are specified and frequently updated online by a human operator evaluating the mission data.

Information Gain: Let $C_{u-}(m_i)$ and $C_{u+}(m_i)$ be the coverage index before and after an inspection action u is performed. Science gain $SG(u)$ and coverage gain $CG(u)$ from the inspection action u are defined as

$$SG(u) = \frac{\sum_i S(m_i)[C_{u+}(m_i) - C_{u-}(m_i)]}{\sum_i S(m_i)} \quad (1)$$

$$CG(u) = \frac{\sum_i A(m_i)[C_{u+}(m_i) - C_{u-}(m_i)]}{\sum_i A(m_i)} \quad (2)$$

where $A(m_i)$ is area of the surface partition m_i . Information gain $IG(u)$ from an inspection action u is defined as

$$IG(u) = \alpha SG(u) + (1 - \alpha) CG(u) \quad (3)$$

where $\alpha \in [0, 1]$ is a parameter that balances exploitation and exploration represented by the science gain and coverage gain, respectively. Note that science gain $SG(u)$, coverage gain $CG(u)$, and information gain $IG(u)$ are normalized indices that take on values between 0 and 1.

Based on the above definitions and assumptions, we develop a collaborative swarm motion planning methodology that achieves the described problem objective.

3. MOTION PLANNING SYSTEM OVERVIEW

Achieving fuel efficient inspection requires multiple servicing spacecraft to cooperatively explore all areas of the target object and exploit areas of high interest through limited orbit and attitude adjustments. Considering the high cost of orbit transfer using thrusters and the relatively low cost of attitude control using reaction wheels, our motion planning approach encompasses performing a single orbit transfer for each spacecraft in the swarm to a set of designed orbits around the target object and then continuously adjusting attitude to follow planned pointing directions to maximize the information gain. To this end, our motion planning methodology is decoupled into first selection of optimal orbits for a team of servicing spacecraft, followed by coordinated attitude planning based on the optimal orbits selected, as shown in Fig. 3.

In the orbit design stage, an optimal set of orbits for a swarm of servicing-spacecraft is designed that collectively maximizes information gain in a fuel-efficient and collision-free manner. First, a set of orbit candidates, OC , is generated from the specified sensing parameters (d_s , ϕ_s , θ_s) and planning horizon, which represents the total duration of coordinated inspection. Then each orbit $o_{i,t} \in OC$ is evaluated for the information gain from the orbit and fuel cost of transferring the spacecraft from its initial pose to the orbit. Based on the evaluation, an optimal orbit is assigned to each spacecraft in the swarm.

In the coordinated attitude planning stage, an optimal attitude plan is generated that guides the pointing direction of each spacecraft on its assigned orbit to collectively maximize information gain while minimizing actuation cost. The attitude planner takes into account sensing constraints (d_s , ϕ_s , θ_s , T_{sense}), attitude control system specifications (ω_{slew} , τ_{settle}), and current science targets and generates a finite horizon attitude plan. The science targets are updated online by a human operator evaluating the data from the inspection, and re-planning is performed at a given frequency to take into account the changes in the science target for improved inspection.

The motion planning algorithms strive towards optimality to maximize information gain and computational efficiency to make on-board computation of the plan possible. However, there exists a trade-off between the optimality and computational efficiency that makes it difficult to achieve both at the same time. This is especially true for swarm motion planning, which is a PSPACE-hard problem [19] that suffers from the curse of dimensionality due to sheer size of the state-space that increases with the number of agents involved. In our motion planner, we leverage upon a multi-agent prioritized planning approach [26, 27] that treats multi-agent planning as a sequence of single-agent planning with problem-specific heuristics governing the order. This approach has no guarantee of global optimality but can generate near-optimal solutions with computation time that is quick and scales linearly with the number of spacecraft in the swarm.

4. SWARM ORBIT DESIGN

Algorithm 1: Swarm Orbit Design

```

1  $x_{j,0}$  = initial position of j-th spacecraft
2  $n_{orbit}$  = total number of orbit candidates
3  $o_{i,t}$  = trajectory of i-th orbit candidate
4  $c_{j,i}$  = cost of orbit transfer from j-th spacecraft to i-th orbit
5  $N_{sc}$  = total number of spacecraft of the swarm
6  $s_i$  = i-th spacecraft in the swarm
7 for  $i=1$  to  $n_{orbit}$  do
8    $o_{i,0} \leftarrow \text{selectOrbitIC}(d_{req})$ ;
9    $o_{i,t} \leftarrow \text{generateOrbitTraj}(o_{i,0}, t_{plan})$ ;
10   $I\hat{G}_i \leftarrow \text{estimateInformationGain}(o_{i,t})$ ;
11  for  $j=1$  to  $N_{sc}$  do
12     $| c_{j,i} \leftarrow \text{orbitTransferCost}(x_{j,0}, o_{i,t})$ ;
13  end
14 end
15  $h(k) \leftarrow \text{heuristicOrdering}()$ ;
16 for  $k=1$  to  $N_{sc}$  do
17   $i^* \leftarrow \arg \max_i (I\hat{G}_i / c_{h(k),i})$ ;
18  while  $\neg \text{collisionFree}(o_{i^*,t})$  do
19     $| I\hat{G}_{i^*} = -1$ ;
20     $| i^* \leftarrow \arg \max_i (I\hat{G}_i / c_{h(k),i})$ ;
21  end
22  Assign orbit  $o_{i^*,t}$  to  $s_{h(k)}$ ;
23  for  $m=1$  to  $n_{orbit}$  do
24     $| I\hat{G}_m \leftarrow \text{subtractInformationGain}(I\hat{G}_m, I\hat{G}_{i^*})$ ;
25  end
26 end

```

Orbital Dynamics

Given that the spacecraft performing inspection will be within close proximity of the target object, the Clohessy-Wiltshire-Hill (CWH) equations can provide a good linear approxima-

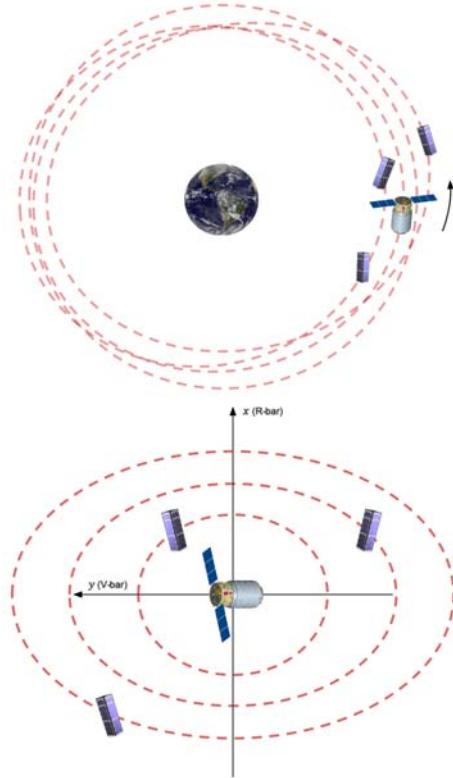


Figure 4: Concentric passive relative orbits (PROs) are utilized to provide a collision-free set of orbits for the swarm of inspecting spacecraft. A small-spacecraft swarm in concentric PROs is shown in the Earth centered frame (above) and the CWH LVLH frame (below).

tion of the relative dynamics between the target object and each agent in the swarm. To use this approximation, it must also be assumed that the target is in a circular orbit and that the Earth is perfectly spherical. The 3-DOF CWH equations, originally developed in [28], describe the motion of a given spacecraft in the Local-Vertical, Local-Horizontal (LVLH) coordinate system. The target object is located at the origin of this coordinate system, where the x -direction (R-bar) points radially away from Earth toward the target, the y -direction (V-bar) points in the direction of the orbital velocity of the target object, and the z -direction points in the direction of the angular momentum of the target. The CWH equations in this coordinate system are given in Eq. (4) as

$$\begin{aligned} \ddot{x} &= 3n^2 x + 2n\dot{y} \\ \ddot{y} &= -2n\dot{x} \\ \ddot{z} &= -n^2 z \end{aligned} \quad (4)$$

The mean motion of the target object, n , is defined in Eq. (5) as

$$n = \sqrt{\frac{\mu}{r_0^3}} \quad (5)$$

where μ is Earth's gravitational constant and r_0 is the radius of the target object's orbit. The closed form solution to the CWH equations is well known and given in Eq. (6).

