

Chapter 2

New Guidance, Navigation, and Control Technologies for Formation Flying Spacecraft and Planetary Landing

Fred Y. Hadaegh, Andrew E. Johnson, David S. Bayard,
Behçet Açıkmüş, Soon-Jo Chung and Raman K. Mehra

2.1 Introduction

The recent landing of the massive Mars Science Laboratory (MSL) rover Curiosity was made possible by the sky-crane touch down system. The sky-crane used a high rate, six degree-of-freedom guidance, navigation, and control (GN&C) system to slowly place the rover on the surface, detect touchdown, and fly away. MSL clearly showed the advantages of on-board closed loop GN&C and has opened the door for infusion of new GN&C technologies into the next Mars lander missions as well as other future spacecraft missions. However, MSL was not equipped with the capabilities of pin-point landing and local hazard avoidance. This chapter begins with a review of recent advances in perception technologies for on-board Hazard Detection (HD) and Terrain Relative Navigation (TRN) in Sect. 2.2. The HD and TRN perception technologies will enable the next Mars lander missions to recognize landmarks for pin-point landing or detect landing hazards on the fly for local hazard avoidance.

F.Y. Hadaegh (✉) · A.E. Johnson · D.S. Bayard · B. Açıkmüş
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California
e-mail: fred.y.hadaegh@jpl.nasa.gov

A.E. Johnson
e-mail: aej@jpl.nasa.gov

D.S. Bayard
e-mail: david.s.bayard@jpl.nasa.gov

B. Açıkmüş
e-mail: behcet@austin.utexas.edu

S.-J. Chung
University of Illinois at Urbana-Champaign, Urbana, IL, USA
e-mail: sjchung@illinois.edu

R.K. Mehra
Scientific Systems Company, Inc., Woburn, MA, USA
e-mail: rkm@ssci.com

© Springer-Verlag Berlin Heidelberg 2016
E. Feron (ed.), *Advances in Control System Technology
for Aerospace Applications*, Lecture Notes in Control
and Information Sciences 460, DOI 10.1007/978-3-662-47694-9_2

This chapter also considers the problem of flying a swarm of spacecraft in Earth orbit. Distributed spacecraft systems can collectively match or exceed the capability of a more complex monolithic space system. Spacecraft swarms [11, 27] will push the envelope of the existing formation flying spacecraft concepts by increasing the number of spacecraft comprising the swarm by one or two orders of magnitude (100–1000s), hence maximizing the benefit of distributed spacecraft systems.

The swarm approach opens up the possibility for enabling many new mission concepts like creating massively distributed large space apertures, distributed antennas, decentralized sensing networks, anti-satellite disruptors, and geometric arrangements optimized for decoy/camouflage. Moreover, spacecraft swarms can be controlled to exhibit desired complex behaviors, which cannot be achieved by a single spacecraft. Spacecraft swarms are motivated by nature and are examples of bio-inspired engineering systems. Examples from nature include a colony of ants searching for food, or a swarm of bees protecting a bee hive from intruders. Swarming behaviors observed in nature inspire and guide the development of efficient coordination and control algorithms for spacecraft swarms.

In order to construct desired formations of spacecraft swarms and permit coordinated maneuvers of spacecraft, the distributed controller needs to efficiently handle a large number of spacecraft in the network. Two methods are presented to deal with the added complexity of large number of spacecraft. First, if the aim is to control the relative motions of the spacecraft to generate desired formations of spacecraft swarms, the new method of automatically generating evolving network topologies, presented in Sect. 2.3, can be used to determine the flow of control information among the spacecraft. Second, Sect. 2.4 presents the novel approach of driving the swarm to a desired density distribution in a prescribed region of the configuration space. Instead of controlling individual spacecraft, the probabilistic guidance approach controls the average number of spacecraft per unit volume, ensuring that spatial averages converge to the desired density distribution. The swarm guidance and control methods described in Sects. 2.3 and 2.4 are predicated on an effective solution to (a) detect potential changes in existing orbital trajectories that may lead to damaging collisions; and (b) localize, track, and assess trajectories headed towards collisions. Hence, Sect. 2.5 expands on necessary models, simulations, and methods for deriving, evaluating, and comparing such optimal constellations that satisfy the stated objectives (a) and (b) for various swarm collision scenarios.

2.2 GN&C Technologies for Planetary Landing in Hazardous Terrain

2.2.1 Introduction

All robotic landers to date, including MSL, have landed blindly. They have measured altitude and surface relative velocity and used these measurements for soft landing, but they have not had the ability to recognize landmarks for pin-point landing or

detect landing hazards on the fly for local hazard avoidance. Blind landing has forced missions to select landing ellipses that are free of hazards. This greatly constrains the possible landing locations and either limits the science to what is possible in benign terrain or requires the addition of a long traverse roving capability. For example, MSL landed on the flat and smooth Gale Crater floor and will have to drive out of its 10 kilometer landing ellipse to reach the most interesting science locations.

Hazard Detection and Avoidance (HDA) [19, 21] is an on-board function that collects sensor data to map the landing site, applies an algorithm to determine the safest place to land and then guides the vehicle to the safe landing site. HDA enables the selection of landing ellipses that have a large number of known hazards that can be avoided with a small divert (10–100 m divert). It also enables landing at possibly hazardous locations with limited reconnaissance. HDA requires perception technology for on-board Hazard Detection (HD) and GN&C technology to land the vehicle precisely at the safe landing site.

In contrast, Pin-Point Landing (PPL) is an on-board function that collects sensor data and matches this data to a map of the landing ellipse generated before landing. This match is then used to obtain a map-relative position fix. From this position fix, the lander can compute a trajectory that guides the vehicle to the landing site (1–10 km divert). There is no active detection of hazards on-board, but PPL can be used to avoid large hazards in the landing ellipse. It can also be used for precision deployment of surface assets (e.g., the multi-mission Mars Sample Return architecture or building up a lunar outpost). PPL requires Terrain Relative Navigation (TRN) perception technology [3, 8, 24, 28] and GN&C technology for fuel optimal powered descent guidance.

Figure 2.1 shows a variety of planetary landing sites that can be made accessible by pin-point landing or hazard detection and avoidance. For planetary science, PPL

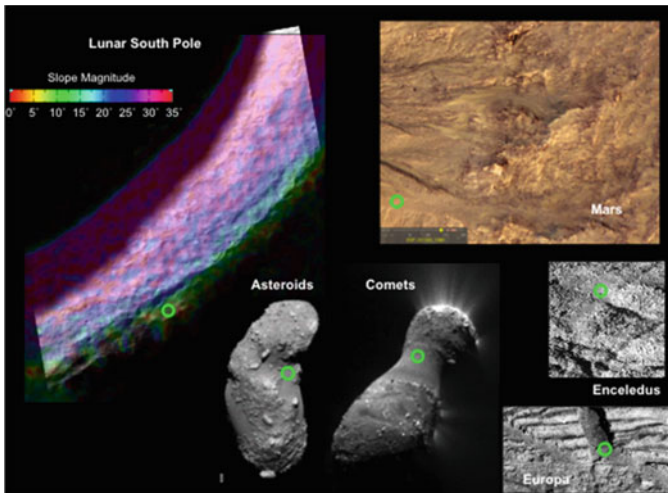


Fig. 2.1 Planetary landing sites that require pin-point landing or hazard detection and avoidance

and HDA will enable access to dust flows on comets and asteroids, seepage features on the side of a crater wall on Mars, the boulder strewn source of water plumes on Enceladus and the cracked and fissured icy terrain of Europa. For manned exploration, these technologies will enable access to the peaks of permanent light near the rugged lunar south pole which are ideal locations for a lunar outpost due to the constant illumination for solar power generation and thermal management and possible near sources of volatiles for in-situ resource utilization. It should be noted that the Apollo lunar program recognized the need for pin-point landing and hazard detection to avoid small and large hazards, but the capability was implemented manually.

The focus of this section is on the TRN and HD perception technologies; whereas GN&C technologies for powered descent are described in another chapter.

2.2.2 Design Considerations

The lander's flight system and descent and landing approach greatly influence TRN and HD design. The trajectory has a major impact on TRN and HD system design because it dictates the time available for data collection and processing, the ranges and off nadir angles for sensor operation, and the attitude rates and velocities for tracking. The mechanical design of the lander sets thresholds on the hazard detection capability by dictating the tolerable surface slope and roughness at touchdown. The performance of the lander's GN&C and propulsion system provide constraints on divert sizes and accuracy, which affect the overall pin-point landing accuracy and safe landing site size. Since TRN and HD algorithms require extensive processing in a short amount of time, it is necessary to have a dedicated, possibly high performance, processor. The size of the lander will also influence the mass, power and volume available for the TRN and HD system.

The environment plays an important role as well. The transmission properties of the atmosphere and the reflectivity of the terrain will influence sensor range and image contrast. If present, dust can reduce sensor range or add noise to sensor measurements. The size and distribution of terrain features (rocks, craters, scarps, hills, etc.) will determine the density of safe landing areas and influence the area needed for HD imaging and PPL divert distances. During passive approaches, the terrain as well as the illumination will influence the appearance of the imaged scene. PPL requires a map made prior to landing; while the performance of TRN depends on the pixel size and quality of the map data. Passive TRN approaches could require rendering of a digital elevation map to generate an image for matching, which makes them more sensitive to map quality than active sensor based approaches [8].

Because planetary landings occur only once, it is difficult if not impossible to test system performance prior to the actual landing. TRN and HD systems must undergo extensive validation and must be designed from the bottom up to be robust. Bolton TRN and HD systems, composed of sensors, processors and algorithms with a low bandwidth interface to the spacecraft, facilitate validation because the entire system can be tested in the field under relevant conditions without the rest of the spacecraft.

Bolt on systems also localize the timing of sensor data and alignment of sensors, which greatly simplifies integration with the spacecraft.

The mission specific design constraints flow down into requirements on the sensors (field of view, number of pixels, maximum range, measurement errors, etc.), algorithms (position accuracy, detection rates, map size, etc.) and processing (clock speed, memory, etc.). As the design space is quite large, there is no generic system for TRN and HD. Instead TRN and HD need to be tailored to each specific application. Since algorithms and sensors have already been developed that meet TRN and HD requirements, the focus has now moved to the development of complete systems. Below, we describe two systems under development: one for Mars robotic landing and the other for crewed lunar landing.

2.2.3 Case Study 1: Mars Robotic System

Mars landers have an entry phase which is used to reduce most of the surface velocity and, in the case of MSL, reduce the landing ellipse size down to 10 Km radius. After entry, a parachute is deployed and the heat-shield is ejected. At this point sensors can image the ground and the TRN function can start. The parachute descent phase lasts from 10 km altitude to 2 Km altitude, with vertical velocities near 100 m/s and horizontal velocities less than 30 m/s. Toward the end of the parachute descent, the off nadir angle is less than 20° and attitude rates are less than $20^\circ/\text{s}$. TRN processing completes when powered descent starts around 2 km. Powered descent performs a TRN commanded large divert and then goes into a vertical descent phase around 250 m with a 30 m/s vertical velocity. HD occurs quickly at the start of vertical descent, when the off-nadir angle is low and the nominal landing site is directly below. Once the safe site is identified, the lander targets it with a small divert. The entire descent from heat-shield separation to landing, takes on the order of 100 s.

Mars landing typically occurs during the day and at low latitudes, to have direct communication with Earth during landing, so that more accurate and mature camera based TRN approaches can be employed. As was done for MSL, a 1 m digital elevation map with co-registered images can be generated using Mars Reconnaissance Orbiter images. This high resolution map enabled TRN accuracy in the order of 10 m.

