12

Artificial Intelligence for Space Applications

Daniela Girimonte and Dario Izzo

European Space Agency, Advanced Concepts Team, ESTEC, EUI-ACT,
Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands,
daniela.girimonte@esa.int, dario.izzo@esa.int

Summary. The ambitious short-term and long-term goals set down by the various national space agencies call for radical advances in several of the main space engineering areas, the design of intelligent space agents certainly being one of them. In recent years, this has led to an increasing interest in artificial intelligence by the entire aerospace community. However, in the current state of the art, several open issues and showstoppers can be identified. In this chapter, we review applications of artificial intelligence in the field of space engineering and space technology and identify open research questions and challenges. In particular, the following topics are identified and discussed: distributed artificial intelligence, enhanced situation self-awareness, and decision support for spacecraft system design.

12.1 Introduction

In the second half of 2003, the European Space Agency (ESA) delivered a roadmap, in the framework of the Aurora program, to bring humans to explore Mars within the next few decades [MO03]. The plan included the successful implementation of several flagstone missions as stepping stones for achieving this final ambitious goal. A few months later, with the vision delivered by U.S. president George W. Bush, the National Aeronautics and Space Administration (NASA) also started to draft plans for the manned exploration of Mars [Bus04]. Their vision included the establishment of a human base on the Moon, among several other advanced preparatory steps. The return of humans to the Moon and a future manned mission to Mars therefore seem to be likely achievements we may witness in the next few decades. At the same time, even more ambitious plans and missions are being conceived by farsighted researchers who dream about the exploration and colonization of even farther planets.

In the framework of these more or less concrete future scenarios, the consolidation of artificial intelligence methods in space engineering is certainly an enabling factor. As an example, the reader may think of a future mission to Mars. This will probably be constituted by a large number of heterogeneous space agents (intended to be satellites, humans, robots, modules, sensors, and so on). In such a scenario, the round-trip communication delay time, depending on the relative positions of Mars and Earth, would range from 6.5 minutes to 44 minutes approximately. Besides, communication with Earth would not be possible at all during a 14 day period every Mars synodic period (approximately 2.1 years). Clearly, for such a mission to happen, the single space agents must be able to make autonomous decisions, to interact harmoniously with each other, and to be able to determine their own health status so as to properly plan their actions. Unfortunately, if we take a look at the current state of the art of artificial intelligence applications in space engineering, we can identify several open issues and showstoppers. Actually, it seems that we are far away from the desirable situation in which these methods can be considered as off-the-shelf tools available to space engineers.

This chapter is addressed to the artificial intelligence community in order to create an awareness of the many open research questions and challenges in the space engineering community. In order to achieve this task, the chapter focuses on a few niche applications only, namely distributed artificial intelligence for swarm autonomy and distributed computing, and enhanced situation self-awareness and decision support for spacecraft system design. Our survey aims to give the reader a general overview of these topics by pointing out some of the relevant activities within the international space community and as such is not intended to cover the entire array of all artificial intelligence applications in space. For example, we deliberately omitted in this discussion research on automated planning and scheduling, which is traditionally the most studied field within artificial intelligence for space, and we refer interested readers to other resources such as the proceedings of the International Workshop on Planning and Scheduling for Space (e.g., 1997, 2000, 2004, and 2006).

12.2 Distributed Artificial Intelligence

At the end of the 1980s, the artificial intelligence community started wondering whether intelligence had to be strictly related to reasoning. Failures in constructing agents able to interact with the environment in real-time following high-level decisions derived via symbolic reasoning led to a new approach in the design of robot control systems: "behavior based" robotics [Bro91]. Starting from the simple observation that most of what we do in our daily lives is not related to detailed planning but rather to instinctual reactions to an unstable and changing environment, behavioral robotics introduced, for the first time, the notion of "emerging" intelligence. Researchers were forced to observe that, in some systems, intelligence could emerge from the interaction with the environment and from indirect relations between system parts and that, in general, intelligence could not always be easily located in one particular part of the system studied. The idea of intelligence residing in a distributed form throughout an agent started the study of intelligent systems made by more than one agent. Hence, "distributed artificial intelligence" (DAI) developed as a discipline studying systems made up of a number of diverse agents that despite their individuality are able to achieve common global tasks. In the following sections, we mainly touch upon two topics of DAI systems for space applications: swarm intelligence and distributed computing.

12.2.1 Swarm Intelligence

There is no common agreement on the definition of swarm intelligence. Definitely a subcategory of distributed artificial intelligence, we define swarm intelligence as the emerging property of systems made by multiple identical and noncognitive agents characterized by limited sensing capabilities. This definition, nearly a description of biological swarm intelligence, stresses the necessity of having agents that interact locally with the environment and between themselves. It may be argued that algorithms historically considered at the center of swarm intelligence research, such as "particle swarm optimization" (PSO) [KE95], sometimes lack this property, which therefore should not be required in the definition.[1] Others would instead take the opposite direction in requiring that the local interaction happen only indirectly through an intentional modification of the environment (stigmergy [TB99]). Other issues arise when trying to decide whether deterministic systems should be excluded from the definition of swarm intelligence. These pages may not be a suitable place to settle or discuss these issues, and we therefore ask the reader to be forgiving should our view not be fully satisfactory to him or her.

Whatever definition one wishes to adopt, a number of features of swarm intelligence are certainly attractive to the space engineering community. The space environment typically puts stringent constraints on the capabilities of single satellites, robots, or anything that needs to survive in space (space agents). Space agents are particularly limited in terms of mobility (propellant- and power-limited), communication (power-limited), and size (mass-limited). At the same time, a high level of adaptability, robustness, and autonomy is required to increase the chances of success of operating in a largely unknown environment. Similar characteristics are found in the individual components of a biological swarm. Moreover, a number of space applications are naturally based on the presence of multiple space agents. The first commercial application proposed and realized for satellite systems was that of Arthur C. Clarke and was a satellite constellation providing global communication services by means of three satellites put in a geostationary orbit [Cla45]. Since then, a large number of constellations have been deployed to provide global communication, navigation, and Earth observation services.