Careful selection of the initial conditions for each spacecraft in the swarm results in a set of passive relative orbits (PROs).

$$\begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} 4 - 3 \cos nt & 0 & 0 & -\frac{1}{n} \sin nt & \frac{2}{n}(1 - \cos nt) & 0 \\ 6(\sin nt - nt) & 1 & 0 & -\frac{2}{n}(1 - \cos nt) & \frac{1}{n}(4 \sin nt - 3nt) & 0 \\ 0 & 0 & \cos nt & 0 & 0 & \frac{1}{n} \sin nt \\ 3n \sin nt & 0 & 0 & \cos nt & 2 \sin nt & 0 \\ -6n(1 - \cos nt) & 0 & 0 & -2 \sin nt & 4 \cos nt - 3 & 0 \\ 0 & 0 & -n \sin nt & 0 & 0 & \cos nt \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} \quad (6)$$

PROs are thrust-free periodic relative spacecraft trajectories. These orbits are ideal for use in fuel-constrained small-spacecraft that perform inspection around a target object [29, 30].

Orbit Selection

In particular, we utilize concentric PROs that provide a fuel-efficient way of ensuring collision avoidance among the swarm of spacecraft [22]. From [22], it has been shown that setting the initial conditions as those given in Eq. (7) will provide PROs with concentric projections in the x - y plane. Fig. 4 shows an example of three CubeSats following concentric PROs. In this configuration, no spacecraft will collide. When two spacecraft are on different PROs, one orbit will be completely inside or outside of the other orbit. When two spacecraft are on the same PRO, the spacecraft will follow each other with a phase difference. Using linearized concentric PROs, it has been shown that a concentric PRO swarm consisting of 500 spacecraft is nearly collision-free for the first 60 orbits, even with the nonlinear dynamics including J_2 perturbations [22]. For an on-orbit inspecting swarm that will operate with fewer spacecraft and require fewer orbits to achieve full coverage, a set of concentric PROs using Eq. (7) provides fuel-efficient collision-free orbits to perform inspection.

$$\dot{x}_0 = \frac{1}{2}ny_0, \quad \dot{y}_0 = -2nx_0, \quad \dot{z}_0 = 0 \quad (7)$$

In the orbit selection stage, `selectOrbitIC` (Alg.1 line 8) first generates a set of initial conditions corresponding to the total number of orbit candidates (n_{orbit}) to be considered in the orbit design. Different sampling schemes can be employed to select n_{orbit} samples (e.g., regular lattice or random sampling) from the independent variables that suit needs of the orbit design. In this process, initial positions are sampled with a constraint that enforces the resulting PROs to comply with the distance requirement (d_{req}) of the sensor. The initial velocities are then set following Eq. (7) based on the initial positions. The cardinality of orbit candidates, n_{orbit} , is a design parameter that is to be changed depending on whether orbit candidates need to be dense or sparse and whether computation will be done off-line or on-line.

Once a set of initial conditions is generated, `generateOrbitIC` (Alg.1 line 9) then propagates forward each initial condition according to the orbital dynamics to orbit trajectories up to a time horizon set by t_{plan} . These orbit trajectories are stored as $o_{i,t}$ for evaluation.

Orbit Evaluation

In the orbit evaluation stage, each PRO candidate generated in the orbit selection stage is assessed for the Information Gain (IG) it brings and fuel cost of transferring to the particular PRO. First an estimate of information gain from each PRO candidate is computed by `estimateInformationGain` (Alg.1 line 10). Computing an exact information gain from

a PRO requires running the attitude planner on the given orbit trajectory, which is computationally expensive. Instead, upper and lower bounds of information gain are computed to provide an estimate.

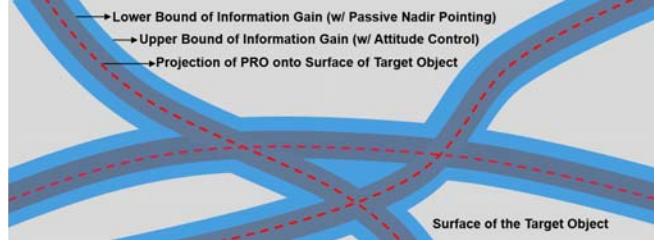


Figure 5: Information gain from each orbit candidate is estimated based on the upper and lower bounds of the information gain, which can be computed efficiently.

As shown in Fig. 5, an orbit trajectory $o_{i,t}$ can be projected down to the surface of the target object, and the area covered from the orbit will form a strip around the projected curve that depends on the field of view and pointing direction of the sensor. A lower bound of the information gain, IG_i^l , is computed assuming that sensor is pointed in the nadir direction throughout the orbit trajectory. An upper bound of the information gain, IG_i^u , is computed assuming that the attitude planner is able to cover all areas between nadir and the maximum angle for effective viewing, ϕ_i . The exact information gain from a given PRO lies in between IG_i^l and IG_i^u . An estimate of information gain, \hat{IG}_i , is set based on the computed upper and lower bounds, with the value nearing to the lower bound if orbit design needs to be conservative. Note that due to the eccentricity, a given PRO may have large variation in its distance to the science target, and coverage can be filtered out when the distance exceeds the maximum sensing distance, d_s .

Once an information gain estimate of each orbit candidate is established, then fuel cost of transferring a spacecraft from its initial pose, $x_{j,0}$, to each orbit candidate, $o_{i,t}$, is computed by `orbitTransferCost` (Alg.1 line 12). Many methodologies exist in the literature that address computation of such orbit transfer [25, 31–34], and an algorithm that suits the orbit design objectives can be adapted. In particular, MPC-SPC [25] provides optimal transfer from one J_2 invariant orbit to another that minimizes fuel-cost in a computationally efficient manner. Note that a broader fuel cost analysis of transferring multiple spacecraft in the mission design [35] can also be utilized.

Orbit Assignment

In the orbit assignment stage, each spacecraft, s_i , in the swarm (of cardinality N_{sc}) is assigned an optimal PRO, $o_{i^*,t}$, from the set of PRO candidates (of cardinality n_{orbit}) evaluated. The objective of the assignment is to generate a set of optimal PROs that collectively maximize the information gain while minimizing the fuel cost and balancing the fuel

budget of the swarm.

The orbit assignment becomes challenging when 1) a large number of spacecraft are involved in a swarm, and 2) there is overlap of coverage from multiple PROs that needs to be singled out. This is a combinatorial optimization problem whose time complexity for an optimal solution scales poorly with the size of the swarm. In our approach, we adopt a prioritized planning methodology [26,27] that favors computational efficiency in a sacrifice of optimality for multi-agent planning.

