TECHNOLOGY FOR AUTONOMOUS SPACE SYSTEMS

Ashley W. Stroupe
Sanjiv Singh, Reid Simmons, Trey Smith, Paul Tompkins,
Vandi Verma, Regina Vitti-Lyons, Michael Wagner

CMU-RI-TR-00-02

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

September 2002

Technology for Autonomous Space Systems
Table of Contents

Abstract

To meet the demands of operating in distance space environments, there is a need for more autonomous space systems. This document presents a survey of the state-of-the-art in space autonomy. This includes technologies relevant to space system autonomy makes recommendations for future research. These technologies include underlying technologies, component technologies, and space systems. Underlying technologies are general approaches to artificial intelligence. Component technologies are aspects of autonomous system implementation using underlying technologies. The space systems discussed in this survey include all past, current, and near-future missions with significant levels of autonomous behavior as well as a representative, though not exhaustive, discussion of many of the (more typical) directly controlled space systems used. The recommendations for future research in space autonomy consists of a list of five thrust areas requiring further development in order to progress toward higher levels of autonomy for space systems.

1 Introduction

Space environments place extreme demands on the engineering and control of systems operating within them. Not only must these systems function at a high level of precision, the setting imposes stringent requirements for robustness and long term deployment. Since manned operation of machines in space is a very expensive — sometimes impossible — proposition, autonomy is a natural component of space systems.

For the purposes of this report, "Space System" may be taken broadly to mean any machine operating in space, having sensors, actuators, its own control loop closure through a computer, and a communication link to a human supervisor. For purposes of completeness, this report includes discussion of general tools that are intended as parts of space systems. A survey of space systems shows that autonomy is found at various levels. At one end of the spectrum are systems that are directly controlled, a human operator must control each degree of freedom at a high rate, and on the other end are fully autonomous machines that require commands at only a very abstract level. Since problems of perception and control are frequently overwhelming, in between the two extremes are systems that are supervised closely or that allow sharing of control between humans and automatic systems. In this regime, human operators intermittently program and continually receive information from a computer that itself closes an autonomous control loop through artificial effectors to the controlled process or task environment [Sheridan 1992]. For a large number of space systems, this is the most appropriate level of autonomy. For historical reasons, however, the preponderance of systems in this survey were directly, albeit remotely, controlled.

This document has three goals: to provide a basic understanding of the technologies required and approaches used for autonomous space systems, to discuss the capabilities of current space systems, and to propose areas for future research key to the further development of autonomous space systems.

Toward the first goal, it is intended to survey autonomy-related technology applicable to space systems. Both underlying technologies (Chapter 2) and component technologies (Chapter 3) are discussed. Underlying technologies are fundamental approaches toward machine intelligence. These technologies, such as model-based systems, expert systems, neural networks serve as the basis for much of Artificial Intelligence and autonomous control. On the other hand, component technologies, such as architectures, real-time control, fault-tolerant systems, and scheduling are specific aspects of autonomous systems or operations.

For the second, this report provides both a retrospective look at the history of prominent systems that have been deployed in space as well as those currently on the drawing board for future missions (Chapter 4). The systems are separated into those applicable to deep space, planetary surfaces or orbits, Earth orbital operations, and human assistance. Systems in each of these areas are listed in order of decreasing autonomy: Fully autonomous, semi-autonomous, teleoperated, and directly controlled. Appendix B at the end of this document provides a quick reference to the approximately 120 systems (past, present, and future) discussed in this document, with dates and reference to other materials available via the Internet.

Lastly, for the third, since the trend is clearly towards an increase in autonomy, this report concludes with a discussion of five thrust areas critical to the development of future space systems: high reliability, autonomy, team coordination, robot worksystems, and robotic exploration and discovery (Chapter 5).

2 Underlying Technologies

In this section, several general fundamental approaches toward machine intelligence for achieving autonomy are discussed. These approaches, called *underlying technologies*, are not specific for any particular system or for space applications, but instead are basic classes of artificial intelligence. As such, the goal of an underlying technology is to provide a framework in which decisions can be made by the system to achieve goals and remain safe.

The underlying technologies discussed here are those that have been applied to robotics and space related projects and are not meant to be a complete representation of the artificial intelligence field. These technologies are physical model systems, structural model systems, empirical model systems, and expert systems. Each of these approaches is a means of modeling either the robot system or the environment in which the robotic system operates in order to interpret sensor data and act appropriately. A description, examples of systems, and strengths and weaknesses will be discussed for each of these approaches.