[1] The so-called social component in the PSO algorithm requires at each step for each agent to know the best solution found so far by the entire swarm. Interagent communication is, in this case, direct and unlimited in range.

More recently, the idea of a number of satellites flying in formation has been used in a number of missions for applications ranging from x-ray astronomy (XEUS) to differential measurements of the geomagnetic field (CLUSTER II), space interferometry, the search for exoplanets (DARWIN), and others. All these missions [2] are able to meet their requirements without making use of an emerging property that can be regarded as swarm intelligence. On the other hand, if available, swarm intelligence methods would represent an attractive design option allowing, for example, achievement of autonomous operation of formations. Simple agents interacting locally could be considered as a resource rather than overhead. At the same time, one would be able to engineer systems that are robust, autonomous, adaptable, distributed, and inherently redundant. Besides, swarms allow for mass production of single components, thus promising mission cost reduction, and represent highly stowable systems, thus allowing reduced launch costs. Recently, these motivations led a number of researchers to simulate some degree of swarm intelligence in a number of space systems and to investigate their behavior.

Kassabalidis et al. [KEM+01] studied the routing problem in wireless communication networks between satellites or planetary sensors. He applied ant-inspired algorithms to achieve a great efficiency in networks that are spatially distributed and changing over time. This type of research is targeted at applications such as those being developed by the NASA sensorweb project [CCD+05]. Distributed cooperative planning between satellites belonging to the same constellation has also been studied, introducing swarm intelligence at the level of coordinated planning [DVC05] (for a typical case study, see Fuego, studied by Escorial et al. [ETR+03]). Recent work on intersatellite communication in constellations observed the birth of emerging properties from a more or less complex system of rules and behaviors [BT07] programmed in the autonomous planners onboard the satellites. More generally, any problem of autonomy for satellite constellations is a problem of distributed artificial intelligence, where the possibility of communication between agents (ISL-intersatellite links) or between an agent and a central planner (ground station) is limited by the complex dynamics of the system and by the agent design.

Another field where swarm intelligence provides a possibility to improve current technology is that of relative satellite motion control. When a system of many satellites has to move in a coordinated way, the control action selected by each satellite may take into account the decisions made by the others at different levels. The information exchanged with the other swarm components is useful but not necessary to define the geometric and kinematical representation of the time-varying environment, which will then influence the satellite action selection. Many studies dealing with terrestrial robot navigation [Kha86], with spacecraft proximity and rendezvous operations [McI95], and self-assembly structures in space [McQ97] have taken the ap-

[2] At the time of writing, CLUSTER II is the only one operational.

proach of defining an artificial potential field to model the environment. The control action is then chosen to follow the steepest descent of the defined potential. Another approach to the action selection problem, based on dynamic systems theory, was introduced by Schoner [SD92]. In this approach, the state-space contains behavioral variables such as heading directions or velocities. All the contributions given by each behavior are combined by means of weighting parameters into a final dynamical system that defines the course of behaviors that each agent will follow. The weighting parameters can be evaluated by solving a competitive dynamic system operating on a faster timescale. Recently, other approaches have been proposed, in particular for space applications, attempting to obtain some degree of decentralized coordination in a group of satellites. Lawton and Beard [RB04, LYB03] introduced what they call a "virtual structure" method to design a decentralized formation control scheme. Their method aims at reaching a unique final configuration in which each satellite has its position preassigned. When a swarm of homogeneous agents is considered and the task is given to acquire a certain final geometry, the final positions occupied by each agent in the target configurations should be chosen in an autonomous way and should be part of the global behavior emerging from the individual tasks assigned. This result is actually possible using a technique [IP07] developed at ESA and inspired by swarm aggregation results [Gaz05] for terrestrial robots. Introducing a behavioral component that accounts for the differential gravity typical of orbital environments, the algorithm allows one to obtain, in a given countable number of final formations, a swarm whose emerging behavior is the solution of the target allocation problem and the acquisition and maintenance of the final formation.

Figure 12.1 illustrates two examples of orbital swarms controlled by this algorithm with the addition of a limited amount of hierarchy in the swarm to allow the lattice formation [IP07]. Note that, in the first case, the swarm autonomously selects its final arrangement from $2.81 \cdot 10^{41}$ different final possible configurations. In the second case, this number is $4.16 \cdot 10^{29}$. A behavior-based control approach for satellite swarms has also been shown to be useful in controlling highly nonlinear systems such as those derived by introducing electrostatic interactions between swarm agents [PIT06].

12.2.2 Distributed Computing

A second example of distributed artificial intelligence with specific applications to space systems, and in particular to trajectory design [IM05], is that of distributed computing. The possibility of sharing the memory and the computing resources of a large network of simple computers is clearly appealing for any kind of application. On the other hand, not every problem is suitable for being solved in a distributed computing environment. The problem structure has to be such as to allow its subdivision into packages that have little or no dependency between each other. This requirement is the main limitation to

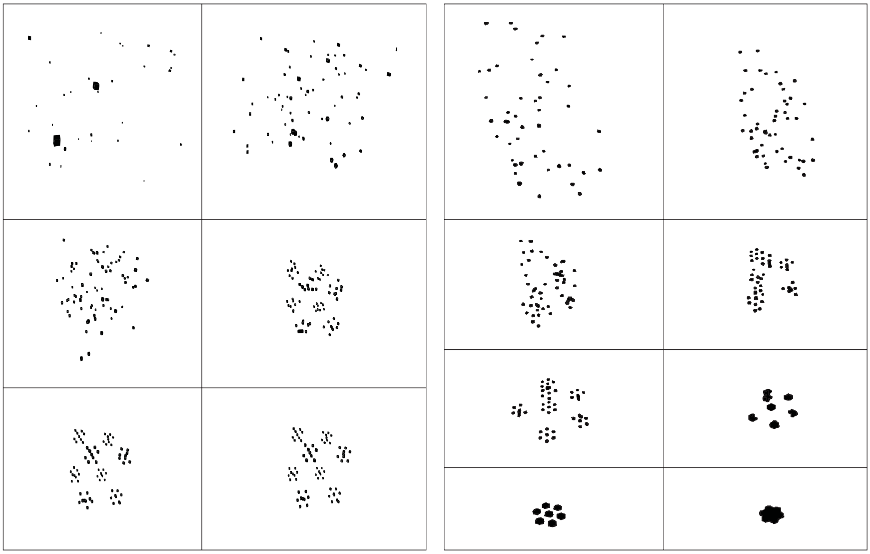

Fig. 12.1. Two examples of orbital swarms assembling a given structure (source: [IP07]).