The orbit assignment is performed sequentially to assign a PRO to each spacecraft in the order given by `heuristicOrdering` (Alg.1 line 15). The heuristic used to decide the sequence in which a spacecraft is assigned to a PRO is based on remaining fuel budget (Δv). The spacecraft that has the lowest fuel budget within the swarm is assigned an orbit first, followed by the spacecraft with next lowest fuel budget and so forth. This heuristic is utilized to balance the fuel budgets of spacecraft in the swarm, ensuring that no one spacecraft becomes non-functional with its fuel exhausted while others have sufficient fuel budget to maneuver. The rationale is that, initially, each PRO candidate carries similar coverage of the target object and hence a spacecraft can be assigned to a PRO that has minimum transfer cost and still has high information gain from it. As assignment continues, each PRO candidate holds vastly different coverage due to overlap of covered area from previously assigned PROs, and a spacecraft may have to be assigned PRO with high transfer cost to have a meaningful information gain from it.

In each assignment step, the orbit $o_{i^*,t}$ that has the maximum ratio of information gain over transfer fuel cost (Alg.1 line 17) is found from the set of orbit candidates. To ensure safety, the selected orbit is then checked for collision using a set buffer distance to the previously assigned set of orbits. Once the orbit $o_{i^*,t}$ is verified to be collision-free, it is assigned to the current spacecraft $s_{h(k)}$ in the heuristic order (Alg.1 line 22). Then, the information gain of each orbit candidate is updated by taking into account the overlap in coverage of the assigned orbit $o_{i^*,t}$ and other orbit candidates (Alg.1 line 24). The assignment continues until each spacecraft in the swarm has an assigned PRO.

5. ATTITUDE PLANNING

The swarm attitude planning stage is designed to act independently of the orbit selection stage. Thus, given any arbitrary set of orbits, the proposed method attempts to optimize each agent's attitude trajectory on the orbit to maximize information gain while minimizing the control effort of the attitude control system. Unlike the orbit selection algorithm, the attitude planning algorithm must be computationally tractable for real time planning on-board each servicing spacecraft. So, we structure our attitude planning algorithm in a similar way to the dynamic programming algorithm proposed for Earth remote sensing in [11] and [12]. That is, we treat each attitude trajectory as a path of pointing directions and use dynamic programming to find the path of highest reward. Our primary contribution to spacecraft pointing planning is the proposed algorithm's ability to satisfy constrained sensing time requirements, consider the trade-off between actuation cost and sensor coverage, and periodically re-plan online for updated science targets.

Dynamic Programming Formulation

We define the global set of pointing options, P , to be the set of vertices of the surface partitions $m_i \in M_o$. At each point in time, we restrict the pointing options of each satellite, $s \in S$, to the local set, $P_{s,t} \subset P$, of points within the sensor's conical FOV limit with angle Φ_s . The discrete time interval is defined to be the sum of travel time and sense time,

$$T_{int} = T_{slew} + T_{sense} \quad (8)$$

where T_{slew} is a fixed value that defines the amount of time in which a spacecraft must slew and settle at the next pointing direction. We define the sequence of pointing directions, $\{p_{t_0}, p_{t_0+T_{int}}, \dots, p_{T_{plan}}\}$, for a spacecraft, s , as the policy π_s . The goal is to find the optimal policy, π_s^* , for each spacecraft. To accomplish this, we utilize the principle of optimality as follows:

$$J_{s,t+T_{int}}^*(j) = \max_i [r(p_{i,t}, p_{j,t+T_{int}}) + J_{s,t}^*(i)] \quad (9)$$

where $J_{s,t}^*(i)$ is the optimal reward up to time t for spacecraft s under policy π_s that ends at the pointing direction $p_{i,t} \in P_{s,t}$. The reward going from $p_{i,t}$ to $p_{j,t+T_{int}}$ is defined as

$$\begin{aligned} r(p_{i,t}, p_{j,t+T_{int}}) = & \beta f(p_{i,t}, p_{j,t+T_{int}}) \\ & - (1 - \beta)g(p_{i,t}, p_{j,t+T_{int}}) \end{aligned} \quad (10)$$

where $f(p_{i,t}, p_{j,t+T_{int}})$ is simply the new information gain acquired from inspection action u_{ij} , as defined in Eqs. (3) and (11). In this case, u_{ij} is the inspection action from time t to $t + T_{int}$ that moves from pointing at p_i and settles at p_j .

$$f(p_{i,t}, p_{j,t+T_{int}}) = IG(u_{ij}) \quad (11)$$

In Eq. (11), we are making an assumption that $P_{s,t_i} \cap P_{s,t_j} = \emptyset$ for all $s \in S$ and $|t_i - t_j| \geq 2T_{int}$. That is, the pointing options $P_{s,t+T_{int}}$ of a satellite s at time $t + T_{int}$ can only overlap with pointing options $P_{s,t}$ at its adjacent time step. This assumption holds when Φ_s is comparably small for the given T_{int} , and the planning time horizon T_{plan} is less than the orbit period. It ensures that the reward function r only depends on the current and previous pointing directions, which is necessary to guarantee optimality. However, even when the assumption does not hold, the DP formulation generates a computationally efficient solution that is near-optimal.

The actuation cost function, $g(p_{i,t}, p_{j,t+T_{int}})$, returns the cost associated with slewing from $p_{i,t}$ to $p_{j,t+T_{int}}$ and maintaining the pointing direction p_j for T_{sense} . We define actuation cost, AC , to simply be the total change in attitude angle associated with changing or maintaining a pointing direction.

$$\begin{aligned} AC(p_{i,t}, p_{j,t+dt}) = & \\ \arctan \frac{\|(x_t - p_{i,t}) \times (x_{t+dt} - p_{j,t+dt})\|}{(x_t - p_{i,t}) \cdot (x_{t+dt} - p_{j,t+dt})} \end{aligned} \quad (12)$$

The cost for attitude keeping is then the case when $i = j$, and the cost for attitude slewing is the case when $i \neq j$. This leads to a definition of the actuation cost function for each combination of transitioning to and inspecting at a given pointing direction.

$$\begin{aligned} g(p_{i,t}, p_{j,t+T_{int}}) = & \\ AC(p_{i,t}, p_{j,t+T_{slew}}) + AC(p_{j,t+T_{slew}}, p_{j,t+T_{int}}) \end{aligned} \quad (13)$$

Algorithm 2: Swarm Attitude Planning

```

1  $x_{s,t_0}$  = initial position of each S/C
2  $P_{s,t_0}$  = set of initial possible pointing directions for each S/C
3  $\Pi_{s,t_0}$  = set of previous optimal policies for each S/C
4  $M_{s,t_0}$  = initial target coverage map according to each S/C
5 while  $t < T_{plan}$  do
6   for  $s \in S$  do
7      $x_{s,t+T_{slew}} \leftarrow \text{orbit}(x_{s,t}, t + T_{slew});$ 
8      $x_{s,t+T_{sense}} \leftarrow \text{orbit}(x_{s,t}, t + T_{sense});$ 
9      $P_{s,t+T_{slew}} \leftarrow \{p \mid p \in \text{range}(x_{s,t+T_{slew}}, t + T_{slew})\};$ 
10     $P_{s,t+T_{slew}+T_{sense}} \leftarrow \{p \mid p \in$ 
11       $\text{range}(x_{s,t+T_{slew}+T_{sense}}, t + T_{slew} + T_{sense})\};$ 
12     $P_{s,t+T_{int}} \leftarrow P_{s,t+T_{slew}} \cap P_{t+T_{slew}+T_{sense}};$ 
13    for  $p_j \in P_{s,t+T_{int}}$  do
14      for  $p_i \in P_{s,t}$  do
15        if  $\text{slew}(p_i, x_{s,t}, p_j, x_{s,t+T_{slew}})$  then
16          |  $r_{ji} \leftarrow \text{reward}(p_i, p_j)$ 
17        end
18      end
19       $p_{i_j^*} \leftarrow \arg \max_{p_i \in P_{s,t}} (r_{ji});$ 
20       $r_{j_i^*} \leftarrow \text{reward}(p_{i_j^*}, p_j);$ 
21       $\pi_{i_j^*} \leftarrow \Pi_{s,t}(p_{i_j^*});$ 
22       $\pi_j \leftarrow \{\pi_{i_j^*}, p_j\};$ 
23    end
24     $\pi_{s,t+T_{int}}^* \leftarrow \arg \max_{\pi_j \in \Pi_{s,t+T_{int}}} (r_{j_i^*});$ 
25     $x_{s,t} \leftarrow x_{s,t+T_{int}};$ 
26     $P_{s,t} \leftarrow P_{s,t+T_{int}};$ 
27     $\Pi_{s,t} \leftarrow \Pi_{s,t+T_{int}};$ 
28    update( $M_{s,t+T_{int}}$ );
29  end
30  if  $t \bmod T_{comm}$  then
31    | ISC( $S, \Pi$ );
32  end
33   $t \leftarrow t + T_{int};$ 
34 end