2.1 Physical Model Systems

Physical models are explicit mathematical formulae characterizing the physics of the system. Physical models can be produced theoretically in advance, by knowing the properties of the modeled system, or they can be learned through any number of artificial intelligence learning methods.

2.1.1 Physical Models

Generally, the transfer functions of systems (the relationship between system behavior and input) are most easily represented by a mathematical formula basic on the dynamics of moving parts and the characteristics of the computing and circuitry controlling them. Environments are typically modeled mathematically as well, such as the equations of motion or equations representing positions in the environment that are accessible to the robot and that are not accessible to the robot (configuration space). [Asada and Slotine 1986, Brogan 1991]. Physical model-based systems are the keystone for controlling most robotic systems, both terrestrial and space. These models are used to predict the results of actions for planning and to compare current state with what is expected to monitor progress. Systems may be completely model based or rely on models for some aspects of their control system.

2.1.2 Applications

<u>System Control</u>

One of the most fundamental uses of model-based systems is system control [Asada and Slotine 1986, Brogan 1991]. Since models can represent the relationship between system inputs and behavior, a model of this relationship can be used to determine appropriate inputs to achieve desired results. This relationship will typically involve both the robotic system and its environment. Additionally, the sensors will be used for feedback control to improve performance. One example, discussed in the Systems section of this document, is DARTS Shell. DARTS Shell uses a mathematical representation of a spacecraft and of zero-gravity dynamics in order to aid an operator (human or machine) in determining appropriate system inputs for desired behaviors.

<u>Planning</u>

By using the same input-behavior relationship that is used for controlling robotic systems, models can be used to plan a robotic system's actions to predict results and search for goal states. The models predict the results of actions. These predictions can then be used to determine the most appropriate action. Remote Agent on Deep Space 1 does this to determine what course corrections are required based on position and velocity.

2.2 Structural Model Systems

Structural models represent relationships among features of the environment or among states of the robot system. These models are not expressed in explicitly mathematical terms. Structural models also are used commonly in robotics. Structural models are typically designed in advance, but can be learned.

2.2.1 Maps

Maps are an example of structural models in which landmarks are connected by directions and distances. Maps are generally represented as networks with landmarks as nodes, and directions and distances as transitions. Maps can be used directly by systems as networks for planning; goal states are defined in relation to feature nodes and any of the search methods can be used to produce paths from an initial state to the goal state. Maps can represent locations in the environment or can represent the relationships between system configurations. As an example, using an environmental feature map, where the features are cities, a route planner whose task is to find a path from Pittsburgh to New York might determine: go east along Rt. 70 until Philadelphia, then northeast along Rt. 95 until New York. [Russell and Norvig 1995].

2.2.2 Templates

Templates are another type of structural model. Templates represent configurations of environmental features or of the robot that can be compared to sensory input to determine the internal or external state of the robot. As an example, edges in an environment may be stored from different vantage points; a vision system with edge detection can compare a captured image with the templates to determine where it is in the environment. Another example of templates would be letter shapes for text and character recognition. [Dickmanns et al 1990, Fennema et al 1990].

2.2.3 Rules and Heuristics

Rules and *heuristics* are models defined as a set of conditional relationships between sensory input, from either internal state sensors or external environment sensors, and actions. Examples might include: “if the engine warning light comes on, turn off the engine,” or “when you see the asteroid directly ahead, turn left.” Rule-based models may be applied with mathematical models, for example, to determine how *much* to turn left. Rules and heuristics also form the underlying model for expert systems and fuzzy logic. [Rolston 1988, Schneider et al 1996].

2.2.4 Applications

System Control

As with physical models, the primary application for structural models is control of systems. In much the same way, structural models can be used to interpret sensor data or to determine the appropriate control input for achieving a desired result. [Asada and Slotine 1986, Brogan 1991].

Feature Extraction

Feature extraction has many uses in robotics systems. Features in sensory input may be extracted for direct use, such as in scientific data analysis, or for matching to templates for robot localization. Features are typically structural in nature (edges in images, for example) and thus this application is ideal for structural modeling. Some examples of feature extraction for scientific data analysis include Autonomous Satellite Detection, Robotic Search for Antarctic Meteorites, and Terrestrial Planet Finder; each of which is discussed in detail in the Systems, Tools section. Examples of feature extraction for localization include the approaches to robot navigation [Dickmanns et al 1990, Fennema et al 1990], which proved that templates of edges in the environment are capable of localizing a surface robot, and Remote Agent (discussed in detail in the Systems, Tools section), which uses templates of star configurations to localize in space.