Robotic landers are small with touchdown areas in the order of 10 m^2 , but Mars is also hazardous with plenty of rocks, scarps and craters. Assuming the MSL hazard tolerance of 55 cm rocks and 22° slopes, studies of the MSL landing sites have shown that a $12 \times 12 \text{ m}$ hazard map generated with a single flash LIDAR image followed by a 6 m divert greatly reduced the chances of landing on a hazard.

Based on these constraints, the Lander Vision System (LVS) was conceived within NASA's Science Mission Directorate as a tightly integrated bolt-on smart sensor system that has well defined path to flight implementation [20]. The LVS measures terrain relative position, velocity, attitude and altitude while also detecting landing hazards. The LVS sensor suite includes an imaging camera for the landmark recognition required for terrain relative position estimation and image-to-image feature

tracking for horizontal velocity estimation [28]. A dual use flash LIDAR is used for near surface hazard detection and long distance ranging, for measuring altitude through the entire descent. An inertial measurement unit (IMU) propagates vehicle motion between image and LIDAR measurements, so that high rate state information can be provided to the spacecraft. The sensors are tightly integrated with a high performance computer that performs all processing required for TRN, HD, altimetry and velocimetry. The current best estimates for the LVS mass and power are 8 kg and 65 W respectively.

The LVS is optimized to generate robust and accurate measurements from a minimal suite of sensors. Each sensor serves multiple purposes, which reduces mass, volume and development costs. The flash LIDAR is the ideal sensor for hazard detection because it can generate all the data required with a single, low noise, range image taken at long distances [23, 29]. By decreasing the width of the laser illumination, the flash LIDAR can also be used to measure range at high altitude, thereby removing the need to add a separate altimeter [22]. The camera provides images of landmarks for position estimation [8]. It also provides image-to-image feature tracks for velocity estimation, which eliminates the need for a separate velocimeter. Finally, the IMU provides the attitude propagated from the spacecraft's cruise phase that is needed to start TRN and it also allows for high rate updates of the entire navigation state, which is required for closed loop powered descent guidance.

The LVS sensors have low development risks. Cameras and IMUs have already flown on many missions and are not a concern for development. Advanced Scientific Corporation (ASC) has developed a flash LIDAR, and, under NASA funded Small Business Innovative Research Grants, ASC has also built a prototype of a flash LIDAR that satisfies the LVS requirements and also uses parts with flight equivalents [29]. Through extensive field testing, the ASC flash LIDAR has demonstrated that it can detect hazards at low altitude (< 500 m) [23] and measure accurate ranges at high altitude (up to 8 km) [22]. Given all these advances toward flight qualification, the flash LIDAR is also a fairly low risk sensor.

The computational tasks are done using a flight qualified processor and a Field Programmable Gate Array (FPGA). The FPGA interfaces with the sensors for fast data acquisition and accurate timing. The FPGA also stores the computationally intensive software vision modules for image normalization, homography-based image warping, image interest operator, frequency domain image correlation and spatial domain image correlation. These high speed modules are used for terrain relative position and horizontal velocity estimation. The processor coordinates the flow of data from the sensors and the processing of modules on the FPGA. It also runs the navigation filter [28] that fuses inertial, imaging and range measurements and interfaces with the spacecraft to receive LVS commands and sends back navigation states and safe landing site locations. The interface between the compute element and the sensors has high bandwidth with tight constraints on timing and data latency. In contrast, the interface with the spacecraft is simpler, with a low data rate and relaxed timetag requirements.

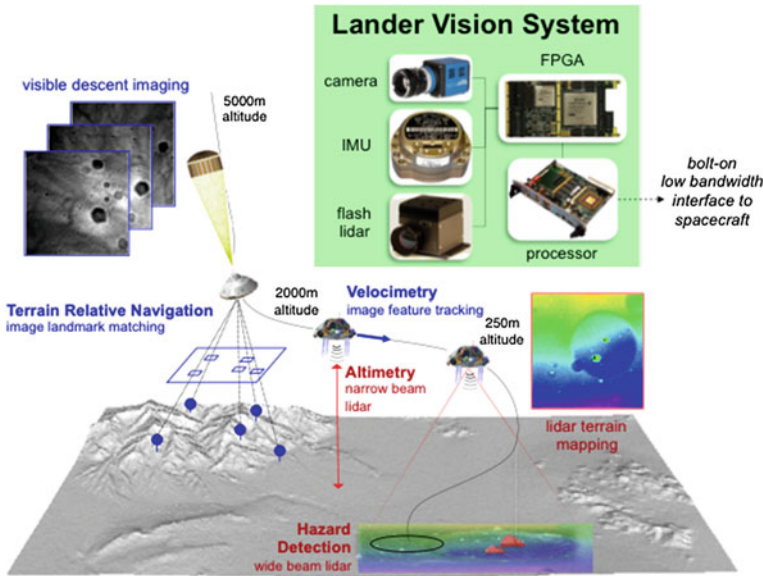


Fig. 2.2 Mars robotic lander vision system concept of operation and components

2.2.4 Case Study 2: Crewed Lunar System

The Autonomous Landing and Hazard Avoidance Technology (ALHAT) project under NASA's Human Exploration and Operations Mission Directorate is developing sensors and algorithms to increase safety during crewed and un-crewed lunar landings [15]. ALHAT's charter is to develop a system that can land anywhere, under any lighting conditions and the lunar south pole is a challenging case that has focused the ALHAT development. Operation under any lighting conditions has resulted in a system that employs active sensors for TRN and HD (Fig. 2.2).

Following the approach used for Apollo, crewed lunar landing starts with a de-orbit maneuver that places the lander on a long shallow trajectory to the surface. During descent, attitude rates are very low and velocities start at 2000 m/s near orbit. When the lander reaches 15 Km altitude, the braking burn begins and the LIDAR is activated to take range measurements. For TRN, the ranges are combined into an elevation contour and this contour is matched with a digital elevation map to obtain a position fix [22, 24]. Based on the TRN measurement, a trajectory is computed on board to clean up the trajectory dispersions (<1 km). This process repeats until around 1–2 km range and 30 m/s velocity, at which point the lander pitches up for landing and crew viewing of the landing site.

The hazard detection phase begins at 1 Km slant range and 30° angle from horizontal. As shown in Fig. 2.3, the ALHAT Hazard Detection System (HDS) raster scans a gimbaled flash LIDAR across a 100×100 m area; the LIDAR combines the flash LIDAR detector from ASC with large collection optics and a high power laser to

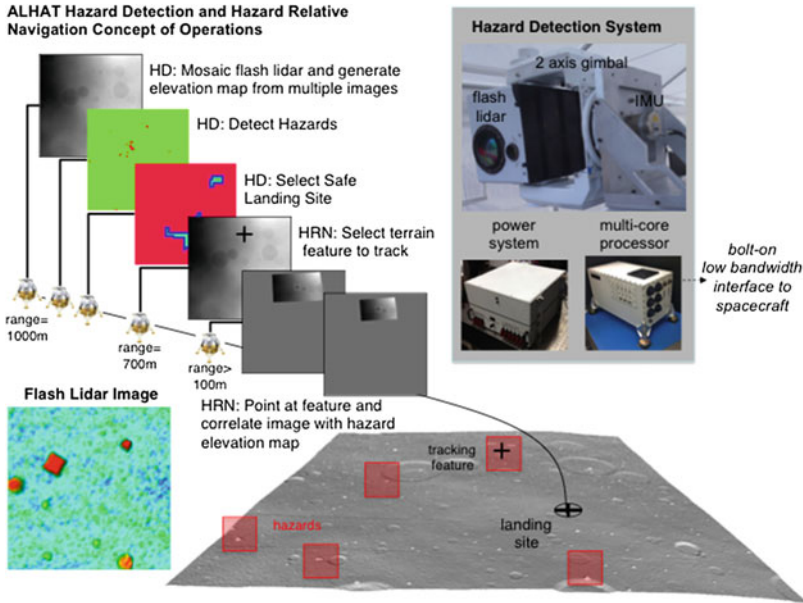


Fig. 2.3 The ALHAT hazard detection system concept of operation and hardware components

obtain the 1 Km imaging range required for ALHAT [3]. The multiple flash LIDAR images are stitched together into a single digital elevation map (DEM) using the onboard navigation solution for coarse placement and a LIDAR data-driven alignment process for fine alignment. The DEM is then passed to an HD algorithm that detects multiple safe landing sites for a lander with a touchdown area around 200 m^2 and sensitive to rocks greater than 30 cm high and slopes greater than 12° .

Once a single safe site is selected by the crew, a trajectory is generated to bring the vehicle to a point 30 m above the target. The HDS then performs Hazard Relative Navigation (HRN) [23] to keep the lander on trajectory to the safe site. During HRN, the LIDAR is pointed at a prominent elevation feature near the landing site, and the feature position is tracked during descent to provide safe site relative navigation updates. The tracked feature actually changes during descent to keep the tracked feature in front of the lander and deal with the large change in sensor footprint. Due to dust kicked up by the propulsion system, the vehicle descends the final 30 m using inertial sensors only. Fortunately, the HRN measurements and velocity from a Doppler LIDAR velocimeter, also developed in ALHAT [3], provide navigation state that is accurate enough to seed this inertial propagation and bring the lander to within 1 m of the selected safe landing site.

There are boulder fields on the Moon, but the more prevalent hazards are small craters and steep slopes. The best lunar terrain maps have been generated from Lunar Reconnaissance Orbiter (LRO) data. LRO is in a polar orbit and has provided a polar

DEM with a ground resolution of 25 m. Near the equator, stereo imaging from the LRO Narrow Angle Camera must be used to generate DEMs of sufficient resolution for TRN.

The hazard detection data collection and processing must be completed in 10 s to leave time for safe site selection by the pilot. The HDS uses a hybrid processing approach to meet this requirement. An FPGA collects and times the data from the sensors. The FPGA then passes the data to a multi-core general purpose processor which runs all of the algorithms for DEM generation, HD and HRN [32]. The multicore processor allows parallelization of time consuming processes and is straight forward to program.

2.2.5 System Comparison

Figure 2.4 shows a side-by-side comparison the LVS and ALHAT systems. The LVS has less challenging requirements because of the daytime landing, small size of the robotic lander and the significant hazard tolerance of the lander. These enabled a system that can perform TRN with the mature computer vision algorithms and sensors derived from the Descent Image Motion Estimation System on Mars Exploration Rover [25] and HD with a single flash LIDAR image, which mitigates the need for a gimbal or stitching of flash LIDAR images. Since the LVS is being developed for the

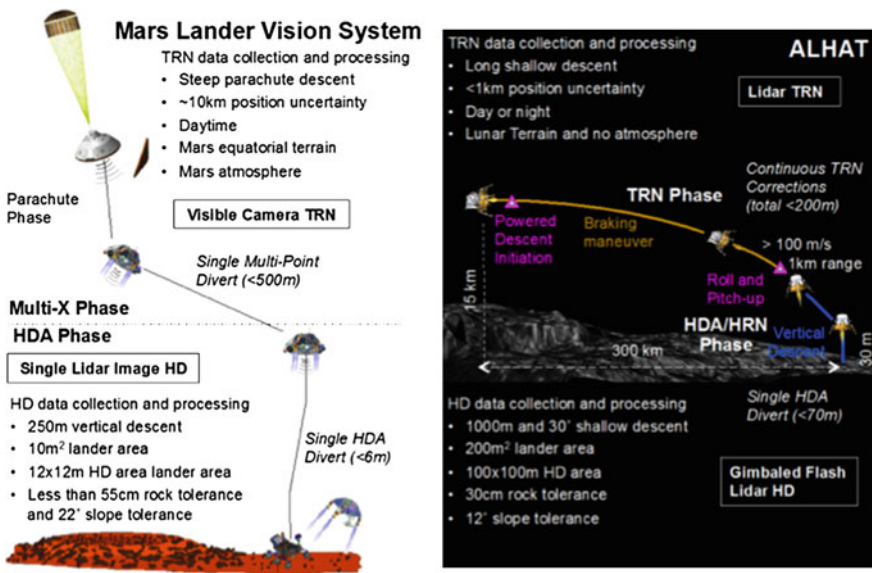


Fig. 2.4 A comparison of TRN and HD systems for Mars robotic missions and Lunar crewed missions

next Mars lander (2016–2022), the design focused on using components that could be flight qualified in the short term. A major early design choice to maintain the short duration path to flight was the selection of an FPGA based processor architecture over one utilizing multi-core processing.