```

Swarm Attitude Planning Algorithm

Algorithm 2 is a computationally efficient way of implementing the DP formulation. First, it initializes the spacecraft state, x_{s,t_0} , the set of possible initial pointing directions, P_{s,t_0} , the target coverage map, M_{t_0} , and the set of optimal policies up to each $p_{j,t_0} \in P_{s,t_0}$, denoted Π_{s,t_0} . The algorithm achieves efficient computation by dividing the planning horizon into discrete time intervals of duration T_{int} , at which times each spacecraft is scheduled to initiate a change of attitude. At each of these intervals, for a given servicing spacecraft, $s \in S$, the algorithm first uses the spacecraft's state at t to predict its state at future times $t + T_{slew}$ and $t + T_{int}$. This prediction is computed using the known orbital mechanics of the system and is denoted by `orbit` (Alg.2 lines 7-8). Then, the spacecraft's options for the next pointing direction in its attitude trajectory are restricted to the subset $P_{s,t+T_{int}} \subset P$ of pointing directions that will be within its sensor FOV boundary (cone with angle Φ_s) from time $t + T_{slew}$ until $t + T_{int}$. These pointing directions are a function of the spacecraft's ACS and sensor capabilities, and this function is denoted by `range` (Alg.2 lines 9-10). Next, the algorithm searches for the best previous pointing direction leading up to each $p_j \in P_{s,t+T_{int}}$. This search is restricted to the set of pointing directions from which the spacecraft ACS can slew to and settle at the next pointing option within the time T_{slew} . If the ACS slew rate, ω_{slew} , is fast enough to make this transition and settle within τ_{settle} , all before T_{slew} expires, then the function `slew` (Alg.2 line 14) returns 1. Then, the reward corresponding to the transition from p_i to p_j

is computed using Eq. (10), denoted by `reward` (Alg.2 line 15). After evaluating the reward for all feasible transitions from $p_i \in P_{s,t}$ to $p_j \in P_{s,t+T_{int}}$, the best policy, π_j , leading up to each pointing direction, p_j , becomes the optimal policy ending in the previous pointing direction with the maximum reward, denoted $\pi_{i_j^*}^*$. The set of best policies for each p_j is saved along with its corresponding reward, which are then used at the next time interval as the initial conditions. The optimal policy for a given satellite s , up to time t , is denoted $\pi_{s,t+T_{int}}^*$, and is defined as the policy, π_j , that ends in the pointing direction p_j with the highest cumulative reward.

After a period of T_{comm} , each spacecraft shares its optimal attitude trajectory with the rest of the swarm through intersatellite communication, denoted `ISC` (Alg.2 line 30). Each spacecraft assumes that these communicated paths are implemented and updates its surface coverage map, $M_{s,t}$, accordingly. When the end of the planning horizon is reached, each spacecraft follows its trajectory until the end of the replanning period, T_{replan} . Then, Algorithm 2 is repeated.

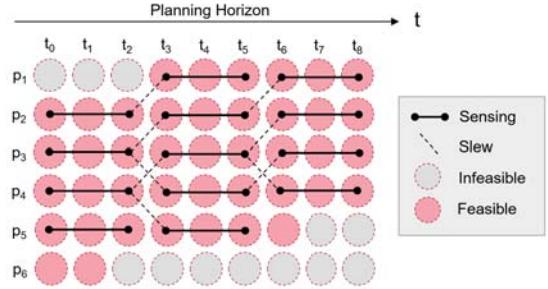


Figure 6: The swarm attitude planning DP algorithm optimizes the sequence of pointing directions to maximize information gain.

Fig. 6 illustrates a simple implementation of Algorithm 2. In this example, the sensing time constraint T_{sense} is three discrete time steps, and the slewing period T_{slew} is one discrete time step. In the figure, the algorithm is evaluating the best policies leading up to each feasible point at t_8 . At times t_3 and t_6 , the figure shows the optimal transitions to each pointing direction in the 'next' set of pointing directions. Infeasible pointing directions are those that were outside of the FOV bounds Φ_s at the corresponding time step t_n .

6. SIMULATION RESULTS

Swarm Orbit Design

The swarm orbit design algorithm makes orbit selection based on the information gain and transfer fuel cost associated with orbit candidates. To better demonstrate the effects of orbit design parameters on the optimal set of orbits, we fix the number of orbit candidates $n_{orbit} = 650$ and required sensor distance $d_s = 50m$. The target object is chosen as a sphere of radius 10m in a circular orbit in LEO and assumed to be not rotating in the relative coordinate frame. Moreover, we assume that the initial pose of each spacecraft is equal to the initial pose of its PRO. This way, the transfer fuel cost is independent of the initial pose of the swarm.

The information gain metric, Eq. (3), used to represent the amount of scientific data collected from inspection has $\alpha \in [0, 1]$ as a parameter that determines balance between

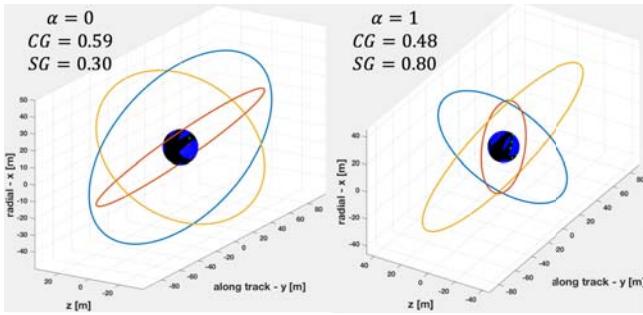


Figure 7: The parameter α can be adjusted to generate a set of orbits that focuses purely on exploration ($\alpha = 0$, maximizes coverage gain) or exploitation ($\alpha = 1$, maximizes science gain). It acts as a dial whose value can be set between 0 and 1 to balance the exploration and exploitation behavior of the generated orbits.