Planning

Planning can also be done using structural models to predict the results of actions, or to compare sensor data to expected readings. ASPEN and COSMO (discussed further in the Systems, Tools section of this document) tools use heuristical models to determine optimal command sequences to achieve specified goal states. These systems are in use on Cassini.

2.3 Empirical Model Systems

Empirical models are those created when the exact physics of a system cannot be known or is to complex to efficiently model. As with physical models, these can be created in advance but are more typically learned. These models can take the form of a mathematical relationship, and can be probabilistic.

2.3.1 Neural Networks

A *Neural Network* is a network of interconnected artificial neurons used to learn and model mathematical functions. Each neuron performs a simple task which, combined with the simple tasks of other neurons, leads to complex problem solving. The approach is based on a model of the human brain, a large number of complexly interconnected simple neurons working together to achieve very complex behavior. [Hagan et al 1996, Kosko 1992, Russell and Norvig 1995].

In theory, neural networks can be trained to model any mathematical expression or logical expression. In practice, the ability of neural networks to model complex expressions is limited. The requirement of large numbers of neurons to solve some problems in turn leads to long training times and reduced chance of converging to an acceptable solution. More complex problems can often be approached by training individual, smaller networks on simpler parts of the complex problem and combining the results, either by an additional neural network or by another means, such as voting or vector summation. [Hagan et al 1996, Kosko 1992, Russell and Norvig 1995].

History

In 1943, McCulloch and Pitts proposed the concept of basing a computer problem solving method on modeling animal brains. They proposed that a neuron network would, in theory, be capable of performing any mathematical or logical function. They first proposed the structure of a network, still used today, as a series of weighted sums passed into simple thresholding mathematical functions and combined to form a result. [McCulloch and Pitts 1943].

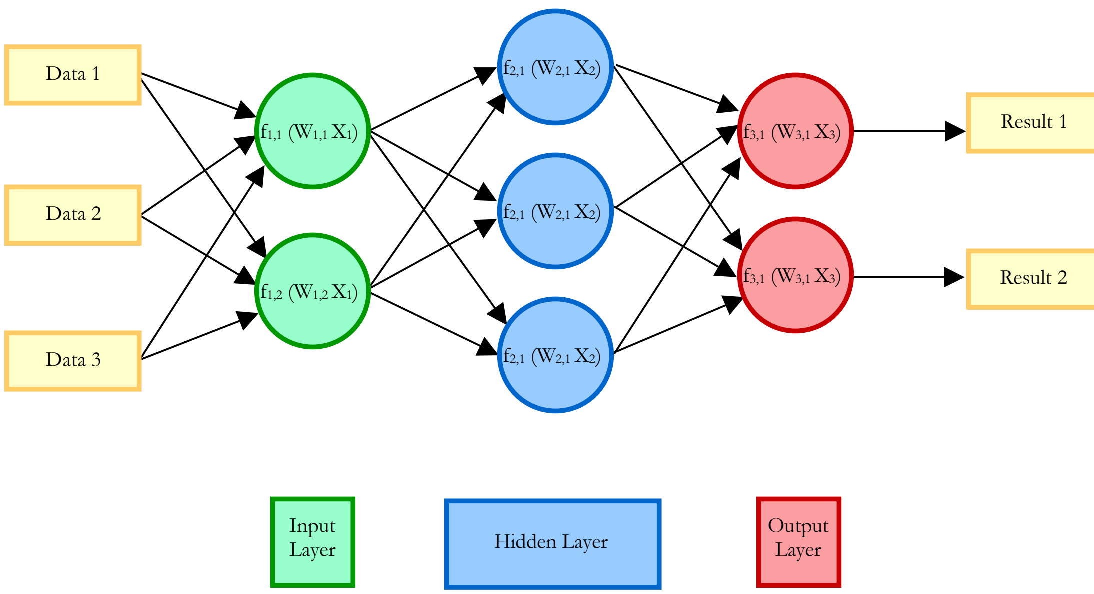
In 1948, Hebb first proposed that networks of neurons could be taught, rather than preprogrammed, to solve complex problems. [Hebb 1949].