the use of distributed computing. The forthcoming sections introduce, briefly, two examples of space applications suitable for distributed computations.

Analysis of Large Quantities of Data

The main purpose of most of the commercial satellites currently orbiting Earth is to provide data. Satellites continuously download data to ground stations in a nonprocessed format (usually, few data manipulations are made by the not too powerful computers onboard satellites). ESA's ENVISAT satellite alone generates 400 terabytes of data each year [FGL+03]. The data are then processed sequentially by computers and the results stored again in mass memories together with raw ones. Over the years, these data accumulate to the point that deletion is sometimes necessary (also due to changes in storage technology). Sophisticated analysis of these datasets can take as long as years to complete, often making the analysis itself obsolete before it has even been concluded. Distributed computing therefore becomes a useful tool to allow efficient use of satellite data, the main asset of the space business. Earth observation data coming from European satellites have already been made available in a computer grid [FGL+03], sharing processing power, memory storage, and processed data. A dedicated generic distributed computing environment that uses the idle CPU time of ESA internal personal computers has also been tested already [IM05] on problems such as ionospheric data processing and Monte Carlo simulations of constellation architectures

[IMN05]. In these types of applications, as no dependency is present between the different parts of the computations, little distributed artificial intelligence is actually present. The huge problem is just divided into small isolated subproblems that, in turn, are solved by different machines located in various parts of a common network. From the technological point of view, the challenges in these types of distributed computations (and the part where artificial intelligence could play a role) are mainly in the coordination of network traffic, in resource sharing, and in the reconstruction of the whole solution from the different parts returned by the different machines.

Distributed Solving of Global Optimization Problems

Distributing global optimization tasks over a large network of computers is certainly more elaborate, as it introduces a dependency between the different computations. Global optimization problems can be found everywhere in industrial processes. Many of the issues engineers face during spacecraft design are global optimization problems. Most notably, global optimization seems to be essential in preliminary trajectory optimization [MBNB04]. Essentially, this can be considered in the rather generic form

$$\begin{gathered}\min : \mathbf{f}(\mathbf{x}) \\ \text{subject to} : \mathbf{g}(\mathbf{x}) \leq 0\end{gathered}$$

with $\mathbf{x} \in \mathcal{U} \subset \mathrm{R}^n$. The problem dimension n depends on the type of trajectory considered and can be as low as 2 but also on the order of thousands.

In order to visualize, for the reader, the problem of trajectory optimization, Figure 12.2 illustrates an example of an optimized interplanetary trajectory. Since the first applications of evolutionary strategies to trajectory design [RC96], heuristic optimization techniques such as differential evolution, simulated annealing, particle swarm optimization, and genetic algorithms have proven to be quite effective in providing preliminary solutions to trajectory problems [BMN$^+$05, DRIV05]. The complete automatization of the optimization process, however, is not yet possible, as the existing algorithms are incapable of replacing the acute reasoning necessary to locate the best possible transfer between celestial bodies. A recent attempt to capture some expert knowledge and to use it to prune the search space in a trajectory problem, called "multiple gravity assist" (MGA), has managed to reduce the MGA problem complexity to polynomial time [IBM$^+$07]. In other cases, NP-complexity cannot be avoided, and the global optimization of an interplanetary trajectory may be untractable for a single machine. Fortunately, global optimization algorithms such as evolutionary algorithms and branch-and bound-based techniques are suitable for distributed environments [GP02, AF98], drastically improving the performance of the search and thus alleviating the "curse of dimensionality". A first attempt to use this possibility in a spacecraft trajectory optimization problem has been performed by ESA

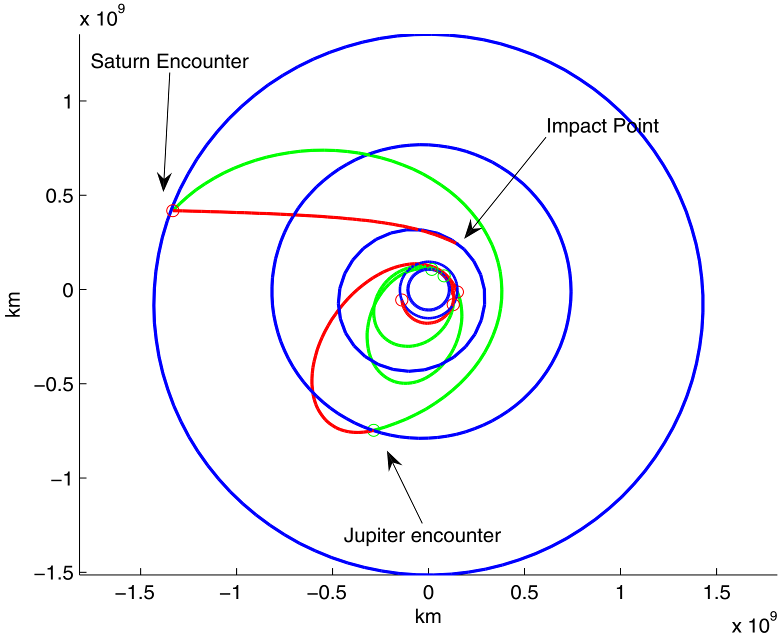

Fig. 12.2. An optimized Earth-Venus-Earth-Venus-Earth-Jupiter-Saturn-asteroid trajectory.