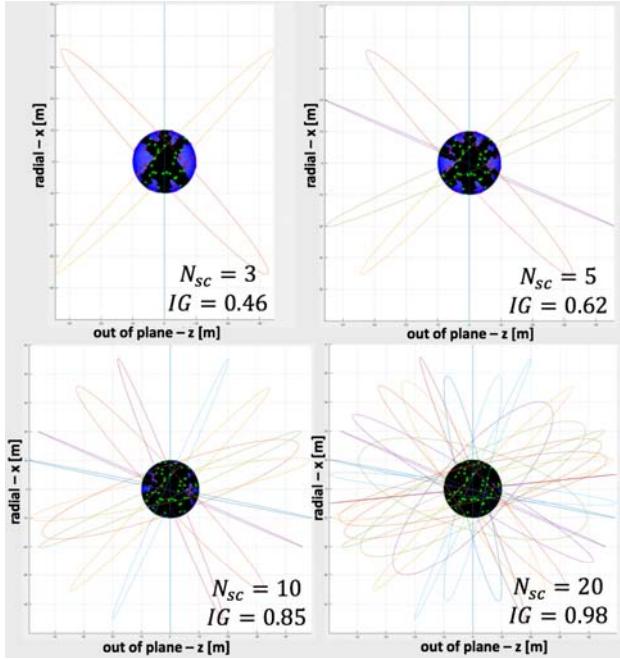


Figure 8: Orbit assignment is performed sequentially using heuristic ordering based on the fuel budget of each spacecraft in the swarm. The assignment has no guarantee of global optimality but can quickly generate a near-optimal set of orbits that scales linearly with the size of the swarm N_{sc} .

exploration and exploitation. Fig. 7 shows the optimal set of orbits generated for two extremes, $\alpha = 0$ and $\alpha = 1$, for three spacecraft. In the figure, black areas are surface partitions of the target object that is covered, and green dots represent science targets that are covered. Setting α to 0 forces the orbit design algorithm to maximize the surface area covered from the set of orbits as shown by the left plot in Fig. 7. On the other hand, setting α to 1 results in a set of orbits that collectively maximizes the science targets covered.

The swarm orbit design algorithm sequentially assigns an optimal PRO to each spacecraft in the swarm in an order that is set by minimum fuel budget heuristics. Fig. 8 shows the optimal set of PROs generated for a varying number of spacecraft in the swarm. The sensor FoV is set to 6 degrees,

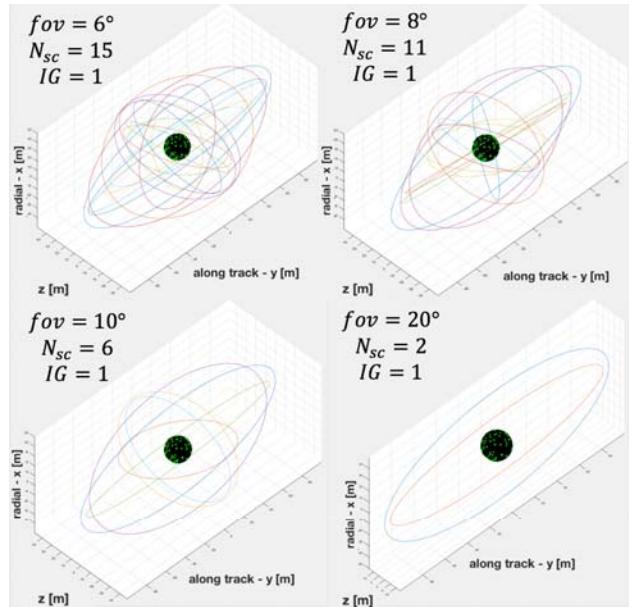


Figure 9: Sensor field of view (FoV) plays an important role in determining the number of spacecraft required to completely cover a given target object from the optimal set of PROs.

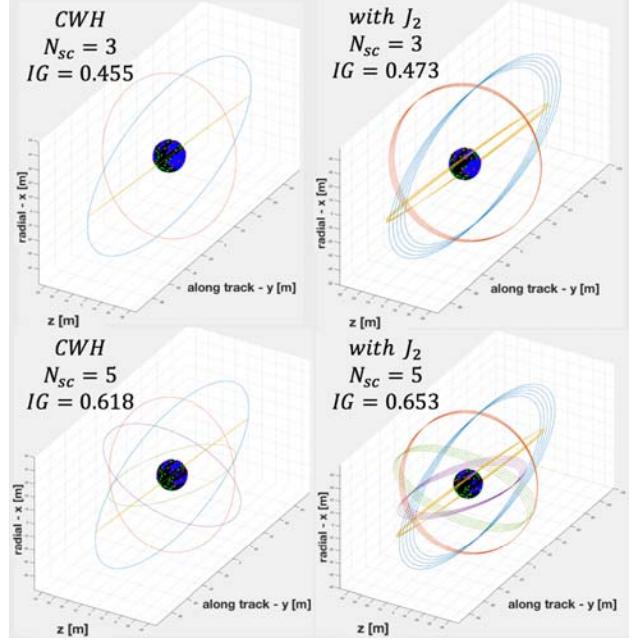
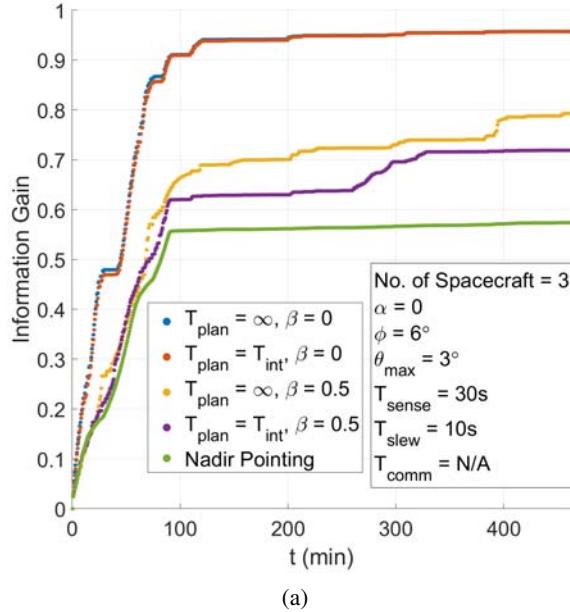
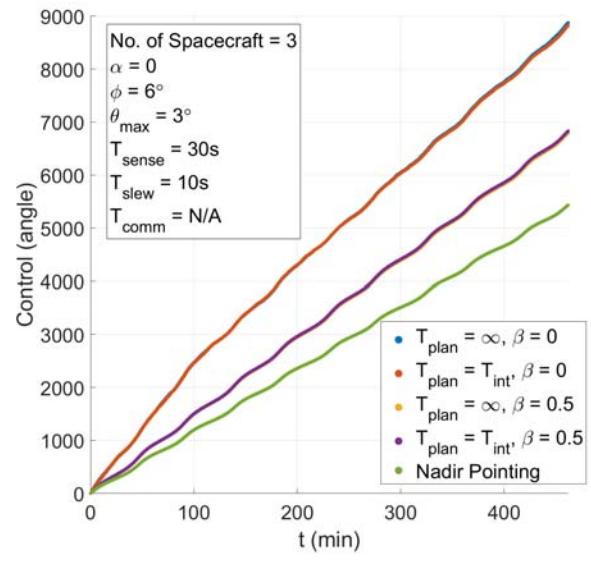


Figure 10: An optimal set of orbits has increased information gain due to drift when J_2 effect is considered for the orbit trajectories.

and the estimate of the information gain is based on the lower bound established to be on the conservative side for the simulation. Due to the sequential nature of the greedy assignment approach taken, the set of PROs generated scales in its computation linearly to the size of the swarm (N_{sc}). Although this computational efficiency comes at the cost of losing the guarantee of global optimality, a near-optimal set of PROs that collectively maximizes information gain can be generated quickly.