The requirements on the ALHAT system are more challenging than what is needed for robotic landing and the infusion period is farther in the future (2020–2030). Consequently, the ALHAT development has focused more on meeting performance requirements and less on detailed path to flight designs. The ALHAT system must detect hazards from far away, with high resolution over a large area to safely land the 'not very hazard tolerant' crewed lunar lander. This drove the design to a gimbaled flash LIDAR and the associated increase in complexity, mass and power. The 'any lighting condition' requirement drives the TRN approach to use the less mature LIDAR sensors and contour matching algorithm. The multi-core processing approach is better from flexibility and ease of programming stand point, but its path to flight requires flight qualification of rad-hard processors.

Both the LVS and ALHAT developments are needed to meet short and long term goals of NASA. Moreover, TRN and HD remain fertile technology development areas for applications beyond planetary landing. Proximity operations around comets and near earth asteroids will use very similar algorithms and sensors. Satellite servicing and orbital debris mitigation could probably use similar algorithms and sensors. On the Earth, autonomous helicopters are being developed for cargo delivery and ship board landing, that will rely on TRN and HD functions to deal with unknown and unprepared landing sites. For all of these application domains, TRN and HD can be expanded to include additional sensing modalities like high frequency radar, thermal imaging and multi-return scanning LIDAR.

2.3 Phase Synchronization Control of Spacecraft Swarms

The objective of this section is to present an effective method for automatically generating evolving network topologies that determine how control information flows among the agents, thereby reducing the complexity of controlling a large number of spacecraft in the swarm. Directed graphs are used instead of undirected graphs to account for heterogeneous sensing or communication capabilities of spacecraft in the swarm network.

We review the recent results from our prior work [7, 9–11, 13, 27] in this section. The proposed framework of adaptive graphs is useful especially when we deal with a large number of spacecraft that perform arbitrary reconfiguration maneuvers. Another benefit of the proposed method is that the required gain for stabilization is smaller than the gain of an uncoupled control law by employing an adaptive graph Laplacian. Also, the error bound of the proposed synchronization control law is shown to be smaller than that of an uncoupled tracking control law. This justifies the use of a synchronization framework that can help to maintain a desired shape, even if individual spacecraft are shifted from their desired locations, as illustrated in

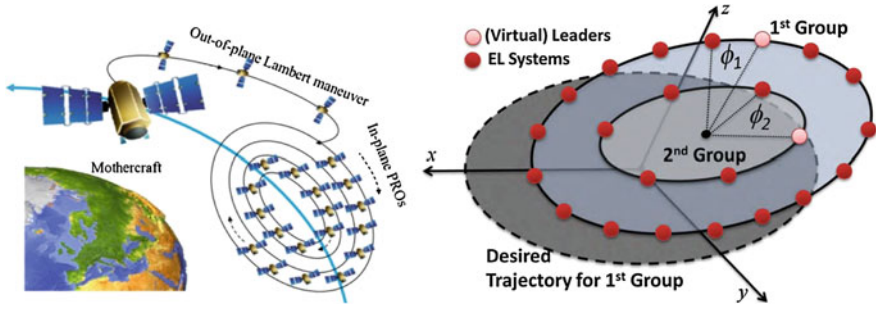


Fig. 2.5 Swarm deployment to relative elliptical orbits called Passive Relative Orbits (PROs) (*left*); Phase synchronization control of multiple spacecraft following elliptical orbits in the LVLH frame (*right*)

Fig. 2.5. The high-fidelity nonlinear dynamic model of swarm spacecraft motions, that include the effects of both Earth's oblateness and air drag in LEO [27], is used to derive the nonlinear stability proof of robust synchronization of coupled Euler-Lagrange equations, first derived in [9, 13].

2.3.1 Problem Statement—Controlling the Phase Differences in Periodic Orbits

Consider multiple spacecraft following some relative elliptical orbits as shown in Fig. 2.5. We use the term *relative*, since the elliptical motions are generated in the local-vertical local-horizontal (LVLH) frame attached to the chief orbital motion. Hence, it should not be confused with an elliptical orbit around the Earth. The relative orbital motions ($q_{tr,j} \in \mathbb{R}^3, 1 \leq j \leq p$) in the LVLH frame, of each (possibly heterogeneous) spacecraft comprising the swarm network, is governed by

$$m_j \mathbf{I}_3 \ddot{\mathbf{q}}_{tr,j} + \mathbf{C}_{tr,j}(\boldsymbol{\alpha}(t)) \dot{\mathbf{q}}_{tr,j} + \mathbf{G}_{tr,j}(\mathbf{q}_{tr,j}, \boldsymbol{\alpha}(t)) + \mathbf{D}_{tr,j}(\mathbf{q}_{tr,j}, \dot{\mathbf{q}}_{tr,j}, \boldsymbol{\alpha}(t)) = \boldsymbol{\tau}_{tr,j} \quad (2.1)$$

where m_j is the mass of each spacecraft and the nonlinear terms, including the gravitation forces with J_2 effects and the dissipative forces due to air drag, are given in [27]. In particular, the vector of six orbital parameters, $\boldsymbol{\alpha}(t)$, which defines the origin of the LVLH frame, as shown in Fig. 2.5, is governed by $\dot{\boldsymbol{\alpha}} = \mathbf{f}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})$ [27]. Furthermore, the attitude dynamics of each spacecraft can be represented by the EL form as

$$\mathbf{M}_{rot,j}(\mathbf{q}_{rot,j}) \ddot{\mathbf{q}}_{rot,j} + \mathbf{C}_{rot,j}(\mathbf{q}_{rot,j}, \dot{\mathbf{q}}_{rot,j}) \dot{\mathbf{q}}_{rot,j} + \mathbf{G}_{rot,j}(\mathbf{q}_{rot,j}, \boldsymbol{\alpha}(t)) = \boldsymbol{\tau}_{rot,j} \quad (2.2)$$

where the attitude vector $\mathbf{q}_{rot,j}$ can be represented by the first three values of quaternions or the modified Rodrigues parameters [9]. note that we can combine

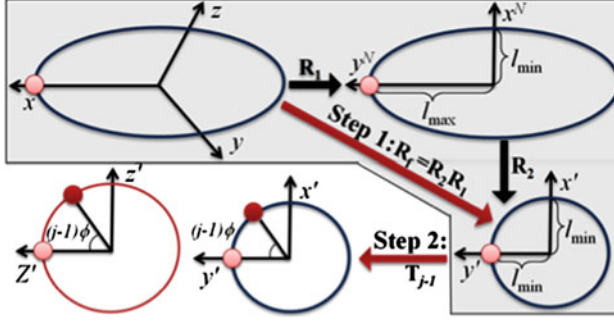


Fig. 2.6 Transformation of elliptical orbit in 3D to two *circles* for phase rotation

the dynamics of (2.1) and (2.2) to derive the dynamics of $\mathbf{q}_j = (\mathbf{q}_{rot,j}^T, \mathbf{q}_{tr,j}^T)^T$. Since $\mathbf{C}_{tr,j}(\alpha(t))$ is skew-symmetric, $\mathbf{M}_{rot,j}(\mathbf{q}_{rot,j}) - 2\mathbf{C}_{rot,j}(\mathbf{q}_{rot,j}, \dot{\mathbf{q}}_{rot,j})$ is also skew-symmetric. This property is essential for the stability proof used in this section.

Fuel efficient, J_2 -invariant elliptical orbits in the LVLH frame can be written as $\mathbf{q}_{d,tr}(t) = (x_e \sin(nt + \psi_{e0}), y_e \cos(nt + \psi_{e0}), z_e \sin(nt + \psi_{z0}))^T$ and they are used as the desired elliptical trajectory [27].

In this section, the complexity of controlling multiple spacecraft is reduced by setting a common phase angle for each desired elliptical orbit (ϕ_1 and ϕ_2 , as shown in Fig. 2.5). This common phase difference in each ellipse also sets some safe collision-free distance between each pair of spacecraft. For spacecraft swarm applications, such as sparse aperture arrays, it is more important to maintain a formation shape, by ensuring constant phase differences between the spacecraft, than exactly following a desired elliptical trajectory. Such a phase control of oscillators, called engineered central pattern generators, has been successfully applied to control multi-joint locomotive systems [12, 13, 26]. Hence we define a common phase angle in an elliptical orbit in 3D. It turns out that $\mathbf{q}_{d,tr}(t)$ can be transformed into $\mathbf{q}'_{d,tr}(t)$ in the new $x'-y'-z'$ frame, comprised of a circular motion in the $x'-y'$ plane and a sinusoidal function in the z' axis (see (Fig. 2.6) and [10] for details).

We can now perform a phase rotation by applying a rotation matrix $\mathbf{T}_{j-1} = \mathbf{I}_{n-3} \oplus \mathbf{T}((j-1)\phi) \oplus \mathbf{R}((j-1)\phi)$ for both the $x'-y'$ frame and the z' and Z' , where Z' is introduced to perform this phase rotation on the z' coordinate [10]. For the attitude dynamics of $\mathbf{q}_{rot,j}$, we do not apply phase synchronization, although it is also a straightforward extension (see [9]).

The control objective is to drive the tracking control error for each j , $\mathbf{v}_j'' = \mathbf{T}_{j-1}^T \mathbf{s}_j'' = [\mathbf{v}_j' \ v_{jZ'}'']^T = \mathbf{T}_{j-1}^T \dot{\mathbf{q}}_j'' - \dot{\mathbf{q}}_d'' + \Lambda''(\mathbf{T}_{j-1}^T \mathbf{q}_j'' - \mathbf{q}_d'')$ or \mathbf{v}_j' exponentially to zero. In other words, in the presence of modeling errors, we should show $\lim_{t \rightarrow \infty} \|\mathbf{v}_j'(t)\| \leq \Delta_T$. The phase synchronization control should yield a smaller synchronization error than the tracking error, such that for each connected pair j and k , $\lim_{t \rightarrow \infty} \|\mathbf{v}_j'(t) - \mathbf{v}_k'(t)\| \leq \Delta_S < \Delta_T$.

A conventional consensus controller without tracking control, results in undesirable drifting of the synchronized states. Then, the synchronization control ensures a smaller synchronization error that helps to maintain a formation shape as shown in Fig. 2.5. In contrast with prior work, this section uses an adaptive control scheme to automatically compute a time-varying network topology. In other words, the adaptive Laplacian matrix determines not only which neighbors each member should communicate with, but also the actual values of the time-varying gains.