researchers, who solved a complicated MGA transfer distributing a differential evolution algorithm in a small number of personal computers. The problem solved[3] using the distributed version of differential evolution was inspired by the 1st Global Trajectory Optimization Competition (GTOC1), an annual event established in 2005 to make international research groups compete to find the best solution to the same trajectory design problem. Depending on the type of spacecraft one is considering (the main difference being the possibility of having impulsive or continuous velocity changes), the mission goal (destination orbit and celestial body), and the launch window considered, the trajectory optimization problem's dimension and complexity vary a lot. As in many other fields, for trajectory optimization, too, there is no available algorithm that outperforms all others. Consequently, this often leaves one to try different techniques until finally the algorithm that in a particular problem performs best is found. In the attempt to make the entire problem-solving process entirely distributed, a novel approach is that of Vinko et al. [VIP07], who consider a central server and a number of clients, which evolve demes (subpopulations) extracted from a larger population stored in the server,

[3] There is actually no mathematical guarantee that the solution found is the global optimum, but the experiment improved previous solutions by approximately 10%.

which then takes care of the reinsertion of the demes. According to the results returned by the various clients in each given phase of the optimization process, the server updates the probabilities to allocate a given algorithm for the next subsearch request to a client. The resulting global optimization environment is able to understand and select the best-performing algorithm in each phase of the solution of a problem. A preliminary version of this intelligent server is being developed and tested [VIP07] for the final purpose of being able to automatically carry out the whole trajectory optimization process without any expert supervision.

12.3 Enhanced Situation Self-Awareness

Ideally, a spacecraft should be able to perform autonomous actions, determine its own health status, and eventually make decisions based on this enhanced self-awareness. Unfortunately, real space missions are instead strongly dependent on the ground segment and on the flight engineers who monitor the enormous amount of telemetry data sent back to Earth during spacecraft operation. This procedure, which requires large numbers of human experts, is of course cumbersome and time-consuming. Sometimes, it might take days before the data are processed, decisions are made, and uploaded commands reach the spacecraft, whereas during critical mission phases such as the launch, information must be processed and decisions make within seconds. Furthermore, humans are not always able to recognize anomalous situations, especially when these involve complex relationships among large numbers of variables. Autonomous systems for enhanced situation self-awareness are therefore a very important research topic in spacecraft engineering.

Classically, two major approaches can be described: model-based methods and data-driven (model-free) methods. Model-based methods use models of the hardware and the physical processes to track the states of the system and detect deviations from nominal behavior. These models are sometimes very expensive to produce because they largely depend on expert knowledge. Moreover, when applied to very complex systems such as spacecraft, they might fail to reproduce all the possible off-nominal modes for which accurate models are lacking most of the time.

On the other hand, data-driven approaches, based on data mining and machine-learning techniques, are not based on a physical system but rather on models that are inferred from the telemetry data (e.g., temperature sensor data). Many activities in this field are being carried out in the framework of the Integrated Vehicle Health Management (IVHM) program of NASA Ames Research Center for the Second Generation Reusable Launch Vehicle (RLV), crew, and cargo transfer vehicles [IVH06]. This program is dedicated to the development of highly integrated systems that will include advanced smart sensors, diagnostic and prognostic software for sensors and components, model-based reasoning systems for subsystem- and system-level managers, advanced onboard and ground-based mission and maintenance planners, and a host of other software and hardware technologies. These hardware and software technologies will provide both real-time and life-cycle vehicle health information, which will enable decision making.

12.3.1 Data-Driven Approaches

The application of data-driven approaches to flight time-series analysis is being researched extensively by the space engineering community for the autonomous identification of suspicious trends that might lead to malfunctions or losses. Only the preventive detection of these trends might allow the ground systems or the intelligent planner and scheduler of the spacecraft to take corrective actions. Most of the data-driven approaches used in daily spacecraft operations are based on unsupervised learning techniques since in safety-critical applications, such as space engineering, it is usually impossible to collect exhaustive datasets for the representation of all possible fault modes. Therefore, most of these methods and algorithms can detect anomalies and off-nominal trends but leave to the flight control operator the delicate task of interpretation. The forthcoming paragraphs introduce a few of these approaches. Far from being an exhaustive list, we intend to give the reader a flavor of some work done in this field.

In [Ive04], the authors propose an "inductive monitoring system" (ISM) to detect off-nominal behaviors. Flight data of past missions are used as training data for a clustering algorithm (i.e., K-means and density-based clustering) that identifies nominal behavior areas (the clusters) in the n-dimensional data space, where n is the number of sensor readings. The clusters, which, according to the authors, represent the ISM knowledge base, can be used for the real-time detection of anomalous behavior during a new flight. Once a new measurement vector is received, the knowledge base returns the cluster to which the vector would belong (according to some cluster limit, preidentified after training). When the membership in a specific cluster cannot be detected, the distance to the closest cluster (with respect to Euclidean metrics in the n-dimensional space) will give the control operator an idea of the system's deviation from its nominal behavior as represented by the training data. The algorithm is tested successfully on the data collected during mission STS-107 of the Columbia space shuttle, which exploded during reentry because of a breach in its thermal protection system [Geh03]. An approach very similar to the one just introduced is presented in [Sch05]. In this work, an unsupervised detection algorithm named Orca, developed by the authors on the basis of the nearest-neighbor approach, is applied to the test data of the space shuttle main engine and of a rocket engine stand.

The K-means clustering algorithm is also used in [VLFD05] on the space of the features extracted from the time series collected from past missions. The authors here make an attempt to find specific relations between fault occurrence and the trend of the parameters by inferring association rules from data by means of the a priori algorithm. Therefore, the fault occurrence data must be part of the training set so that the algorithm can be trained to recognize future similar events. Unexpected fault modes therefore cannot be detected by the algorithm.