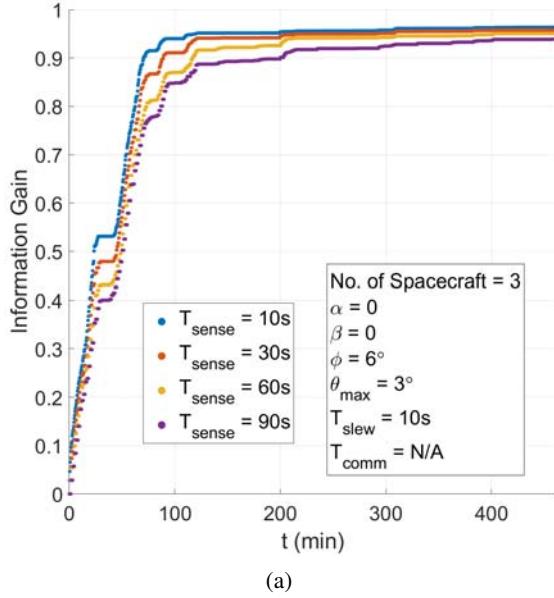


(a)

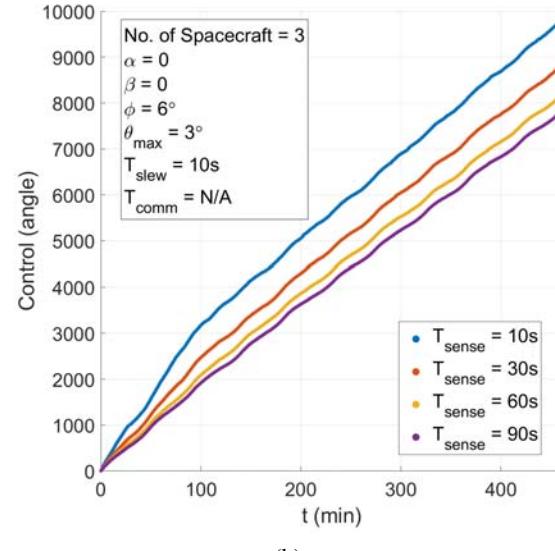


(b)

Figure 11: Optimized attitude planning achieves higher information gain with less fuel used when compared to the trivial greedy solution.



(a)



(b)

Figure 12: Coverage and ACS actuation per unit time decrease as the required sensing time increases.

The size of swarm required to completely cover the target object depends significantly on the specified sensor field of view. Fig. 9 shows minimum number of spacecraft and its corresponding set of PROs needed to ensure complete coverage of the target object. As expected, increasing the sensor FoV dramatically reduces the required size of the swarm to a point where two spacecraft can fully cover the object when FoV is increased to 20 degrees.

The swarm orbit design algorithm is not tied to specific orbital dynamics. Since it numerically integrates forward each orbit trajectory from a set of initial conditions (Alg.1 line 9), orbital dynamics with higher order terms can be used in place of the linearized dynamics. Fig. 10 shows set of

PROs generated when the J_2 effect is included in the orbital dynamics compared to its linearized counterpart. The result indicates that J_2 -perturbed orbits have increased information gain due to drift that does not exist in the orbits computed with CWH equations.

Attitude Planning

To study the various trade-offs in the attitude planning algorithm, we isolate significant parameters in simulation. Initially, we consider the information gain simply to be the cumulative sensor coverage of the swarm on the target ($\alpha = 0$). So, in the Figures 11 and 12 below, let 1 and 0 be 100% and 0% coverage of the target, respectively. Then, in Figures 13 and 14, we consider the information gain to be a weighted

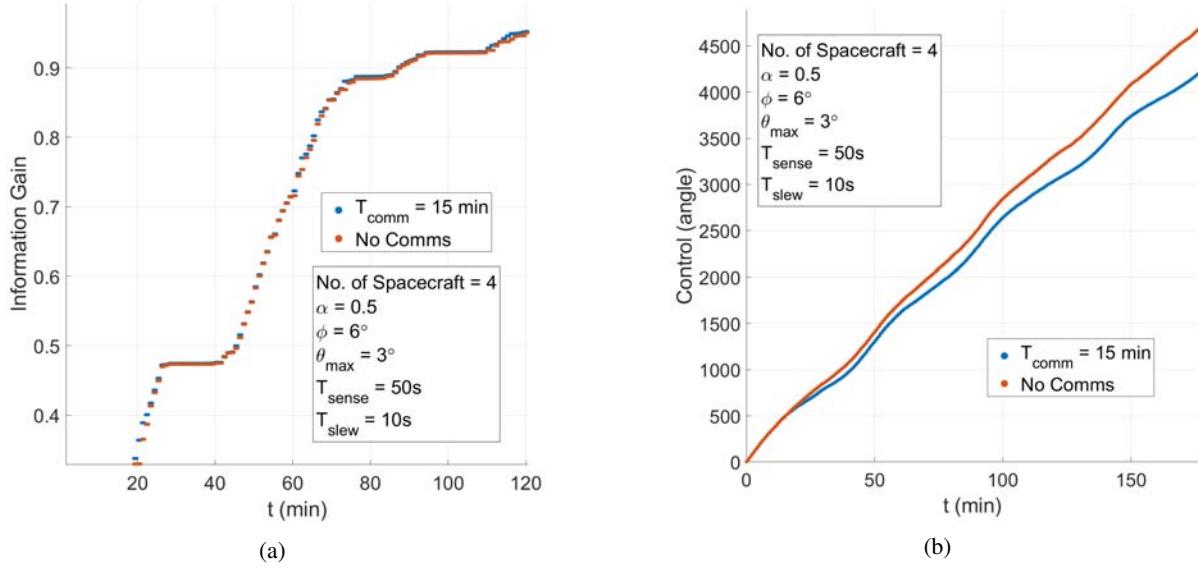


Figure 13: Information transfer between spacecraft improves efficiency of the attitude planning.

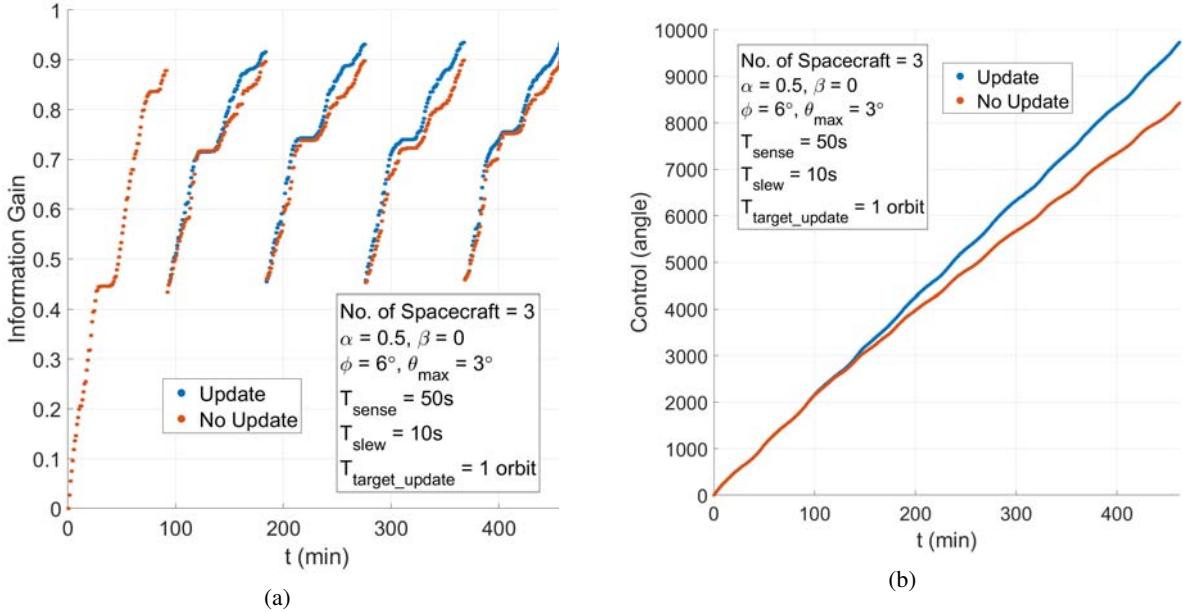


Figure 14: The attitude planning algorithm adapts to updated target locations of interest.

sum of the coverage gain and science gain, as defined in Eq. (3). In all plots, the ACS actuation cost is defined to be the cumulative attitude angle change measured in degrees and given in Eq. (12).