2.3.2 Phase Synchronization Control Law with Adaptive Graphs

The matrices and functions in (2.1) and (2.2) are converted to the new $x'-y'-z'$ frame by the matrix \mathbf{R}_f that defines $\mathbf{q}'_j = \mathbf{R}_f \mathbf{q}_j$ and $\mathbf{q}'_d = \mathbf{R}_f \mathbf{q}_d$. Then, the dynamics in the new frame and the controllers are given as

$$M''_j(\mathbf{q}_j)\ddot{\mathbf{q}}''_j + C''_j(\mathbf{q}_j, \dot{\mathbf{q}}_j)\dot{\mathbf{q}}''_j + G''_j(\mathbf{q}_j) + D''_j(\mathbf{q}_j, \dot{\mathbf{q}}_j) = \begin{bmatrix} \mathbf{R}_f \tau_j(t) \\ \tau_{jZ'} \end{bmatrix} \quad (2.3)$$

$$\begin{aligned} \begin{bmatrix} \mathbf{R}_f \tau_j(t) \\ \tau_{jZ'} \end{bmatrix} &= M''_j(\mathbf{q}_j)\ddot{\mathbf{q}}''_{j,r} + C''_j(\mathbf{q}_j, \dot{\mathbf{q}}_j)\dot{\mathbf{q}}''_{j,r} + G''_j(\mathbf{q}_j) \\ &+ D''_j(\mathbf{q}_j, \dot{\mathbf{q}}_j) - (k_1 + c_{jj}(t))\mathbf{s}''_j - \mathbf{T}_{j-1} W''_j(\{\mathbf{v}''\}, \tilde{\mathbf{n}}_j) \mathbf{c}_j \end{aligned} \quad (2.4)$$

where $W''_j(\{\mathbf{v}''\}, \tilde{\mathbf{n}}_j) = [-\tilde{n}_{j1}\mathbf{v}'_1 \cdots -\tilde{n}_{j(j-1)}\mathbf{v}'_{j-1} -\tilde{n}_{j(j+1)}\mathbf{v}'_{j+1} \cdots -\tilde{n}_{jp}\mathbf{v}'_p]$ and

$$c_{jj}(t) = \sum_{k=1, k \neq j}^p \tilde{n}_{jk}(t) |c_{jk}(t)|. \text{ Also, } \mathbf{c}_j = [c_{j1} \ c_{j2} \ \cdots \ c_{j(j-1)} \ c_{j(j+1)} \ \cdots \ c_{jp}]^T$$

$$\text{is adapted by } \dot{\mathbf{c}}_j = \sum_j \text{Proj}(\mathbf{c}_j, W_j^T(\{\mathbf{v}'\}, \tilde{\mathbf{n}}_j)\mathbf{v}'_j) - \sum_j \mathbf{S}_{\mathbf{c}_j}(\mathbf{c}_j)\mathbf{c}_j.$$

The nonnegative function $\tilde{n}_j = [\tilde{n}_{j1} \cdots \tilde{n}_{jp}]^T$ sets the adaptation rate based on the relative distance with its neighbors and the synchronization errors. Note that \tilde{n}_{jp} is a nonzero scalar only if the relative distance is within the maximum communication or sensing distance ($d_{jk} \leq d_{\text{limit},j}$): for $j \neq k$,

$$\tilde{n}_{jp} \left(\|\mathbf{v}'_j - \mathbf{v}'_k\|, d_{jk} \right) = \frac{\tanh(\alpha_j \|\mathbf{v}'_j - \mathbf{v}'_k\|) + \tilde{n}_0}{1 + \tilde{n}_0} \frac{1 - \tanh(\beta_j(d_{jk}^2 - r_{c,j}^2))}{1 + \tanh(\beta_j r_{c,j}^2)} \quad (2.5)$$

Furthermore, each positive element of the diagonal matrix $\mathbf{S}_{\mathbf{c}_j}$ switches to zero if the distance is outside the communication/sensing boundary or the corresponding c_{jk} exceeds the maximum allowable value. This means that outside the communication boundary, the coupling gain exponentially tends to zero.

The closed-loop systems from (2.3) and (2.4) are coupled through a diffusive term in each controller, whose coupling gains are computed by an adaptive control law. Then, the information flow in the network is epitomized by the adaptive graph Laplacian matrix $[\mathbf{L}(t)]_a$ defined $[\mathbf{L}(t)]_a = \text{diag}([c_{11} \cdots c_{pp}]) \otimes \mathbf{I}_n + [\mathbf{c}(t)]$ where

$$[\mathbf{c}(t)] = \begin{bmatrix} 0 & -\tilde{n}_{12}(t)c_{12}(t) & \cdots & -\tilde{n}_{1p}(t)c_{1p}(t) \\ -\tilde{n}_{21}(t)c_{21}(t) & 0 & \cdots & -\tilde{n}_{2p}(t)c_{2p}(t) \\ \vdots & \vdots & \ddots & \vdots \\ -\tilde{n}_{p1}(t)c_{p1}(t) & -\tilde{n}_{p2}(t)c_{p2}(t) & \cdots & 0 \end{bmatrix} \otimes \mathbf{I}_n \quad (2.6)$$

Note that many elements of $[\mathbf{c}(t)]$ are zero, since many pairs of the agents have no directed communication link. Hence, $[\mathbf{L}(t)]_a$ of a large network will inevitably be a sparse matrix.

2.3.3 Main Stability Theorems and Simulation Results

Theorem 2.1 ([10]) *The network EL systems from (2.3) and (2.4) globally exponentially converge to their desired trajectories with bounded errors such that the distance $R_2(t) = \int_{\mathbf{q}'_d}^{(\mathbf{T}_{j-1}^T \mathbf{q}''_j)'} \|\delta z\|$ between each $(\mathbf{T}_{j-1}^T \mathbf{q}''_j)' = [\mathbf{I}_n \ \mathbf{0}_{n \times 1}] \mathbf{T}_{j-1}^T \mathbf{q}''_j$ and the desired trajectory $\mathbf{q}'_d(t)$ exponentially tends to the ball of radius $R_2(t) \leq \frac{\lambda_{\max}(\mathbf{H}(\mathbf{q}))}{\lambda'' k_0 \lambda_{\min}(\mathbf{H}(\mathbf{q}))} (\|\{\Delta_d\}\|)$ with a contraction rate of $k_0/\lambda_{\max}(\mathbf{H}(\mathbf{q}))$ for \mathbf{s}''_j and λ'' for \mathbf{q}''_j , if each diagonal element ℓ_k of $\mathbf{S}_{\mathbf{c}_j}$ satisfies $\ell_k > k_0$ for the design parameter $k_0 > 0$ chosen, and if*

$$k_1(t) \geq k_0 + \min \left(\deg_o \tilde{n}_m c_{\max}/2, \sqrt{2m_n(m_n + \deg_o)} \tilde{n}_m c_{\max} \right) \quad (2.7)$$

Note that c_{\max} denotes the known boundary value of c_{ij} , used for the adaptive control law, and $\deg_o \leq p - 1$ the maximum out-degree of each member (vertex). The out-degree for each vertex defines how many other agents are receiving information from that member. Also, m_n is the maximum in-degree of each system, that is, the maximum number of nonzero elements in each row of $[\mathbf{c}(t)]$.

The results of Theorem 2.1 can also be applied when the adaptive graph Laplacian $[\mathbf{L}(t)]_a$ is augmenting a certain (fixed) baseline digraph constructed by a graph Laplacian $[L(k_1, k_2)]$ in the closed-loop system. For this purpose, the main control law (2.4) can be modified as

$$\begin{aligned} \begin{bmatrix} \mathbf{R}_f \tau_j(t) \\ \tau_{jZ'} \end{bmatrix} &= M''_j(\mathbf{q}_j) \ddot{\mathbf{q}}''_{j,r} + C''_j(\mathbf{q}_j, \dot{\mathbf{q}}_j) \dot{\mathbf{q}}''_{j,r} + G''_j(\mathbf{q}_j) \\ &\quad + D''_j(\mathbf{q}_j, \dot{\mathbf{q}}_j) - (k_1 + c_{jj}(t)) \mathbf{s}''_j + \sum_{k \in N_j} k_2 \mathbf{s}''_k - \mathbf{T}_{j-1} W''_j \mathbf{c}_j \end{aligned} \quad (2.8)$$

where $k_1 > 0, k_2 > 0$. The fixed baseline topology is determined by the set of (incoming) neighbors N_j .

Theorem 2.2 *Adaptive coupling gain augmentation. For the closed-loop system obtained by (2.3) and (2.8), Theorem 2.1 holds by replacing k_1 in (2.7) by $\lambda_{\min}([L(k_1, k_2)]_{\text{sym}})$ by using Weyl's theorem [18].*

We now show that the proposed control law guarantees both faster convergence and smaller error bounds of synchronization between the coupled variables than those of tracking control.

Theorem 2.3 ([10]) *Faster synchronization. A balanced graph with the symmetric Laplacian matrix $[L(t)]_b$ can be constructed for each $[L(t)]_a$ as*

$$[L(t)]_b = \left([L(t)]_a + [L(t)]_a^T \right) / 2 - \text{diag} \left([L(t)]_a^T [1, 1, \dots, 1]^T / 2 \right)$$

If Theorem 2.1 holds, there exists a subset $(\mathbf{V}_{ss}^T \{\mathbf{v}'_{tr}\} = \mathbf{0})$ of the original synchronization manifold, $(\mathbf{V}_s^T \{\mathbf{v}'_{tr}\} = \mathbf{0})$, with \mathbf{V}_{ss} being a subset of orthonormal eigenvectors (\mathbf{V}_s) of $[L(t)]_b$ such that

1. synchronization occurs faster than tracking:

$$\lambda_s = \frac{k_1 + \lambda_{\min}(\mathbf{V}_s^T [L]_{a,\text{sym}} \mathbf{V}_s)}{\lambda_{\max}([1]^T [M'] [1])} > \lambda_t = \frac{k_1}{\lambda_{\max}(\mathbf{V}_s^T [M'] \mathbf{V}_s)} \quad (2.9)$$

2. the synchronization error is smaller than the tracking error if the disturbance field is more co-directional (i.e., $\| [1]^T \mathbf{d}(t) \| > \| \mathbf{V}_{ss}^T \mathbf{d}(t) \|$):

$$\int_0^{\mathbf{V}_s^T \{\mathbf{v}'_{tr}\}} \|\delta \mathbf{y}_s\| \geq \frac{\lambda_{\max}(\mathbf{V}_s^T [M'] \mathbf{V}_s) \|\mathbf{V}_s^T \mathbf{d}(t)\|}{\lambda_{\min}(\mathbf{V}_s^T [M'] \mathbf{V}_s) (k_1 + \lambda_{\min}(\mathbf{V}_s^T [L]_{a,\text{sym}} \mathbf{V}_s))} \quad (2.10)$$

Hence, the control objective in Sect. 2.3.1 is met despite disturbances and modeling errors. Again, the benefit of Theorem 2.3 is that a formation shape on relative elliptical orbits can be maintained more precisely than tracking some desired positions. Moreover, the adaptive graphs of communication or relative sensing connections are automatically determined by synchronization errors and relative distances. Figure 2.7 shows a result of simulation of reconfiguring 275 spacecraft moving in the LVLH relative frame by using (2.1) and the proposed adaptive phase-synchronization control (2.4). Also, the figure shows the changes in the network topology during the reconfiguration maneuver, due to changes in the adaptive graph Laplacian matrix. These results demonstrate that spacecraft can automatically determine in a distributed manner, an evolving digraph topology of a large network of spacecraft swarms based on the synchronization errors and relative distances. The adaptive graph Laplacian matrix represents a time-varying digraph that considers the heterogeneous capabilities of communication or relative sensing among the spacecraft members in the

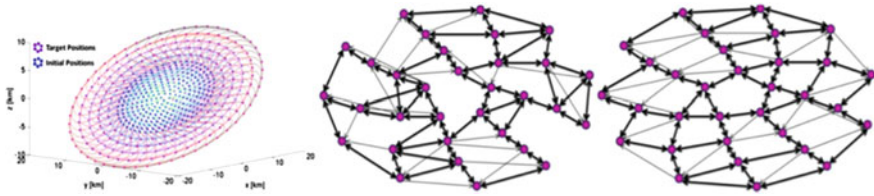


Fig. 2.7 Reconfiguration of 275 spacecraft using the proposed control law (*left*); Change of the network topology when we reconfigure 32 spacecraft (Images taken from [10])

swarm network. The objective of phase synchronization control is to maintain a desired phase difference on a periodic orbit. This implies that the complexity of controlling such a large number of spacecraft moving in relative elliptical orbits reduces to setting a proper phase difference. Interestingly, this method of phase synchronization is conceptually similar to phase synchronization of CPG oscillators that generate phase synchronized rhythms in a self-sustained fashion on spinal cords of vertebrates [12, 30].