12.3.2 Model-Based Approaches

Most of the model-based approaches for enhanced situation awareness that have been researched and developed in space systems in recent years used as a reference the Livingstone model-based diagnosis engine [WN96] and its successors Livingstone 2 (L2) and Livingstone 3 (L3). Livingstone flew on Deep Space 1, and L2 has been uploaded to the Earth Observing One (EO-1) satellite [HSC$^+$04, CST$^+$04] for the “autonomous sciencecraft experiment” (ASE), which provides an onboard planning capability. The task of these diagnosis engines is to predict nominal state transitions initiated by control commands monitoring the spacecraft sensors and, in the case of failure, isolate the fault based on the discrepant observations. Fault detection and isolation is done by determining a set of component modes, including most likely failures, which satisfy the current observations.

L3 is the most recent and advanced of these architectures and consists of three main components. The “system model” stores the model of the system and is responsible for tracking the modes of operation of the different components and determining the constraints that are valid at any point in time. The “constraint system” serves the role of tracking the overall system behavior using constraint programming techniques. It receives constraints from the system model indicative of the current configuration of the system and propagates these constraints to try to assign consistent values to variables in the system. When observations are different from propagated values for corresponding components, the “candidate manager” is responsible for generating candidate faults that resolve all the conflicts and that can possibly explain all of the inconsistencies. In order to deal with uncertainties, the dynamic behavior of the system is tracked through Bayesian approaches such as “particle filtering” in order to assign posterior probability distributions to the candidate faults [NDB04, NBB04].

Bayesian approaches are also used in [GIB06], where the authors present the preliminary results of dual filtering techniques for the detection of possible variations of the thermal properties of the spacecraft that result from variations of its physical properties and for determining a complete thermal mapping of the system. System and sensor uncertainties are taken into account in the lumped parameter modeling of the thermal system, and a dual unscented Kalman filter is run on the stochastic model in an alternating optimization fashion to estimate the thermal state and coefficients of the resulting thermal network from the readings of a few strategically placed thermal sensors. Events such as faults can be detected by the dual filter as well as new values of system parameters (e.g., radiative couplings) that result from a variation of the spacecraft geometry (e.g., from the deployment of antennas, solar panels, etc.). This method would be particularly attractive in networks whose state and parameters can be estimated by the filter using a minimal amount of readings. The relation between the network topology and this minimal number is therefore an issue strictly related to the observability of the system, which is here approached using graph theory.

12.4 Decision Support for Spacecraft System Design

As the complexity of space systems increases, innovative approaches to system design are needed to allow assessment of the largest possible number of design concepts at an early stage. In space system design, several disciplines corresponding to all different subsystems[4] must be considered, and the overall spacecraft is the result of a "multidisciplinary design optimization" (MDO) [BS02, Roy96]. MDO can be described as a methodology for the design of systems where the interaction between several disciplines must be considered and where the designer is free to significantly affect the system performance in more than one discipline. In this sense, the space design process is an integrated optimization[5] that receives as inputs the mission requirements in the form of constraints and produces as output an optimal design.

In the classical approach to MDO, each specialist would prepare a subsystem design relatively independently from the others using stand-alone tools. Design iterations among the different discipline experts would take place in meetings at intervals of a few weeks. This well-established approach has the drawbacks of reducing the opportunity to find interdisciplinary solutions and to create system awareness in the specialists. A considerable step toward a multidisciplinary approach in the early phases of space system design has been achieved through an MDO based on concurrent engineering, where a sequential iterative approach to system design is replaced by a parallel and cooperative approach. Design facilities where these methodologies are implemented are, among others, the ESA Concurrent Design Facility [BMO99], the NASA Goddard Integrated Mission Design Center [KMSR03], and the Concept Design Center at The AeroSpace Corporation [ADL98].

In these concurrent MDO approaches, however, the subsystem experts are the core of the decision process of the design. Over the last couple of years, much research has been dedicated to the achievement of decision support systems or that of autonomous system design methods, which try to capture the reasoning of the experts toward an optimal and robust design.

[4] A spacecraft is constituted by the following subsystems: attitude determination and control, telemetry tracking and command, command and data handling, power, thermal structures and mechanisms, and guidance and navigation [WL99].

[5] The term optimization is not used here in the strict mathematical sense but rather to indicate any procedure that aims to find a solution that is either optimal or suboptimal.

Therefore, the spacecraft design started to be viewed as the solution of an optimization problem under constraints: given a set of decision variables D (e.g., the dimension of solar arrays) and a set of constraints C on D (e.g., their volume and mass), the constrained optimization algorithm looks for the values of D that minimize or maximize an objective function $F(X)$ subject to C.[6] However, finding the optimal design point was revealed to be a very difficult task, and traditional global optimization approaches most of the time fail to find the global optimum in the design space [FCM$^+$97]. To tackle this problem in spacecraft design, a quite common approach is based on the employment of heuristic solvers. The Jet Propulsion Laboratory implemented an optimization assistant (OASIS) that depending on the design problem selects and tunes either a genetic algorithm or a simulated annealing algorithm [FCM$^+$97]. The goal of OASIS was to facilitate rapid "what-if" analysis of spacecraft design by developing a spacecraft design optimization system that maximizes the automation of the optimization process and minimizes the amount of customization required by the user. More recently, evolutionary algorithms have been used to evolve the design of the antenna that flew on NASA's Space Technology 5 (ST5) mission [HGLL06] and for trajectory design as discussed in the previous section.

The problem of tackling the conflicting situations that might emerge during the system design activity when interests from different disciplines must be harmonized in the same project or when different goals must be reached within the same mission has been studied in [AFA$^+$04]. The neighborhood approach aims at finding by means of dedicated heuristics a set of "paretian" solutions at the system level. To efficiently reduce the total number of such solutions to a small subset that is to be considered "optimal" from the point of view of conflict reduction, "game theory" and "multicriteria decision analysis" are used.