The first milestone in the practical use of neural networks was the Perceptron, proposed by Rosenblat in 1958. The Perceptron feed-forward linear classifier was a simple neural network with a series of inputs, a layer of neurons, and an output neuron. It was developed for pattern recognition, and is capable of classifying data when class clusters are linearly separable. A Perceptron was built and demonstrated. [Rosenblat 1958].

Independently in 1972, Teuro Kohonen [Kohonen 1972] and James Anderson [Anderson 1972] each built neural networks to be used as memories. Later, J. J. Hopfield introduced the use of a recurrent network as an associative memory [Hopfield 1982].

Architecture

A neural network consists of any number of interconnected layers of neurons. The input to each successive layer is a weighted sum of outputs from the neurons in the previous layer. Each neuron performs a simple mathematical calculation on this weighted sum. The first layer, called the input layer, receives inputs from the environment. There are generally as many input neurons as inputs, though inputs may outnumber input layer neurons. The output layer consists of one neuron per output and calculates the final result of the neural network calculation. The result of the network is a series of one or more numerical values, often binary. The series can represent responses to individual signals or can be combined into a single output. Below is an example of an architecture for a small neural network with three inputs, a two-neuron input layer, a single three-neuron hidden layer, and two final result outputs. [Hagan et al 1996, Kosko 1992, Russell and Norvig 1995].



A typical small feedforward neural network with three layers. The neuron function is represented by f. X represents inputs to the neuron and W represents the weights applied to those inputs. On the functions, weights and inputs, the first subscript indicates the layer while the second indicates the number of the input to the layer.

The calculation performed by each neuron is defined by internal function. Common functions include simple thresholding, sigmoid functions for smooth thresholding, and linear functions with saturation levels. In trained networks, the form of these functions is predetermined, but the parameters are adjustable. Such parameters may include the slope of linear or sigmoid functions, the threshold value, or a bias term. [Hagan et al 1996, Kosko 1992, Russell and Norvig 1995].

Feed-forward networks, such as the Perceptron, pass the outputs of each layer directly to the inputs of the next. Feedback can often improve performance and training time. Networks with feedback are called *recurrent* networks. [Hagan et al 1996].

Training

Training a neural network consists of adjusting the internal neuron function parameters and the weights on neuron inputs in order to achieve the desired results. Generally, the weights and function parameters are randomly initialized. Training proceeds as labeled examples are passed through the network and the achieved results are compared with the desired results. The errors are calculated and used to determine what adjustments are required. In most cases the desired results are specified explicitly, but in other cases, these can be learned as well using cluster learning methods which varying classifications and determine which classifications make the most sense. [Hagan et al 1996, Kosko 1992, Russell and Norvig 1995].

Two methods of training have been commonly used. The first learning rule proposed was the Widrow-Hoff learning rule of 1960 [Widrow and Hoff 1960]. This method does not always converge and is often slow. In 1986, David Rummelhart and James McClelland introduced the Backpropagation gradient method [Rummelhart and McClelland 1986]. In Backpropagation, the weights are adjusted based on the amount of difference between achieved and desired results and on the rate at which changing the parameters will change the results. This rate term is the gradient in the weights. An additional parameter called the *learning rate* determines how fast jumps are made in the weights; increasing the learning rate improves convergence time to the optimum parameters but may increase the chance of jumping past the optimal values and failing to converge. [Hagan et al 1996, Kosko 1992, Russell and Norvig 1995].

Two final distinctions can be made in training methods. In a priori training all the training for a network is done in advance. A single set of examples is used to train the network before it is implemented. On-line (or adaptive) training occurs while the network is operating on actual data that is accompanied by the desired result. One or both of these methods may be used on any single neural network system. Adaptive training allows neural networks to respond to changes in the system or the environment, but may cause the system to forget how to appropriately respond to conditions that are not frequently revisited. Additionally, adaptive training may cause a system to learn an erroneous response if there is noise in the system. [Hagan et al 1996, Narendra and Parthasarathy 1990].

Applications

Neural networks have been proposed as solutions to all classes of artificial intelligence problems. In practice, they seem most applicable to a few specific problems.