2.4 Application of Probabilistic Guidance to Swarms of Spacecraft Operating in Earth Orbit

2.4.1 Introduction

This section reviews the theory behind a probabilistic guidance approach to guiding an arbitrary swarm of agents [1], and reviews its application to coordinating a spacecraft swarm operating in Earth orbit [2]. The main idea is to drive the swarm to a desired density distribution in a prescribed region of the configuration space. Rather than control individual spacecraft, the probabilistic guidance approach controls the average number of spacecraft per unit volume, ensuring that spatial averages converge to the desired targeted density distribution (see Fig. 2.8 for an example scenario). The targeted density distribution is achieved as an emergent behavior of the swarm, and is associated mathematically with achieving a statistical steady-state condition, i.e. a fixed point of a spatial Markov process.

In its simplest form, the probabilistic guidance approach is completely decentralized and does not require communication or collaboration between spacecraft. Specifically, spacecraft make statistically independent probabilistic decisions based solely on their own state, which ultimately guides the swarm toward the desired targeted density distribution. In addition to being completely decentralized, the probabilistic guidance approach has a novel autonomous self-repair property: Once the desired swarm density distribution is attained, the spacecraft automatically repair damage to the swarm distribution without collaborating and without explicit knowl-

Fig. 2.8 In this example scenario, thousands of spacecraft are deployed in Earth orbit. First they match their periods to ensure that they do not quickly drift apart. Then they configure themselves in order to achieve a desired prescribed pattern. The motion of this pattern is determined by the Earth orbital dynamics



edge that damage has occurred. First, the probabilistic guidance method is reviewed for swarms operating in deep space. Then an adaptation of the probabilistic guidance concept relevant to Earth orbiting swarms is reviewed, where orbital dynamics are explicitly considered based on Hill's equations. An illustrative example is given showing theoretical swarm convergence and emergent behaviors.

2.4.2 Probabilistic Guidance Problem

This section describes the swarm distribution guidance problem. The physical domain over which the swarm agents are distributed is denoted as R . It is assumed that region R is partitioned as the union of m disjoint sub-regions, which are referred to as bins:

$$R_i, i = 1, \dots, m, \text{ such that } R = \bigcup_{i=1}^m R_i$$

Let an agent have position $r(t)$ at time index t . Let $x(t)$ be a vector of probabilities such that the sum of its entries is one and the i 'th element $x[i](t)$ is the probability of the event that this agent will be in the i 'th bin R_i at time t ,

$$x[i](t) := \text{Prob}(r(t) \in R_i) \quad (2.11)$$

The time index t will also be referred to as the “stage” in the remainder of the section. Consider a swarm comprised of N agents. Each agent is assumed to act independently

of the other agents, so that (2.11) holds for N separate events,

$$x[i](t) := \text{Prob}(r_k(t) \in R_i), \quad k = 1, \dots, N$$

where $r_k(t)$ denotes the position of the k 'th agent at time index t , and the probabilities of these N events are jointly statistically independent. We refer to $x(t)$ as the swarm distribution. This is to be distinguished from the ensemble of agent positions $\{r_k(t)\}_{k=1:N}$ which, by the law of large numbers, has a distribution that approaches $x(t)$ as the number of agents N is increased.

The distribution guidance problem is defined as follows: Given any initial distribution $x(0)$, it is desired to guide the agents toward a specified steady-state distribution described by a probability vector v ,

$$\lim_{t \rightarrow \infty} x[i](t) = v[i] \text{ for } i = 1, \dots, m$$

$$\text{where } v[i] \geq 0, \quad \sum_{i=1}^m v[i] = 1$$

subject to motion constraints specified in terms of an adjacency matrix A_a as follows:

$$A_a^T[i, j] = 0 \Rightarrow r(t+1) \notin R_i \text{ when } r(t) \in R_j, \forall t.$$

Here, the adjacency matrix A_a of the edges of a directed graph specifies the allowable transitions between bins.

The desired distribution v has the following interpretation: We have m bins in the physical space corresponding to where agents can be located, and the element $v[i]$ is the desired probability of finding an agent in the i 'th bin. If there are N agents, then $Nv[i]$ describes the expected number of agents to be found in the i 'th bin. Let $n = [n[1], \dots, n[m]]^T$ denote the actual number of agents in each bin. Then the number of agents $n[i]$ found in the i 'th bin will generally be different from $Nv[i]$, although it follows from the independent and identically distributed (iid) agent realizations that $v = \mathbb{E}[n]/N$, and from the law of large numbers that $n/N \rightarrow v$ as N becomes large. Hence the vector v is a discrete probability distribution specifying the desired average fraction of agents in each bin of the physical domain, which, in practice, will only be approximated by the histogram n/N of agents. However, the nature of the approximation is that v is equal to the mean $\mathbb{E}[n/N]$ of the agent histogram, and by the law of large numbers, $n/N \rightarrow v$ as N becomes large.

2.4.3 Probabilistic Guidance Algorithm (PGA)

The key idea of the probabilistic guidance law is to synthesize a column stochastic matrix [6, 18] M , which we call Markov matrix for PGA, with v as its eigenvector

corresponding to its largest eigenvalue 1 [16, 18], that is, M must satisfy

$$M \geq 0, \quad \mathbf{1}^T M = \mathbf{1}^T$$

where $\mathbf{1}$ is a vector of ones. The entries of matrix M are defined as transition probabilities. Specifically, for any time instance,

$$M[i, j] = \text{Prob}(r(t+1) \in R_i | r(t) \in R_j), \quad i, j = 1, \dots, m$$

i.e., an agent in bin j transitions to bin i between two consecutive stages with probability $M[i, j]$. The matrix M determines the time evolution of the probability vector x as

$$x(t+1) = Mx(t), \quad t = 0, 1, 2, \dots \quad (2.12)$$

Note that the probability vector $x(t)$ stays normalized in the sense that the sum of its entries is one for all time instances. The probabilistic guidance problem becomes one of designing a specific Markov process (2.12) for x that converges to a desired distribution v .

It is desired for $x(t)$ to asymptotically converge to v , i.e., for v to be a globally attractive stationary distribution for M . The main result of this section shows that asymptotic convergence to v is ensured by imposing an additional constraint on the design of matrix M , denoted as the spectral radius condition,

$$\rho(M - v\mathbf{1}^T) < 1 \quad (2.13)$$

The synthesis of the Markov matrix for PGA can be achieved by using a variety of methods. Methods to synthesize PGA using convex optimization and the “Metropolis-Hastings Algorithm” have been developed in [1]. Once the Markov matrix is synthesized, the following PGA can be used to implement it by providing a copy of the matrix M to each of the agents, and then having each agent propagate its position as an independent realization of the Markov chain as follows:

Probabilistic Guidance Algorithm (PGA)

1. Each agent determines its current bin, e.g., $r_k(t) \in R_i$.
2. Each agent generates a random number z uniformly distributed in $[0, 1]$.
3. Each agent goes to bin j , i.e., $r_k(t+1) \in R_j$, if $\sum_{l=1}^{j-1} M[l, i] \leq z \leq \sum_{l=1}^j M[l, i]$.

The first step determines the agent’s current bin number. The last two steps sample from the discrete distribution defined by the column of M corresponding to the agent’s current bin number.

The following theorem (see [1, 2] for a proof) gives a necessary and sufficient condition for asymptotic convergence of x to v for the generic PGA.

Theorem 2.4 *Consider the PGA below with column stochastic matrix M such that $Mv = v$. Then for any at probability vector $x(0)$, it follows that $x(t) \rightarrow v$ as $t \rightarrow \infty$ for the system in (2.11) if and only if Eq. (2.13) is satisfied.*

Theorem 2.4 is important because it indicates that a probabilistic guidance law for a swarm is asymptotically convergent if and only if the spectral radius condition (2.13) is satisfied. This condition has been interpreted in the context of Perron-Frobenius theory and can be used as a basis for applying modern control theory to synthesizing asymptotically convergent swarm guidance laws in [2].

2.4.4 Adaptation of PGA to Earth Orbiting Swarms

We consider spacecraft swarms in circular orbits around Earth. The dynamics of each spacecraft relative to the circular orbit are given by Hill's equations [31], also referred to as the Clohessy Wiltshire equations,

$$\begin{aligned}\ddot{x} &= 2\omega\dot{y} + 3\omega^2x + f_x \\ \ddot{y} &= -2\omega\dot{x} + f_y \\ \ddot{z} &= -\omega^2z + f_z\end{aligned}$$

Here x , y and z are the spacecraft's local-vertical local-horizontal (LVLH) cartesian coordinates associated with an orbital frame, which is defined at a point on the orbit, and oriented such that the x -axis points in the zenith direction, the z -axis is normal to the orbital plane, and the y -axis completes the right-hand system; f_x , f_y and f_z are the specific forces applied to each axis; and ω is the orbital frequency. From the analytic solution to the Hill's equations it can be shown that the swarm remains bounded if all spacecraft initial states satisfy the following condition (in the absence of any other external forces than the central gravitational field),

$$-6\omega x(0) - 3\dot{y}(0) = 0$$

In this case, all the spacecraft are period matched. From this point on we will only consider spacecraft swarms that are period matched. Furthermore, all spacecraft are assumed to be in-plane, i.e., $z(t) = 0$ for all t . This latter constraint is not necessary but imposed to keep the discussion contained according to the space limitations. The complete generalization of all subsequent results to out-of-plane motion is given in [2]. We now define useful notion of a swarm that is Orbit Type Matched (i.e., an OTM swarm).

Definition 2.1 A swarm is referred to as a planar OTM swarm if it satisfies the following conditions: For all $i = 1, \dots, N$,

1. Period Matched with parameter ω , $\dot{y}_i = -2\omega x_i(0)$

2. Centroid Matched with parameter y_o , $y_{o,i} := y_i(0) - 2\dot{x}_i(0)/\omega = y_o$
3. $z_i(t) = 0$ for all t

A new set of position coordinates is introduced and denoted as Scaled Rotated (SR) coordinates that are instrumental in defining the motion of a planar OTM swarm:

$$\begin{bmatrix} \bar{x}(0) \\ \bar{y}(0) \end{bmatrix} = R^T(\omega t) S^{-1} \begin{bmatrix} x(t) \\ y(t) - y_o \end{bmatrix} \text{ where } R(\omega t) = \begin{bmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

Note that the coordinate transformation is time dependent while the SR coordinates of any spacecraft in a planar OTM swarm are time-invariant. In a planar OTM swarm, there are only two degrees of freedom that determine the motion of each spacecraft. Out SR coordinates naturally capture this fact. Next we can describe the PGA adaptation for in-plane OTM swarms, where each agent follows the following steps:

- Step 1: At t , map current position $(x(t), y(t))$ in LVLH to SR coordinates $r(t) = [\bar{x}(0), \bar{y}(0)]^T$ as,

$$r(t) = R^T(\omega t) S^{-1} \begin{bmatrix} x(t) \\ y(t) - y_o \end{bmatrix}$$

- Step 2: Determine current region R_j s.t. $r(t) \in R_j$, and propagate the Markov Chain one step to calculate the next desired state $r(t+1) \in R_i$ using,

$$M[i, j](t) = \text{Prob}(r(t+1) \in R_i | r(t) \in R_j)$$

- Step 3: Map $r(t+1)$ back to LVLH according to,

$$\begin{bmatrix} x(t+1) \\ y(t+1) - y_o \end{bmatrix} = SR(\omega(t+1))r(t+1)$$

- Step 4: Complete the desired state in LVLH by OTM,

$$\begin{aligned} z(t+1) &= 0 \\ \dot{x}(t+1) &= \frac{\omega}{2}(y(t+1) - y_o) \\ \dot{y}(t+1) &= -2\omega x(t+1) \end{aligned}$$

- Step 5: Command agent to new state $[x, y, z, \dot{x}, \dot{y}, \dot{z}](t+1)$.
- Step 6: $t \leftarrow t+1$, and go back to Step 1.