Other approaches to autonomous space system design look not only at the achievement of an optimal design but also at its robustness with respect to uncertainties of the design variables and models involved in the design.[7] In this framework, the most common approach in space system design is essentially based on safety margins and expert knowledge. The safety margins, which are the most conservative way of handling uncertainties, identify the worst possible conditions that might be encountered during the operational phase in order for the resulting design to be adequate. Probabilistic approaches have been introduced in space system design as a consequence of the Challenger accident in 1986 [Fey86] and are essentially based on "probabilistic risk analysis" [PF93]. However, in general, the probability of infeasibility for a given design cannot be determined reasonably without knowing the joint

[6] In the case of spacecraft design, the objective function is most of the time the cost, which is ultimately proportionally linked to be the spacecraft's total mass.

[7] For an extensive qualitative and quantitative overview of these uncertainties, the reader may consult [Thu05].

distribution of the uncertain variables or having sufficient amounts of data samples from past observations. Sometimes, the probability model assumptions can be replaced by deterministic data, for which a rigorous worst-case analysis could be performed by using numerically reliable tools, such as verified interval calculations. In the most recent literature on system design under uncertainties, design variables are modeled by a range of values (intervals), by membership-degree functions of fuzzy sets [LF02], or by evidence theory [CCV07]. The European Space Agency's Advanced Concepts Team is assessing a promising new approach for an autonomous and robust design based on the concept of clouds [Neu04, DP05]. Clouds capture useful properties of the probabilistic and fuzzy uncertainties, enabling the user to utilize the collected empirical information (even if limited in amount) in a reliable and validated way. Being a hybrid between probabilistic and deterministic models, clouds can provide risk analysis using tools from optimization, in particular global optimization, and constraint satisfaction techniques. The numerical techniques for solving such problems have recently become much more reliable and powerful and allow one to compute bounds for the expected values of any multivariate functions of design processes and also for probabilities of qualitative statements involving design variables [NFD$^+$07].

12.5 Summary

The aim of this chapter is to give the reader an overview of some of the research carried out within the international space community on artificial intelligence. Having identified artificial intelligence as one of the enabling technologies for the achievement of the various short- and long-term goals of the international space agencies, we believe that a synergic effort of scientists from both fields is required to effectively tackle the numerous open issues and challenges in this area. In more recent years, we have observed a growing number of researchers getting interested in the benefits of using artificial intelligence methods for space applications. These applications go beyond the more classical automated planning and scheduling field and include different mission phases from conceiving the preliminary design to the mission operation phase.

References

[ADL98] J.A. Aguilar, A.B. Dawdy, and G.W. Law. The Aerospace Corporation's concept design center. In *8th Annual International Symposium of the International Council on Systems Engineering*, volume 2. INCOSE Technical Working Group Papers, 1998. Published in CD-ROM.

[AF98] I.P. Androulakis and C.A. Floudas. Distributed branch and bound algorithms in global optimization. In P.M. Pardalos, editor, *IMA Volumes* *in Mathematics and Its Applications*, volume 106, pages 1–36. Springer-Verlag, New York, 1998.

[AFA+04] V. Amata, G. Fasano, L. Arcaro, F. Della Croce, M.F. Norese, S. Palamara, R. Tadei, and F. Fragnelli. Multidisciplinary optimisation in mission analysis and design process. GSP programme ref. GSP 03/N16, contract number 17828/03/NL/MV, European Space Agency, Noordwijk, 2004.

[BMN+05] V.M. Becerra, D.R. Myatt, S.J. Nasuto, J.M. Bishop, and D. Izzo. An efficient pruning technique for the global optimisation of multiple gravity assist trajectories. In I. Garcia, L.G. Casado, E.M.T. Hendrix, and B. Toth, editors, *Proceedings of the International Workshop on Global Optimization*, pages 39–45, Almeria, Spain, April 2005.

[BMO99] M. Bandecchi, S. Melton, and F. Ongaro. Concurrent engineering applied to space mission assessment and design. *ESA Bulletin*, 99, September 1999. Available at `http://esapub.esrin.esa.it/bulletin/bullet99.htm`.

[Bro91] R.A. Brooks. Intelligence without reason. In J. Mylopoulos and R. Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 569–595, Sydney, Australia, August 1991. Morgan Kaufmann, San Mateo, CA.

[BS02] V. Belton and T.J. Stewart. *Multiple Criteria Decision Analysis: An Integrated Approach.* Kluwer Academic Publishers, Dordrecht, 2002.

[BT07] G. Bonnet and C. Tessier. On-board cooperation for satellite swarms. In D. Girimonte, D. Izzo, T. Vinko, S. Chien, and V. Kreinovich, editors, *Workshop on Artificial Intelligence for Space Applications at IJCAI'07*, pages 26–27, Hyderabad, India, January 2007. IJCAI.

[Bus04] G.W. Bush. A renewed spirit of discovery: the President's vision for U.S. space exploration. 2004.

[CCD+05] S. Chien, B. Cichy, A. Davies, D. Tran, G. Rabideau, R. Castano, R. Sherwood, D. Mandl, S. Frye, S. Shulman, J. Jones, and S. Grosvenor. An autonomous Earth-observing sensorweb. *IEEE Intelligent Systems*, 20(3):16–24, May–June 2005.

[CCV07] N. Croisard, M. Ceriotti, and M. Vasile. Uncertainty modelling in reliable preliminary space mission design. In D. Girimonte, D. Izzo, T. Vinko, S. Chien, and V. Kreinovich, editors, *Workshop on Artificial Intelligence for Space Applications at IJCAI'07*, pages 22–23, Hyderabad, India, January 2007. IJCAI.

[Cla45] A.C. Clarke. Extra-terrestrial relays. *Wireless World*, LI(10):305–308, October 1945.