First, we examine the trade-off between maximizing sensor coverage and minimizing ACS actuation cost. Fig. 11 shows that, when the reward function only maximizes sensor coverage, the optimal attitude trajectory performs almost equally as well as a simple greedy solution. However, when ACS actuation cost is considered in the reward function, the optimal solution covers significantly more of the target than the greedy solution with approximately the same amount of ACS actuation.

We also look at the simple trade-off of changing the constrained sensing time. As expected, the results in Fig. 12

show that, in each case, the same final coverage value is approached but at a slower rate for longer sensing times.

The use of information transfer during the attitude planning stage has its greatest effect when there is significant overlap between each spacecraft's field of view boundary. This occurs when PROs are closer together, either due to a high number of spacecraft or due to concentration around particular areas of interest. To obtain the simulation results shown in Fig. 13, we simulated four spacecraft in orbit, comparing the use of periodic information transfer to no information transfer. For these simulations, we also randomly selected fixed science target partitions that cover 1% of the target surface, and we give each spacecraft a map of these targets. The results show that information transfer allows for more efficient sensing, from which full sensor coverage is obtained at a lower ACS actuation cost. In other cases, when overlap is even greater,

information transfer should prove to be even more valuable.

Last, we study the attitude planning algorithm's ability to adapt when the science targets are periodically updated by a human operator. In these simulations, the science targets are randomly re-selected at the beginning of each orbit. In each case, all spacecraft have a map of the initial science target locations, but only in one case is the spacecraft's map updated every orbit. In the orbit selection stage of the motion planner, we still select PROs to maximize only coverage (i.e. initially, $\alpha = 1$). This way, the swarm can adapt to updated locations of science targets. Fig. 14 shows the benefit of updating each spacecraft's knowledge of the locations, where the science target partitions cover roughly 5% of the target surface.

7. CONCLUSION

On-orbit servicing (OOS) is a sought-after capability both by NASA and the commercial space sector. Utilizing a swarm of small-spacecraft has the potential to provide more versatile, robust, and cost-effective servicing. In particular, on-orbit remote inspection would greatly benefit from swarm technology. This paper addresses the challenge of coordinating the motion of a spacecraft swarm for an effective on-orbit inspection. The proposed motion planner selects optimal orbits and plans attitude pointing directions for a swarm of spacecraft that maximizes the amount of scientific data gathered from on-orbit inspection of an Earth-orbiting satellite. The orbit selection algorithm takes a prioritized planning approach with heuristics to efficiently generate a near-optimal set of orbits and scales linearly with the size of the swarm. The attitude planning algorithm achieves computational tractability by discretizing the target surface into a set of pointing directions and optimizing the attitude trajectory along these pointing directions using dynamic programming. Both algorithms have explicit parameters that can be adjusted to balance the competing mission objectives of maximizing information gain and minimizing actuation cost. Simulation results show the planner to be effective in planning spacecraft motion for varying design parameters involved with on-orbit inspection. The results also demonstrate the planner's ability to adapt to updates in the science target through re-planning of the attitude online.

Future research will address more aspects of the remote inspection. For instance, the full dynamics of the attitude control system should be accounted for in the attitude planning stage, rather than treating the ACS as a set of constant parameters. The information gain model also has significant room for improvement, and a more realistic model should be developed. Additionally, the constraints related to the inter-satellite communication link should be considered. Moreover, future research will involve more phases of the remote inspection mission, such as the fuel cost and motion planning required to bring each spacecraft into its optimal orbit. Ultimately, further research should incorporate these motion planning algorithms into a unified remote inspection system. Such a system might involve storing small-spacecraft aboard the target satellite. Then, each small-spacecraft would be deployed into optimal PROs to inspect the target. Upon completion of the inspection, each OOS spacecraft would then rendezvous with the target and prepare for future inspections. This hypothetical system would have potential applications in high-value satellites or even crewed spacecraft, such as the ISS or Lunar Gateway.

ACKNOWLEDGMENTS

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). We would like to thank Yashwanth Nakka and Wolfgang Hoening for their valuable feedback on the work and also Michael Wolf for reviewing the manuscript. ©2019 California Institute of Technology. All rights reserved.

REFERENCES

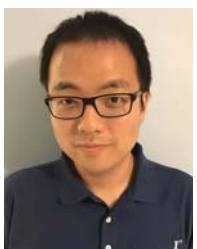
- [1] J. P. Davis, J. P. Mayberry, and J. P. Penn, "On-orbit servicing: Inspection, repair, refuel, upgrade, and assembly of satellites in space," The Aerospace Corporation, report, 2019.
- [2] R. B. Friend, "Orbital express program summary and mission overview," in *Sensors and systems for space applications II*, vol. 6958. International Society for Optics and Photonics, 2008, p. 695803.
- [3] "Mission extension vehicle," https://www.northropgrumman.com/Capabilities/SpaceLogistics/Documents/MEV/_Factsheet.pdf, accessed: 2019-10-13.
- [4] B. Banker and R. Askew, "Seeker 1.0: Prototype robotic free flying inspector mission overview," in *33rd Annual Conference on Small Satellites*, 2019.
- [5] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1 – 26, 2014.
- [6] R. Foust, S.-J. Chung, and F. Hadaegh, "Autonomous in-orbit satellite assembly from a modular heterogeneous swarm using sequential convex programming," in *AIAA/AAS Astrodynamics Specialist Conference*, 2016, p. 5271.
- [7] D. Piskorz and K. L. Jones, "On-orbit assembly of space assets: A path to affordable and adaptable space infrastructure," The Aerospace Corporation, report, 2018.
- [8] P. Culbertson and M. Schwager, "Decentralized adaptive control for collaborative manipulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 278–285.
- [9] S. Bandyopadhyay, R. Foust, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Review of formation flying and constellation missions using nanosatellites," *Journal of Spacecraft and Rockets*, no. 0, pp. 567–578, 2016.
- [10] K. Knobelspiesse and S. Nag, "Remote sensing of aerosols with small satellites in formation flight," *Atmospheric Measurement Techniques*, vol. 11, no. 7, 2018.
- [11] S. Nag, A. S. Li, and J. H. Merrick, "Scheduling algorithms for rapid imaging using agile cubesat constellations," *Advances in Space Research*, vol. 61, no. 3, pp. 891–913, 2018.
- [12] S. Nag, A. S. Li, V. Ravindra, M. S. Net, K.-M. Cheung, R. Lammers, and B. Bledsoe, "Autonomous scheduling of agile spacecraft constellations with delay tolerant networking for reactive imaging," in *International Conference on Automated Planning and Scheduling*, 2019.
- [13] B. Yost, E. Agasid, R. Burton, R. Carlino, G. De-