System Control

Most systems include complex, nonlinear behavior that is difficult to predict and model. A neural network that is monitoring the input and outputs of such a system can be trained to model the system's behavior without predefining a complete and explicit mathematical function [Narendra and Parthasarathy 1990]. Models of such systems can be used for controlling the system or planning for the system. One example is Multimode Proximity Operations Device (MPOD) which used a neural network to learn and control neutral buoyancy docking maneuvers. MPOD is discussed in the Systems, Human Assistance section of this document.

Pattern Recognition and Classification

The most common application for neural networks is pattern recognition and classification. Since most classification problems can be expressed geometrically or mathematically, neural networks are often well suited to model them. In particular, problems with a finite number of well-defined classes can be solved with neural networks. Examples of classification problems for robotics and space systems include scientific data analysis on images, or instrument data and localization, where the pattern classification is done on sensory data so that it can be matched to templates. [Grossberg 1976, Hagan et al 1996, Russell and Norvig 1995].

Diagnostics

In practice, diagnostics is a specialized case of pattern recognition and classification. Patterns in system behavior can be indicative of whether a system is behaving normally or abnormally. The nature of the variation in the pattern from normal can further be specific to the type of malfunction. This is equally true of a robotic system or a biological one. [Kosko 1992].

2.3.2 Model-Based Learning

Most real-world systems defy modeling in terms of simple physical constructs. While methods such as neural nets provide a "black box" methodology, systems can be complex enough that such global representations are insufficient. One method that has been used with some success for complex non-linear systems is to keep a representative set of training data continuously without reducing it to a parametric model. That is, instead of modeling all the data simultaneously, predictions are based on local models fitted to the data in the neighborhood of a query point. Such methods require that all experiences are explicitly stored in memory and hence the term "memory based". A simple memory based method is nearest-neighbor. In this method, when a query is made, the memory is searched for the n nearest points in the input space. The dependent values of these points are averaged to provide a prediction. Instead of simply averaging the n nearest points, a linear model could be formed over this subset. That is, the surface of the function is approximated by local hyperplane patches. A further extension is to weight the points used in the linear model by their proximity to the query point. Intuitively, the further a point is in the input space from the query point, the less it should be weighted. This method is commonly called Locallly Weighted Regression (LWR).

History

Variants of memory based learning have been used by statisticians for many years and early versions of LWR have been described in [Cleveland88]. More recently these methods have been used and extended by researchers in robotics to model complex phenomena [Atkeson91] [Moore92] [Moore94].

Architecture

Use of LWR is predicated on the search for the n nearest neighbors to a query point. A naive implementation requires n distance computations, that is, O(nD) operations, where D is the number of independent variables. A faster implementation is to use K-D trees to partition the input space such that determination of nearest neighbors is improved [Friedman77]. The time to compute a prediction using LWR with K-D trees is given by:

$$\min(\alpha_1 n, \alpha_2 \log n) \bullet D^2$$

where α_1 and α_2 are constants that encode overhead costs. Computation time is asymptotically logarithmic in n although for large D, the overhead costs dictate that the prediction time will vary linearly until the size of the data set, n, is very large. One way to deal with large data sets is to train on a randomly chosen subset. Accuracy is surprisingly unaffected by reducing the number of data points provided they are chosen randomly from a larger, denser data set.

Training

Typically, the use of weighted regression as a learning method is accompanied by a "training" phase that searches for the kernel width and the best scaling of the input parameters that best minimize the sum of squares error. The standard method randomly selects training set from the data set and does a cross-validation with the rest of the data to verify the coefficients to be used for weighting. This phase is generally quick although it comes at a price of computation during prediction. A key advantage of using LWR is that since all the data is kept around, every query can have associated with it an estimate of certainty (measure of variance in the data close to the query point).

Applications

As mentioned earlier, a main application of memory based learning is in function approximation. For purposes of planning and robotics, this is akin to learning the *post condition* of a set of operators or a *forward* model of a dynamical system.. Memory based methods have been used in applications such as robot juggling [Moore 92], robotic earthmoving [Singh95] and for autonomous aircraft control [Bagnell 01].