[CST+04] S. Chien, R. Sherwood, D. Tran, B. Cichy, G. Rabideau, R. Castano, A. Davies, R. Lee, D. Mandl, S. Frye, B. Trout, J. Hengemihle, J. D'Agostino, S. Shulman, S. Ungar, T. Brakke, D. Boyer, J. Van Gaasbeck, R. Greeley, T. Doggett, V. Baker, J. Dohm, and F. Ip. The EO-1 autonomous science agent. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 420–427, New York, New York, 19–23 July 2004. IEEE Computer Society, Washington, DC.

[DP05] D. Dubois and H. Prade. Interval-valued fuzzy sets, possibility theory and imprecise probability. In E. Montseny and P. Sobrevilla, editors, *Proceedings of the International Conference in Fuzzy Logic and Technology*, pages 314–319, Barcelona, Spain, 8–10 September 2005. Universitat Politènica de Catalunya.

[DRIV05] P. Di Lizia, G. Radice, D. Izzo, and M. Vasile. On the solution of interplanetary trajectory design problems by global optimisation methods. In I. Garcia, L.G. Casado, E.M.T. Hendrix, and B. Toth, xeditors, *Proceedings of the International Workshop on Global Optimization*, pages 159–164, Almeria, Spain, April 2005. Available at `http://www.esa.int/gsp/ACT/publications/pub-mad.htm`.

[DVC05] S. Damiani, G. Verfaillie, and M.C. Charmeau. An Earth watching satellite constellation: how to manage a team of watching agents with limited communications. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 455–462, Utrecht, Netherlands, 25–29 July 2005. ACM Press, New York, NY.

[ETR+03] D. Escorial, I.F. Tourne, F.J. Reina, J. Gonzalo, and B. Garrido. Fuego: a dedicated constellation of small satellites to detect and monitor forest fires. *Acta Astronautica*, 52(9–12):765–775, 2003.

[FCM+97] A. Fukunaga, S. Chien, D. Mutz, R.L. Sherwood, and A.D. Stechert. Automating the process of optimization in spacecraft design. In *Proceedings of the IEEE Aerospace Conference*, volume 4, pages 411–427, Snowmass at Aspen, CO, 1–8 February 1997. IEEE, New York.

[Fey86] R. Feynman. Report of the presidential commission on the space shuttle Challenger accident. Volume 2 (appendix f), U.S. Government Printing Office, Washington, DC, 1986.

[FGL+03] L. Fusco, P. Goncalves, J. Linford, M. Fulcoli, A. Terracina, and G. D'Acunzo. *ESA Bulletin*, 114:86–90, 2003.

[Gaz05] V. Gazi. Swarm aggregations using artificial potentials and sliding mode control. *IEEE Transactions on Robotics*, 21(6):1208–1214, December 2005.

[Geh03] H.W. Gehman. The Columbia accident investigation board report. Technical report, U.S. Government Printing Office, Washington, DC, 2003.

[GIB06] D. Girimonte, D. Izzo, and L. Bergamin. Reasoning under an uncertain thermal state. In *Proceedings of the 57th International Astronautical Congress*, Valencia, Spain, 2–6 October 2006. International Astronautical Federation.

[GP02] C.; Gagne and M. Parizeau. A new versatile C++ framework for evolutionary computation. In W.B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M.A. Potter, A.C. Schultz, J.F. Miller, E. Burke, and N. Jonoska, editors, *Proceeding of Genetic and Evolutionary Computation Conference*, page 888, New York, NY, April 2002. Morgan Kaufmann, San Francisco, CA.

[HGLL06] G.S. Hornby, A. Globus, D.S. Linden, and J.D. Lohn. Automated antenna design with evolutionary algorithms. In *Proceedings of the AIAA Space Conference*, San Jose, California, 19–21 September 2006. Available at `http://ase.arc.nasa.gov/publications`.

[HSC+04] S.C. Hayden, A.J. Sweet, S.E. Christa, D. Tran, and S. Shulman. Advanced diagnostic system on Earth Observing One. In *Proceedings of the*

AIAA Space Conference, 2004. Available at `http://ase.arc.nasa.gov/publications`.

[IBM+07] D. Izzo, V.M. Becerra, D.R. Myatt, S.J. Nasuto, and J.M. Bishop. Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. *Journal of Global Optimisation*, 38(2):283–296, June 2007.

[IM05] D. Izzo and M.Cs. Markot. A distributed global optimisation environment for the European Space Agency internal network. In I. Garcia, L.G. Casado, E.M.T. Hendrix, and B. Toth, editors, *Proceedings of the International Workshop on Global Optimization*, pages 133–140, Almeria, Spain, April 2005. Available at `http://www.esa.int/gsp/ACT/publications/pub-inf.htm`.

[IMN05] D. Izzo, M.Cs. Markot, and I. Nann. A distributed global optimiser applied to the design of a constellation performing radio-occultation measurments. In Univelt Inc., editor, *Proceedings of the 2005 AAS/AIAA Space Flight Mechanics Conference*, volume 121, pages 739–748, Tampa, Florida, 2005. AAS Publications Office. Paper AAS 05–150.

[IP07] D. Izzo and L. Pettazzi. Autonomous and distributed motion planning for satellite swarm. *Journal of Guidance Control and Dynamic*, 30(2):449–459, 2007.

[Ive04] I.L. Iverson. Inductive system health monitoring. In H.R. Arabnia and Y. Mun, editors, *Proceedings of the International Conference on Artificial Intelligence IC-AI'04*, volume 2, pages 605–611, Las Vegas, Nevada, 21–24 June 2004. CSREA Press.

[IVH06] Integrated vehicle health management (IVHM), March 2006. Available at `http://www.nasa.gov/centers/ames/research/`.

[KE95] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, Perth, Australia, 1995. IEEE, New York.

[KEM+01] I. Kassabalidis, M.A. El-Sharkawi, R.J. Marks II, P. Arabshahi, and A. Gray. Swarm intelligence for routing in communication networks. In *Proceedings of the IEEE Global Telecommunications Conference GLOBECOM*, volume 6, pages 3613–3617, San Antonio, TX, 25–29 November 2001. Available at `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=966355`.