The idea behind this sequence of steps is to have each agent move according to a Markov chain in SR coordinates. Physically, this corresponds to each agent moving from one Hill's trajectory to another in the plane of motion. Since the statistics of the swarm propagate as the Markov chain, the swarm will converge asymptotically to the desired distribution v in SR coordinates, regardless of the initial distribution. The converged asymptotic distribution in LVLH, in turn, will be a rotated and stretched version of v , corresponding to the time-varying mapping from SR coordinates to

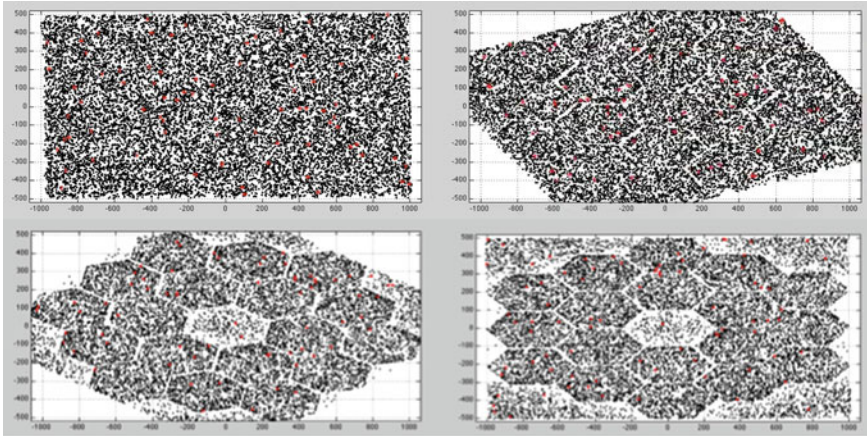


Fig. 2.9 An in-plane OTM swarm of 16,000 spacecraft in a near-circular Earth orbit starts as a uniform distribution and evolves over time into a representative (18 *hexagon*) large aperture pattern

LVLH coordinates. Within this parameterization, PGA can guide the swarm to attain a wide variety of shapes and distributions useful for a diverse range of potential applications. A simulation is performed that demonstrates 16,000 spacecraft in a near-circular Earth orbit. The spacecraft start at a random initial distribution and converge out to a prescribed swarm distribution. Figure 2.9 shows four instances of the swarm density evolution. A concluding remark is given in Sect. 2.6.

2.5 Nonlinear State Estimation And Sensor Optimization Problems for Detection of Space Collision Events

The objective of the collision event detection and tracking is to prevent damaging collisions of currently active LEO satellites with other space objects. This includes efficient detection and tracking of changes in trajectories of Resident Space Objects (RSOs) that might cause the collisions. The collision concept considered here is presented in Fig. 2.10. An important challenge in achieving the collision avoidance objective is the possibility of a short warning time, that is, the time between the change to a collision trajectory and the collision itself may be as short as a few minutes. In this case, we have very stringent requirements on timing of collision RSO's detection and estimation of its new dynamic state values that would allow possible effective avoidance actions.

An effective solution to the collision and avoidance problem is to design and deploy constellations of LEO satellites equipped with EO/IR sensors that can: (1) detect potential changes in existing orbital trajectories that may lead to damaging collisions; and (2) localize, track, and assess trajectories headed towards collisions.

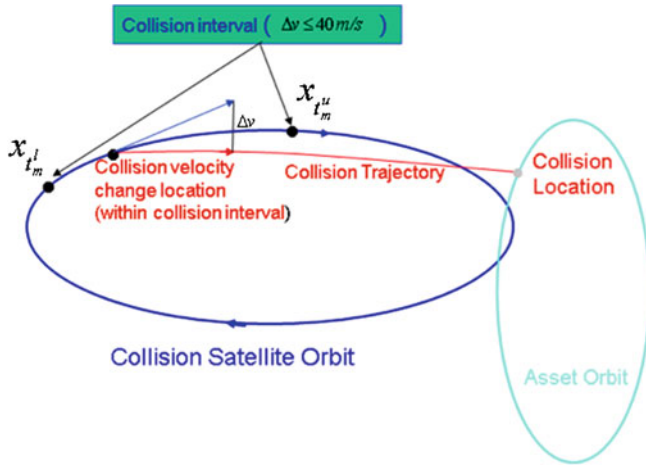


Fig. 2.10 Collision event concept: An event is a collision interval $[t_m^l, t_m^u]$ during which a RSO can change from its current orbit (dark blue) to a collision trajectory (red) that leads to a collision location on the asset orbit (cyan). The collision interval is determined from a limit on the collision satellite's change in velocity (Δv m/s)

We have developed necessary models, simulations, and methods for deriving, evaluating, and comparing such constellations and also derived optimal constellation designs that satisfy the stated objectives for given collision scenarios. In the following sections, we give short summary of our results. In Sect. 2.5.1, we summarize the design of collision event testbed and candidate LEO sensor constellation designs. Satellite collision modeling and algorithms for tracking, collision detection, and sensor management are presented in Sect. 2.5.2.

2.5.1 LEO Sensor Constellation Design and Collision Event Testbed

The scenario that we consider assumes that there are three asset satellites that can possibly collide with 23 other satellites during the day of April 1st of 2006. The three satellites are AQUA, PARASOL, and AURA with the Two Line Elements (TLEs) given from the NORAD catalog (<http://www.space-track.org>). We derive LEO trajectories by using “standard” Simplified General Perturbations-4 (SGP4) propagator [17] that may include higher fidelity orbital perturbations caused by environmental factors such as gravity, atmosphere, and solar pressure and also satellite physical properties such as ballistic coefficient. In a similar way,

EO/IR sensor constellation design: We considered a large number of constellation designs for sensor orbits in order to obtain the best coverage of collision events with minimal number of sensors [34, 35]. We considered sensor constellation designs

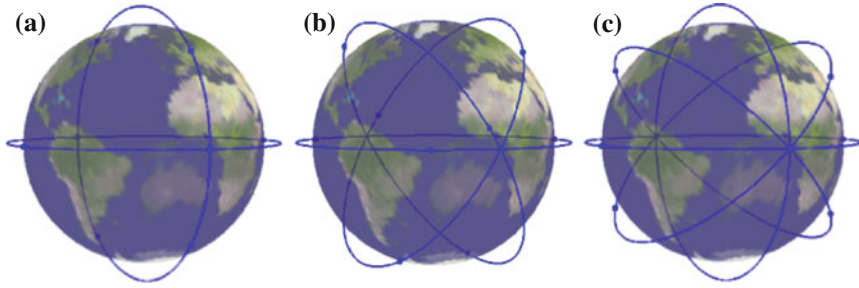


Fig. 2.11 Three EO/IR constellations for collisions event comparisons: design in Fig. 2.11a is shown to be optimal for detecting collisions. **a** $N_p = 2$, $N_{\frac{s}{p}} = 4$. **b** $N_p = 3$, $N_{\frac{s}{p}} = 2$. **c** $N_p = 3$, $N_{\frac{s}{p}} = 3$

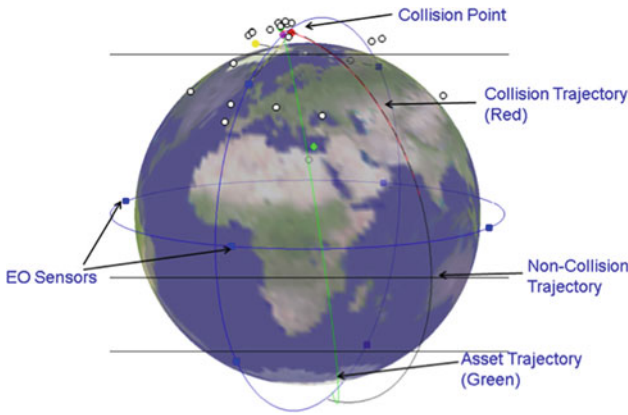


Fig. 2.12 Testbed snapshot showing collision satellite OPS 0203 colliding with satellite PARASOL at 11:26:06 (Hms) with collision trajectory length 21.4 min and collision interval 28.42 min

using symmetry requirement, number of orbital planes N_p , and number of satellites per orbital plane $N_{\frac{s}{p}}$. An example of candidate constellations designs considered in [35] is shown in Fig. 2.11. By considering Space Based Visible (SBV) observations, our analysis reported in [14] demonstrated that it is enough to have four observations of the collision satellite locations after the end of the event interval to determine if the collision related to that event is going to happen or not. This result was derived by assuming that the time between observations is $\Delta t = 50$ s. Testbed simulation snapshot for one of collision events is shown in Fig. 2.12.

2.5.2 Satellite Collision Modeling and Estimation

From a given TLE set, we derive six Kepler parameters (orbital elements) $\xi_t = (\alpha_t, e_t, i_t, \Omega_t, \omega_t, M_t)$ that slowly change with time $t \geq 0$, where the *semi-major axis* a_t and the *eccentricity* e_t , describe the form of the orbit, the mean anomaly M_t defines the position along the orbit, and the three other elements that include the *right ascension of the ascending node* Ω_t , the *inclination* i_t , and the *argument of perigee* ω_t , define the orientation of the orbit in space [26]. The derived state variable is $\xi_t = [\dot{\xi}_t, \xi_t]$ where the first six coordinates are derivatives of the Kepler parameters $\dot{\xi}_t$. More sophisticated stochastic modeling of the satellite states were derived and applied in [36] as extension of simplified state model in [35]. The measurement model of azimuth angle $\alpha_t^{i,j}$ and elevation angle $\beta_t^{i,j}$ looking from j 'th sensor platform toward i 'th collision satellite at time $t \geq 0$, for $\gamma_t^{i,j} = [\alpha_t^{i,j}, \beta_t^{i,j}]$, is given by

$$\tilde{\gamma}_t^{i,j} = \gamma_t^{i,j} + \sigma_a v_t^{i,j}, t \geq 0,$$

where the error standard deviation is $\sigma_a = 1$ arcsec, and $v_t^{i,j} \tilde{N}(0, I_2)$ are normalized Gaussian variables.

A measurement is collected by a sensor only if the following three conditions are satisfied: (a) the sensor points its CCD array toward the predicted location of the target; (b) the true location of the target is within the sensor field of view (FoV); and (c) the target has “star background” looking from the sensor. Condition (c) is necessary for accurate sensor attitude determination.