- fouw, A. D. Perez, A. D. Karacalolu, B. Klamm, A. Rademacher, J. Schalkwyck, R. Shimmin, J. Tilles, and S. Weston, "Small spacecraft technology state of the art," NASA Ames Research Center, report, 2018.
- [14] C. M. Pong, "On-orbit performance and operation of the attitude and pointing control subsystems on asteria," in *Proc. of the 32nd Annual AIAA/USU Conf. on Small Satellites*, 2018.
- [15] A. Rahmani, S. Bandyopadhyay, F. Rossi, M. Wolf, J.-P. de la Croix, and J. Vander Hook, "Swarm of space vehicles and future opportunities," NASA Jet Propulsion Lab, report, 2018.
- [16] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.
- [17] "Jpl strategic implementation plan 2018," NASA Jet Propulsion Lab, report, 2018.
- [18] F. Y. Hadaegh, S.-J. Chung, and H. M. Manohara, "On development of 100-gram-class spacecraft for swarm applications," *IEEE Systems Journal*, vol. 10, no. 2, pp. 673–684, 2014.
- [19] K. Solovey and D. Halperin, "On the hardness of unlabeled multi-robot motion planning," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1750–1759, 2016.
- [20] J. K. Johnson, "On the relationship between dynamics and complexity in multi-agent collision avoidance," *Autonomous Robots*, vol. 42, no. 7, pp. 1389–1404, 2018.
- [21] S. Bandyopadhyay, G. P. Subramanian, R. Foust, D. Morgan, S.-J. Chung, and F. Hadaegh, "A review of impending small satellite formation flying missions," in *53rd AIAA Aerospace Sciences Meeting*, 2015, p. 1623.
- [22] D. Morgan, S.-J. Chung, L. Blackmore, B. Acikmese, D. Bayard, and F. Y. Hadaegh, "Swarm-keeping strategies for spacecraft under j2 and atmospheric drag perturbations," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 5, pp. 1492–1506, 2012.
- [23] K. Alfriend, S. R. Vadali, P. Gurfil, J. How, and L. Breger, *Spacecraft formation flying: Dynamics, control and navigation*. Elsevier, 2009, vol. 2.
- [24] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1261–1285, 2016.
- [25] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, "Model predictive control of swarms of spacecraft using sequential convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 6, pp. 1725–1740, 2014.
- [26] J. P. Van Den Berg and M. H. Overmars, "Prioritized motion planning for multiple robots," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 430–435.
- [27] M. Čáp, P. Novák, A. Kleiner, and M. Selecký, "Prioritized planning algorithms for trajectory coordination of multiple mobile robots," *IEEE transactions on automation science and engineering*, vol. 12, no. 3, pp. 835–849, 2015.
- [28] W. Clohessy and R. Wiltshire, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.
- [29] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen, "A survey of spacecraft formation flying guidance and control (part i): Guidance," in *Proceedings of the 2003 American control conference*. IEEE, 2003, pp. 1733–1739.
- [30] ——, "A survey of spacecraft formation flying guidance and control (part ii): Control," in *Proceedings of the 2004 American control conference*, vol. 4. IEEE, 2004, pp. 2976–2985.
- [31] J. A. Starek, E. Schmerling, G. D. Maher, B. W. Barber, and M. Pavone, "Real-time, propellant-optimized spacecraft motion planning under clohessy-wiltshire-hill dynamics," in *2016 IEEE Aerospace Conference*. IEEE, 2016, pp. 1–16.
- [32] D. J. Zanon and M. E. Campbell, "Optimal planner for spacecraft formations in elliptical orbits," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 1, pp. 161–171, 2006.
- [33] M. E. Campbell, "Planning algorithm for multiple satellite clusters," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 5, pp. 770–780, 2003.
- [34] L. Breger and J. P. How, "Gauss's variational equation-based dynamics and control for formation flying spacecraft," *Journal of guidance, control, and dynamics*, vol. 30, no. 2, pp. 437–448, 2007.
- [35] D. Sternberg, M. Chodas, C. Jewison, M. Jones, and O. De Weck, "Multidisciplinary system design optimization of on orbit satellite assembly architectures," in *2015 IEEE Aerospace Conference*. IEEE, 2015, pp. 1–14.

BIOGRAPHY



Benjamin Bernhard is pursuing his B.S. in electrical engineering at the University of Notre Dame. His research interests include motion planning, estimation, and control with applications in autonomous spacecraft, multi-robot systems, and self-driving cars. During the summer of 2019, he participated in a research internship at Jet Propulsion Laboratory. Now, he is conducting research at Notre Dame's Interdisciplinary Studies in Intelligent Systems Lab before graduating in May 2020.



Changrak Choi is a robotics technologist in the Maritime and Multi-Agent Autonomy group at JPL. His research focuses on trajectory and motion planning for autonomous systems that are dynamically interesting, drawing upon algorithms, optimization, and controls. The systems of interest range from ground to on-water and aerial vehicles as well as soft robots. Changrak received his PhD from MIT as a member of Laboratory for Information and Decision Systems (LIDS) in Aerospace Robotics and Embedded Systems group. He earned MS and BS, both in Mechanical Engineering, from MIT and Seoul National University respectively.



Amir Rahmani is a robotics systems engineer at JPL. Amir's current research interests encompass coordination and control of teams and swarms of autonomous vehicles. His expertise include designing estimation and control frameworks for formation flying spacecraft, as well as unmanned aerial, ground, and surface swarms. Prior to joining JPL, Amir was an assistant professor of Mechanical and Aerospace Engineering at the University of Miami. He received his PhD from University of Washington and held a postdoctoral appointment at the Georgia Institute of Technology.

among others. His research interests include optimal estimation and control as applied to distributed spacecraft. Dr. Hadaegh is a JPL Technical Fellow and a Fellow of The American Institute of Aeronautics and Astronautics. He has received numerous awards, including NASA's Exceptional Service and Exceptional Achievement medals and JPL's Award of Excellence for Flight Validation of Autonomous Rendezvous in Earth Orbit.



Soon-Jo Chung (M'06-SM'12) received the B.S. degree (*summa cum laude*) in aerospace engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1998, and the S.M. degree in aeronautics and astronautics and the Sc.D. degree in estimation and control from Massachusetts Institute of Technology, Cambridge, MA, USA, in 2002 and 2007, respectively.

He is currently the Bren Professor of Aerospace in the Graduate Aerospace Laboratories of the California Institute of Technology (GALCIT) and also a Research Scientist of the NASA Jet Propulsion Laboratory, Pasadena, CA, USA. He was with the faculty of the University of Illinois at Urbana-Champaign (UIUC) during 2009-2016. His research interests include spacecraft and aerial swarms and autonomous aerospace systems, and in particular, on the theory and application of complex nonlinear dynamics, control, estimation, guidance, and navigation of autonomous space and air vehicles.

Dr. Chung was the recipient of the UIUC Engineering Deans Award for Excellence in Research, the Beckman Faculty Fellowship of the UIUC Center for Advanced Study, the U.S. Air Force Office of Scientific Research Young Investigator Award, the National Science Foundation Faculty Early Career Development Award, and three Best Conference Paper Awards from the IEEE and the American Institute of Aeronautics and Astronautics. He is an Associate Editor of IEEE Transactions on Robotics and Journal of Guidance, Control, and Dynamics.



Fred Hadaegh (S'75-M'80-SM'89-F'07) received the B.S.E.E. (Hons.) and M.S.E.E. degrees from University of Texas at Austin, Austin, TX, USA, in 1975 and the Ph.D. degree in electrical engineering from University of Southern California, Los Angeles, CA, USA, in 1984.

In 1984, he joined the Jet Propulsion Laboratory (JPL), California Institute of Technology, Pasadena, CA, USA, where he is currently JPL's Chief Technologist. He is also a Senior Research Scientist. For two decades, he supervised the Guidance and Control (G&C) Analysis Group, which is responsible for leading research in G&C theory and for developing algorithms and software for planetary and astrophysics missions, including the Mars Pathfinder, the Mars Exploration Rover, the Cassini Mission to Saturn, the space telescope Spitzer, and the Mars Science Laboratory for the delivery of the Curiosity rover