[Kha86] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[KMSR03] G. Karpati, J. Martin, M. Steiner, and K. Reinhardt. The Integrated Mission Design Center (IMDC) at NASA Goddard Space Flight Center. In *Proceedings of the IEEE Aerospace Conference*, volume 8, pages 3657–3667. IEEE, New York, 8–15 March 2003.

[LF02] M. Lavagna and A.E. Finzi. A multi-attribute decision-making approach toward space system design automation through a fuzzy logic-based analytic hierarchical process. In T. Hendtlass and M. Ali, editors, *Proceedings of the 15th International Conference on Industrial and Engineering. Applications of Artificial Intelligence and Expert Systems*, volume 2358 of *Lecture Notes In Computer Science*, pages 596–606. Springer-Verlag, London, 17–20 June 2002.

[LYB03] J. Lawton, B. Young, and R. Beard. A decentralized approach to elementary formation maneuvers. *IEEE Transactions on Robotics and Automation*, 17(6):933–941, 2003.

[MBNB04] D.R. Myatt, V.M. Becerra, S.J. Nasuto, and J.M. Bishop. Advanced global optimisation tools for mission analysis and design. Technical Report 03-4101, European Space Agency, Noordwijk, 2004. Available at `http://www.esa.int/gsp/ACT/ariadna/completed\_studies.htm`.

[McI95] C. McInnes. Autonomous rendezvous using artificial potential functions. *Journal of Guidance Control and Dynamics*, 18(2):237–241, 1995.

[McQ97] F. McQuade. Autonomous control for on-orbit assembly using artificial potential functions, 1997. PhD thesis, Faculty of Engineering, University of Glasgow, Scotland.

[MO03] P. Messina and F. Ongaro. Aurora: the European space exploration programme. *ESA Bulletin*, 115:34–39, 2003.

[NBB04] S. Narasimhan, L. Brownston, and D. Burrows. Explanation constraint programming for model-based diagnosis of engineered systems. In *Proceedings of the IEEE Aerospace Conference*, volume 5, pages 3495–3501, Big Sky, Montana, 6–13 March 2004. IEEE, New York.

[NDB04] S. Narasimhan, R. Dearden, and E. Benazera. Combining particle filters and consistency-based approaches for monitoring and diagnosis of stochastic hybrid systems. In *Proceedings of the 15th International Workshop on Principles of Diagnosis DX'04*, Carcassonne, France, 23–25 June 2004. Available at `http://www.laas.fr/DX04/`.

[Neu04] A. Neumaier. Clouds, fuzzy sets and probability intervals. *Reliable Computing*, 10:249–272, 2004.

[NFD+07] A. Neumaier, M. Fuchs, E. Dolejsi, T. Csendes, J. Dombi, B. Banhelyi, and Z. Gera. Application of clouds for modeling uncertainties in robust space system design. ACT Ariadna Research ACT-RPT-05-5201, European Space Agency, Noordwijk, 2007. Available at `http://www.esa.int/gsp/ACT/ariadna/completed\_studies.htm`.

[PF93] M. Pate-Cornell and P. Fischbeck. Probabilistic risk analysis and risk based priority scale for the tiles of the space shuttle. *Reliability Engineering and System Safety*, 40(3):221–238, 1993.

[PIT06] L. Pettazzi, D. Izzo, and S. Theil. Swarm navigation and reconfiguration using electrostatic forces. In *Proceedings of the 7th International Conference on Dynamics and Control of Systems and Structures in Space*, pages 257–268, Greenwich, UK, 16-20 July 2006. Cranfield University Press, London.

[RB04] W. Ren and R.W. Beard. A decentralized scheme for spacecraft formation flying via the virtual structure approach. *AIAA Journal of Guidance, Control and Dynamics*, 27(1):73–82, January 2004.

[RC96] G.A. Rauwolf and V.L. Coverstone-Carroll. Near optimal low-thrust orbit transfers generated by a genetic algorithm. *Journal of Spacecraft and Rockets*, 33(6):859–862, 1996.

[Roy96] B. Roy. *Multicriteria Methodology for Decision Aiding*. Kluwer Academic Publishers, Dordrecht, 1996.

[Sch05] M. Schwabacher. Machine learning for rocket propulsion health monitoring. *SAE Transactions*, 114(1):1192–1197, 2005.

[SD92] G. Schoner and M. Dose. A dynamics systems approach to task level systems integration used to plan and control autonomous vehicle motion. *Robotics and Autonomous Systems*, 10(4):253–267, 1992.

[TB99] G. Theraulaz and E. Bonebeau. A brief history of stigmergy. *Artificial Life*, 5(2):97–116, 1999.

[Thu05] D.P. Thunnissen. Propagating and mitigating uncertainty in the design of complex multidisciplinary systems, 2005. PhD thesis, California Institute of Technology, Pasadena.

[VIP07] T. Vinko, D. Izzo, and F. Pinna. Learning the best combination of solvers in a distributed global optimisation environment. Technical Report ACT-RPT-TV-5200-LBCSDGOE, ESA Advanced Concepts Team, Noordwijk, 2007.

[VLFD05] E. Vecchio, B. Lazzerini, S. Foley, and A. Donati. Spacecraft fault analysis using data mining techniques. In B. Battrick, editor, *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, München, Germany, 5–8 September 2005. Published in CDROM.

[WL99] J.R. Wertz and W.J. Larson. *Space Mission Analysis and Design.* Microcosm Press, Torrance, CA, 3rd edition, 1999. Space Technology Library.

[WN96] B.C. Williams and P. Nayak. A model-based approach to reactive self-configuring systems. In *Proceedings of the 13th National Conference on Artificial Intelligence*, volume 2, pages 971–978, Portland, OR, 4–8 August 1996. AAAI Press/The MIT Press.