For each event interval, we collect observations to determine if the collision satellite has changed its trajectory to any one of the candidate collision trajectories. Since for realistic bound on change in velocity (usually denoted by and called “delta v”), the collision can happen only on a small part of the asset's orbit and therefore, as an realistic approximation, we choose one point on the asset orbit as a location of the collision that then corresponds to a unique intercept time. Then for any given time inside the event interval, by using the Generalized Parametrized Battin (GPB) method presented in [35] and derived from [5], we can determine a unique collision trajectory that is parametrized by ξ (e.g. Kepler elements). By discretizing the event interval at every second, we have a finite number of possible collision trajectories that we denote by ξ_l^q and its corresponding hypothesis by H_l^q , where superscript q denotes event (event index) and subscript l is an index of hypothesis for each event. As time progresses (in our case at each second), we are creating hypotheses corresponding to candidate collision trajectories. The zero hypothesis ($l = 0$) corresponds to the non-collision trajectory. Each hypothesized trajectory is initialized with an appropriate uncertainty matrix to account for time discretization approximation of the event interval. Also, each hypothesis is tracked by applying a given tracking algorithm, e.g. see a version described in Sect. 2.5.2.2. As observations are collected, we calculate posterior probability of each hypothesis by applying the collision detection algorithm described in Sect. 2.5.2.2. The hypotheses with very small

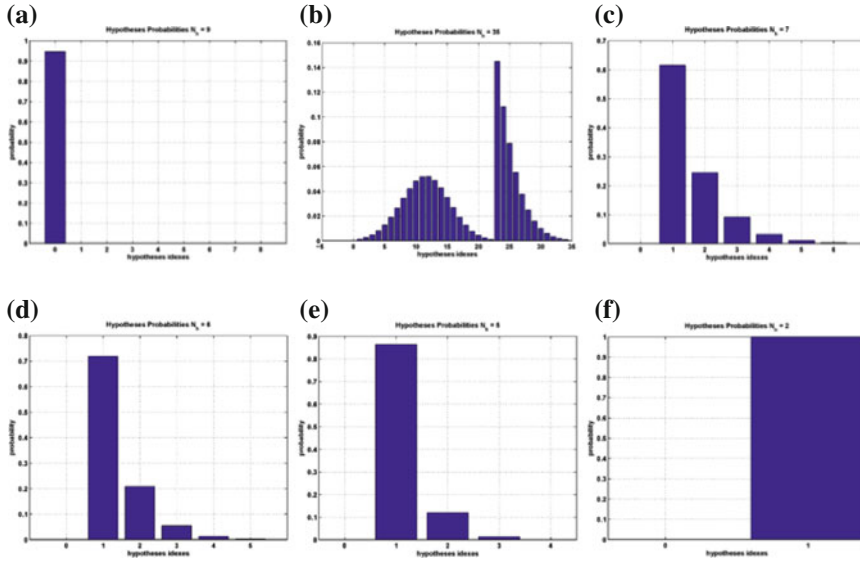


Fig. 2.13 Hypotheses posteriors at each scan (50 s between scans) after the collision satellite moved from its non-collision trajectory (hypothesis 0) to a collision trajectory (hypothesis 1): (a) after $t = 8$ s, collision probability is 0.055, i.e. collision threat is not detected; (b) after $t = 58$ s, collision probability is one, i.e. collision threat is detected; (c) after $t = 108$ s, collision trajectory is singled out (probability ≥ 0.6); (d) after $t = 158$ s, number of hypotheses decreases and true collision trajectory probability is ≥ 0.7 ; (e) after $t = 208$ s, true collision trajectory probability is > 0.8 ; (f) after $t = 258$ s, true trajectory probability is one and all other hypotheses disappeared, i.e. collision trajectory is uniquely determined

probabilities are discarded and ones with higher probabilities are kept till they are fully resolved as shown in Fig. 2.13. The sensors management algorithm is discussed in Sect. 2.5.2.3.

2.5.2.1 Tracking Algorithm

A simplified version of state perturbation model and its corresponding filter are derived and demonstrated in [35]. The more advanced stochastic model of state perturbations, i.e. stochastic target state model, with nonlinear particle filter based tracking and estimation algorithm is derived and demonstrated in [36].

Tracking algorithms developed and demonstrated in [14, 33–36] consists of the following five steps:

- Step 1: Initialization
- Step 2: Derivation of prediction state estimate
- Step 3: Estimation of optimal sensor parameters (Sensor Management and Tracking)
- Step 4: Collection of measurements
- Step 5: Updating of state estimates

Nonlinear state and measurement equations are used in posterior propagation and update using Particle Filtering algorithms involving propagation of particle states and update of particle weights, respectively. The selected tracking results are summarized in Fig. 2.15.

2.5.2.2 Collision Detection Algorithm

The collision detection algorithm is based on multi-hypothesis testing, using likelihood derivation and initialization [14] as follows:

1. At time step k , we create $N_k^{h,q}$ hypotheses for each current event q .
2. For event q at time step k , we have hypothesis H_l^q that is associated with element set $\xi_k^{q,l}$.
3. From the measurement model, we derive hypotheses likelihoods:

$$\log p(t, \tilde{\gamma}_t^{i,j} | H_l^q) \sim -\frac{1}{2\sigma_a^2} \|\tilde{\gamma}_t^{i,j} - \gamma_t^j(\xi_t^{q,l})\|^2$$

where $\gamma_t^j(\xi_t^{q,l})$ is an azimuth-elevation pair associated with hypothesis H_l^q and $p(t, \cdot | H_l^q)$. conditional probability of azimuth-elevation pairs given hypothesis H_l^q at time $t \geq 0$

4. We derive hypotheses conditional posterior $p(t, H_l^q | \gamma_t)$ given all measurements Y_t up to the current time $t \geq 0$ by using Bayes' Theorem and then select a set of hypotheses that have posteriors above a given threshold.

Selected results from [14] are shown in Fig. 2.13 demonstrating the effectiveness of the collision detection algorithm. Assuming 50s between any two consecutive observations (SBV scans), plots (a) and (b) in Fig. 2.13 demonstrate that it takes two measurements to get that probability of non-maneuver to be zero (hypothesis indexed by zero in Fig. 2.13). The rest of the plots (c)–(f) of Fig. 2.13 demonstrate that it takes extra two to four measurements (four to six from the change of the trajectory) to uniquely determine the collision trajectory (hypothesis index one).

2.5.2.3 Sensor Management Algorithm

The details of sensor management algorithms and its effectiveness for LEO collision detection and tracking using a SBV sensor platform are described in [14]. The algorithm is extended to disparate and dispersed sensors (radars and SBV platforms) and also its effectiveness is demonstrated for continuous tracking of LEO satellites in [33]. Our sensor management algorithm is based on maximization of the Posterior Expected Number of Targets of Interest (PENTI) objective function $f_k(\cdot)$ that is information-theoretic representation of the expected number of well localized

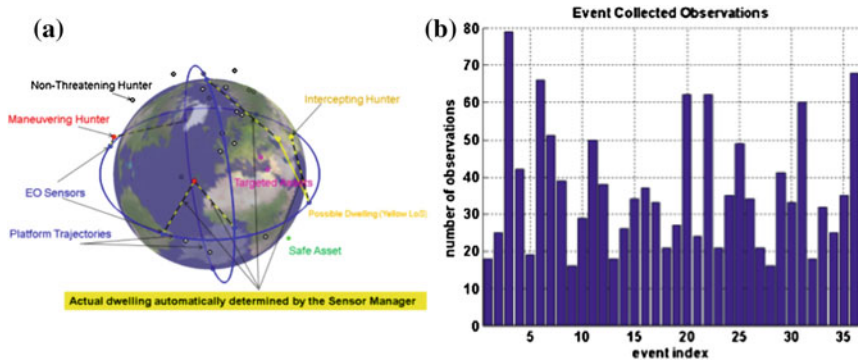


Fig. 2.14 Results of sensor management for collision detection and tracking: (a) simulation snapshot example of a typical LEO complex environment; and (b) number of observations per event

targets that are of tactical interest [26]. The sensor management is reduced to finding pairs $(j_1, l_1), \dots, (j_n, l_n)$ chosen from available sensors (j_1, \dots, j_n) and targets (l_1, \dots, l_n) which are a subset of all collision satellites of interest.

The sensor management problem then consists in determining the pointing angles $(\gamma_{k,j_1}, \dots, \gamma_{k,j_n})$ that are approximate solution of the following optimization problem:

$$(\gamma_{k,j_1}, \dots, \gamma_{k,j_n}) = \arg \max_{\gamma_{j_1}, \dots, \gamma_{j_n}} f_k(\gamma_{j_1}, \dots, \gamma_{j_n}).$$

The sensor management simulation results are shown in plots (a) and (b) of Fig. 2.14. Plot (a) of Fig. 2.14, is a snapshot of LEO complex environment that consists of EO/IR sensor platforms (blue squares), assets that are not currently in danger of collisions (green diamonds), assets that are in danger of being intercepted (magenta diamonds), collision satellites that are in event interval (red circles), collision satellites that are not currently associated with any event (white circles), and collision satellites that might be on collision trajectories (yellow circles). Possible observation collections and actual observations are represented by yellow and dash black lines. In plot (b) of Fig. 2.14, we show the total number of collected measurements for each event.

We implemented and demonstrated multisensor-multitarget sensor management and multi-hypothesis based collision detection and tracking on the SSCI's LEO environment testbed [35]. The simulation results for sensor constellation in plot (a) of Fig. 2.11, are shown in plots (a)–(c) of Fig. 2.15. Note that the standard deviations in plot (a) of Fig. 2.11 are very low (shades of blue) for all collision satellites during collision intervals which indicates good observability of the collision events and therefore excellent sensor management. Oscillatory property of location and velocity errors indicates an accurate modeling of state dynamics [35]. All collision events are tracked with high accuracy and on a timely basis for effective collision avoidance. A conclusion is given in Sect. 2.6.

2.6 Conclusion

This chapter discusses new guidance and control technologies for planetary landing and guidance, navigation, estimation, and control for swarms of spacecraft in low Earth orbit. In Sect. 2.2, we compared the Lander Vision System (LVS) and Autonomous Landing and Hazard Avoidance Technology (ALHAT) systems for planetary landing applications. The LVS has less challenging requirements because of the day-time landing, small size of the robotic lander, and the significant hazard tolerance of the lander. These enabled a system that could perform TRN with the mature computer vision algorithms and other sensor measurements such as a single flash LIDAR image. On the other hand, the requirements on the ALHAT system are more challenging. The ALHAT system must detect hazards from far away at high resolution over a large area to safely land the crewed lunar lander. This resulted in the design of a gimbaled flash LIDAR and the associated increase in complexity, mass, and power. In Sect. 2.3, the new formation control and phase synchronization strategy for swarms of spacecraft was discussed, in which orbital and attitude motions could be modeled as coupled Lagrangian systems moving in elliptical periodic orbits. The adaptive control strategy of automatically computing evolving network topologies eliminated the need for defining a fixed communication or relative sensing topology on a digraph for synchronization stability. Such an evolving communication network gave rise to the adaptive graph Laplacian matrix. The error bound of the proposed synchronization control law with an adaptive graph Laplacian was shown to be smaller than that of an uncoupled tracking control law. This justified the use of a synchronization framework, for application where synchronization errors should be kept smaller than tracking errors, like stellar interferometry applications. In Sect. 2.4, the Probabilistic Guidance Algorithm (PGA) was applied to the problem of guiding swarms of spacecraft, operating under dynamic constraints imposed by being in Earth orbit. The main simplifying assumption was that all agents have nearly circular orbits and they obey Hill's linearized equations of motion. A simulation example showed the basic feasibility of the method. Due to space limitation, the discussion was restricted to in-plane motion only, but references have been provided that generalized all results to the out-of-plane case. Section 2.5 demonstrated that it is possible

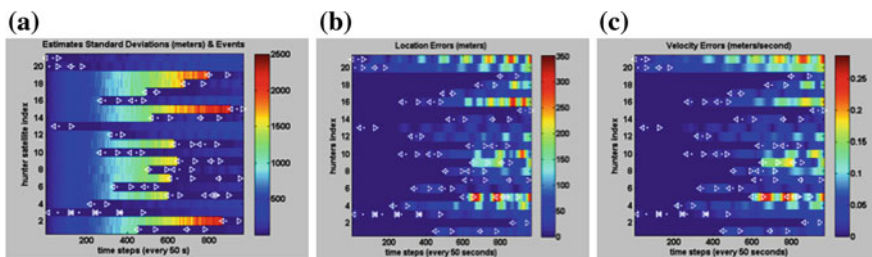


Fig. 2.15 Collision events (*left triangles and dots* represent event intervals and *right triangle* represents approximate intercept time) tracking and detection results: (a) estimation standard deviations; (b) location errors; and (c) velocity errors for all satellites

to detect and avoid space collision events using a small number of space-based EO/IR sensors. We implemented realistic scenarios and models for LEO collisions, developed appropriate metrics for the evaluation of different EO/IR sensor constellations, and evaluated the tracking and sensor management performance for different LEO EO/IR sensor constellations. The constellation that is designed on two orbital planes with four satellites on each offers the best compromise between the number of satellites and overall performance. The developed capabilities are expected to lead to significant improvement in Space Situational Awareness (SSA). The Space Based Visible sensors/constellations can be used for enhancing the capabilities of Space Surveillance Network (SSN). Future developments of an effective space surveillance system can utilize a swarm of spacecraft. The algorithms developed during our study will be applicable to those designs and lead to cost-effective implementations.

Acknowledgments The research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. The following people are thanked for their contributions: Saptarshi Bandyopadhyay and Daniel Morgan at UIUC; Milan Mandic at JPL; and Adel El-Fallah and Aleksandar Zatezalo at SSCI.

References

1. Acikmese, B., Bayard, D.: A markov chain approach to probabilistic swarm guidance. *Am. Control Conf. (ACC)* **2012**, 6300–6307 (2012). doi:[10.1109/ACC.2012.6314729](https://doi.org/10.1109/ACC.2012.6314729)
2. Acikmese, B., Bayard, D.S.: Probabilistic guidance for swarms of autonomous agents. Technical Report D-73778, Jet Propulsion Laboratory, JPL (2012)
3. Adams, D., Criss, T., Shankar, U.: Passive optical terrain relative navigation using apnav. In: *Aerospace Conference, 2008 IEEE*, pp. 1–9 (2008). doi:[10.1109/AERO.2008.4526303](https://doi.org/10.1109/AERO.2008.4526303)
4. Amzajerjian, F., Pierrotet, D., Petway, L., Hines, G., Roback, V.: Lidar systems for precision navigation and safe landing on planetary bodies. In: *Proceedings of the SPIE*, vol. 8192, pp. 819, 202–819, 202–207 (2011). doi:[10.1117/12.904062](https://doi.org/10.1117/12.904062). <http://dx.doi.org/10.1117/12.904062>
5. *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA education series. American Institute of Aeronautics and Astronautics, Reston (1999). <http://opac.inria.fr/record=b1096182>
6. Berman, A., Plemmons, R.J.: *Nonnegative Matrices in the Mathematical Sciences*. SIAM, Philadelphia (1994)
7. Chang, I., Chung, S.J., Blackmore, L.: Cooperative control with adaptive graph laplacians for spacecraft formation flying. In: *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 4926–4933 (2010). doi:[10.1109/CDC.2010.5717516](https://doi.org/10.1109/CDC.2010.5717516)
8. Cheng, Y., Clouse, D., Johnson, A., Owen, W., Vaughan, A.: Evaluation and improvement of passive optical terrain relative navigation algorithms for pin-point landing. In: *AAS Space Flight Mechanics Meeting (AAS-SFM 2011)* (2011)
9. Chung, S.J., Ahsun, U., Jacques, E., Slotine, J.: Application of synchronization to formation flying spacecraft: Lagrangian approach. *J. Guid. Control Dyn.* **32**(2), 512–526 (2009)
10. Chung, S.J., Bandyopadhyay, S., Chang, I., Hadaegh, F.Y.: Phase synchronization control of complex networks of lagrangian systems on adaptive digraphs. *Automatica* **49**(5), 1148–1161 (2013). doi:[http://dx.doi.org/10.1016/j.automatica.2013.01.048](https://doi.org/10.1016/j.automatica.2013.01.048). <http://www.sciencedirect.com/science/article/pii/S0005109813000496>
11. Chung, S.J., Hadaegh, F.Y.: Swarms of femtosats for synthetic aperture applications. In: *4th International Conference on Spacecraft Formation Flying Missions and Technologies (SFFMT)*. St-Hubert, Quebec, Canada (2011)

12. Chung, S.J., Slotine, J.J.: On synchronization of coupled hopf-kuramoto oscillators with phase delays. In: Decision and Control (CDC), 2010 49th IEEE Conference on, pp. 3181–3187 (2010). doi:[10.1109/CDC.2010.5717962](https://doi.org/10.1109/CDC.2010.5717962)
13. Chung, S.J., Slotine, J.J.E.: Cooperative robot control and concurrent synchronization of lagrangian systems. *Trans. Rob.* **25**(3), 686–700 (2009). doi:[10.1109/TRO.2009.2014125](https://doi.org/10.1109/TRO.2009.2014125). <http://dx.doi.org/10.1109/TRO.2009.2014125>
14. El-Fallah, A., Zatezalo, A., Mahler, R., Mehra, R.K., Pham, K.: Joint search and sensor management of space based eo/ir sensors for leo event estimation. In: Proceedings of the SPIE, vol. 7330, pp. 73,300N–73,300N–12 (2009). doi:[10.1117/12.818292](https://doi.org/10.1117/12.818292). <http://dx.doi.org/10.1117/12.818292>
15. Epp, C., Robertson, E., Brady, T.: Autonomous landing and hazard avoidance technology (alhat). In: Aerospace Conference, 2008 IEEE, pp. 1–7 (2008). doi:[10.1109/AERO.2008.4526297](https://doi.org/10.1109/AERO.2008.4526297)
16. Fiedler, M.: Special Matrices and Their Applications in Numerical Mathematics. Dover, Mineola (2008)
17. Hoots, F.R., Schumacher, P.W., Glover, R.A.: History of analytical orbit modeling in the u. s. space surveillance system. *J. Guid. Control Dyn.* **27**(2), 174–185 (2004). doi:[10.2514/1.9161](https://doi.org/10.2514/1.9161). <http://dx.doi.org/10.2514/1.9161>
18. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, New York (1985)
19. Huertas, A., Cheng, Y., Madison, R.: Passive imaging based multi-cue hazard detection for spacecraft safe landing. In: Aerospace Conference, 2006 IEEE, pp. 14 (2006). doi:[10.1109/AERO.2006.1655794](https://doi.org/10.1109/AERO.2006.1655794)
20. Johnson, A., Golombek, M.: Lander vision system for safe and precise entry descent and landing. In: LPI Workshop on Concepts and Approaches for Mars Landing (2012)
21. Johnson, A., Huertas, A., Werner, R., Montgomery, J.: Analysis of on-board hazard detection and avoidance for safe lunar landing. In: Aerospace Conference, 2008 IEEE, pp. 1–9 (2008)
22. Johnson, A., Ivanov, T.: Analysis and testing of a lidar-based approach to terrain relative navigation for precise lunar landing. In: AIAA Guidance Navigation and Control Conference (AIAA-GNC 2011) (2011)
23. Johnson, A., Keim, J., Ivanov, T.: Analysis of flash lidar field test data for safe lunar landing. In: Aerospace Conference, 2010 IEEE, pp. 1–11 (2010). doi:[10.1109/AERO.2010.5447025](https://doi.org/10.1109/AERO.2010.5447025)
24. Johnson, A., Montgomery, J.: Overview of terrain relative navigation approaches for precise lunar landing. In: Aerospace Conference, 2008 IEEE, pp. 1–10 (2008). doi:[10.1109/AERO.2008.4526302](https://doi.org/10.1109/AERO.2008.4526302)
25. Johnson, A., Willson, R., Cheng, Y., Goguen, J., Leger, C., Sanmartin, M., Matthies, L.: Design through operation of an image-based velocity estimation system for mars landing. *Int. J. Comput. Vis.* **74**(3), 319–341 (2007). doi:[10.1007/s11263-006-0022-z](https://doi.org/10.1007/s11263-006-0022-z). <http://dx.doi.org/10.1007/s11263-006-0022-z>
26. Montenbruck, O., Gill, E.: Satellite Orbits, Models, Methods, and Applications. Springer, New York (2005)
27. Morgan, D., Chung, S.J., Blackmore, L., Acikmese, B., Bayard, D., Hadaegh, F.Y.: Swarm-keeping strategies for spacecraft under J2 and atmospheric drag perturbations. *J. Guid. Control Dyn.* **35**(5), 1492–1506 (2012). doi:[10.2514/1.55705](https://doi.org/10.2514/1.55705). <http://dx.doi.org/10.2514/1.55705>
28. Mourikis, A., Trawny, N., Roumeliotis, S., Johnson, A., Ansar, A., Matthies, L.: Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Trans. Robot.* **25**(2), 264–280 (2009). doi:[10.1109/TRO.2009.2012342](https://doi.org/10.1109/TRO.2009.2012342)
29. Poberezhskiy, I., Johnson, A., Chang, D., Ek, E., Natzic, D., Spiers, G., Penniman, S., Short, B.: Flash lidar performance testing: configuration and results. In: SPIE Laser Radar Technology and Applications XVII (2012)
30. Seo, K., Chung, S.J., Slotine, J.J.: Cpg-based control of a turtle-like underwater vehicle. *Auton. Robots* **28**(3), 247–269 (2010). doi:[10.1007/s10514-009-9169-0](https://doi.org/10.1007/s10514-009-9169-0). <http://dx.doi.org/10.1007/s10514-009-9169-0>
31. Vallado, D., Clain, W.: Fundamentals of Astrodynamics and Applications. College custom series. McGraw-Hill, New York (1997). <http://books.google.com/books?id=HuWMPwAACAAJ>

32. Villalpando, C., Johnson, A., Some, R., Oberlin, J., Goldberg, S.: Investigation of the tilera processor for real time hazard detection and avoidance on the altair lunar lander. In: Aerospace Conference, 2010 IEEE, pp. 1–9 (2010). doi:[10.1109/AERO.2010.5447023](https://doi.org/10.1109/AERO.2010.5447023)
33. Zatezalo, A., El-Fallah, A., Mahler, R., Mehra, R.K., Brown, J.: Dispersed and disparate sensor management for tracking low earth orbit satellites. In: Proceedings of the SPIE, vol. 7336, pp. 73,360I–73,360I–12 (2009). doi:[10.1117/12.819299](https://doi.org/10.1117/12.819299). <http://dx.doi.org/10.1117/12.819299>
34. Zatezalo, A., El-Fallah, A., Mahler, R., Mehra, R.K., Pham, K.: Optimal constellation design of low earth orbit (leo) eo/ir sensor platforms for space situational awareness. In: Proceedings of the SPIE, vol. 7330, pp. 73,300T–73,300T–11 (2009). doi:[10.1117/12.818304](https://doi.org/10.1117/12.818304). <http://dx.doi.org/10.1117/12.818304>
35. Zatezalo, A., El-Fallah, A., Mahler, R., Mehra, R.K., Pham, K.: Eo/ir satellite constellations for the early detection and tracking of collision events. In: Proceedings of the SPIE, vol. 7697, pp. 76,970L–76,970L–12 (2010). doi:[10.1117/12.851217](https://doi.org/10.1117/12.851217). <http://dx.doi.org/10.1117/12.851217>
36. Zatezalo, A., El-Fallah, A., Mahler, R., Mehra, R.K., Pham, K.D.: Multimodel filtering of partially observable space object trajectories. In: Proceedings of the SPIE, vol. 8050, pp. 80, 500K–80,500K–12 (2011). doi:[10.1117/12.884609](https://doi.org/10.1117/12.884609). <http://dx.doi.org/10.1117/12.884609>

Advances in Control System Technology for Aerospace
Applications

Feron, E. (Ed.)

2016, XII, 180 p. 75 illus., 65 illus. in color., Softcover

ISBN: 978-3-662-47693-2