2.3.3 Bayes Networks

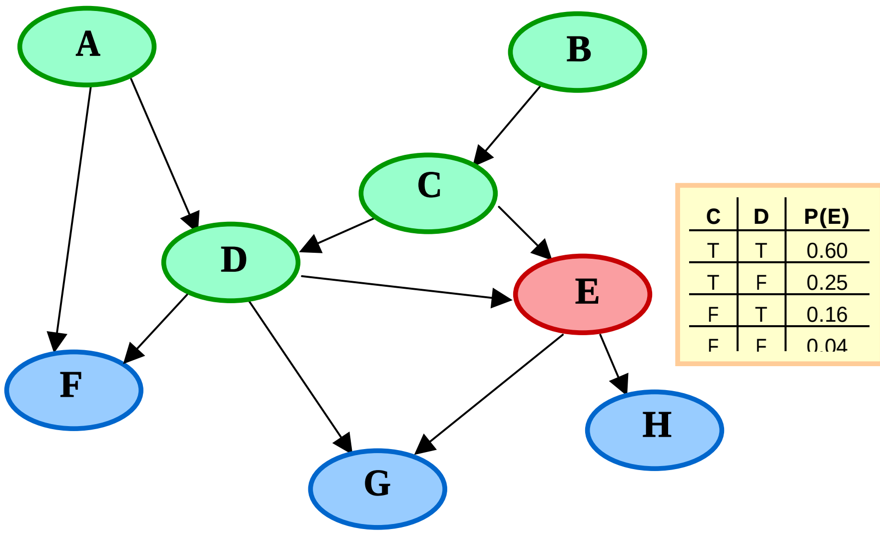
Bayes Networks, also called *Belief Networks* are a means of modeling a system probabilistically. This allows representation of the dependence between nodes in a non-deterministic way. The nodes of the network represent states of the system (or environment) and connections between them are assigned conditional probabilities. These networks can be used to compute the conditional probability of any joint state of the system. Additionally, inferences about potentially unobservable states in the system can be made based on the conditional probabilities and values of known states. The mathematics of the Bayes Network is based on Bayes' Rule, which relates conditional probabilities. Probabilities assigned can be computed initially or learned. [Russell and Norvig 1995]

History

Wright, investigating genetics, was the first to represent probabilistic information in a network in the 1920's. The introduction of Bayesian inference in a belief network occurred by Good in 1961. While these early networks used binary variables, continuous variables have been incorporated since the late 1980's, though only those that are distributed normally [Russell and Norvig 1995].

Architecture

The structure of a Bayes Network is demonstrated in this simple example. Each node represents a variable of the system, and is represented by a conditional probability table. Some variables may be directly observable, while others are not. Inferences about those that are not observable are made using Bayes' Rule on the conditional probabilities, following the structure of the network. All dependencies between variables are modeled as network connections.



A simple Bayes Network. Nodes A, B, C, D, and E are unobservable. Nodes F, G, and H are observable. The sample conditional probability table is shown for node E.

In the figure above, a sample Bayes Network is shown. The only observable variables are in nodes F, G, and H. In this example, the node of interest is E; its conditional probability table is shown. In order to make inferences about the state of node E, the observed states of G and H are required, because they directly connect to node E. Additionally, node F is connected to D and C, which both affect E through a different path than G and H; thus, the state of F must also be taken into account.

Applications

Bayes Networks can be applied to non-deterministic situations to determine the most likely outcome or in situations in which an action must be selected when the provided information in incomplete. One example is a medical diagnostic network, Pathfinder developed at the Stanford Medical Computer Science program in the 1980's [Russell and Norvig 1995]. For robots, the world is rarely entirely deterministic. Sensors and actuators have noise (false signals and measurement error) and resolution limits, and are therefore often modeled probabilistically. This makes Bayes Networks highly applicable to robotic applications.

2.3.4 Markov Models and Markov Decision Processes

The *Markov Model* is a network representation of states and state transitions within a system. Unlike finite state machines, Markov models have probabilities associated with each state-to-state transition. Markov models assume that the results of any transition depends only on the current state, and not on any historical context. States may include "values," which indicate how good or bad it is to be in that state. In a fully observable Markov model, the current state of the system is unambiguously known, either through direct knowledge or observation, as are all the transition probabilities. When the state is unknown, Hidden Markov Models (HMMs) can be used; observations are made after each transition in order to make inferences about current state. In each case, the transition probabilities can be assigned initially or learned. [Russell and Norvig 1995]

Markov models can be used to make decisions about how to act in the world when used in a *Markov Decision Process* (MDP). In an MDP, state-to-state transitions are motivated by actions. Each state has a set of allowable actions. The result of each action is probabilistic, and can therefore lead to different states. In systems where the state is partially observable, Partially Observable Markov Decision Processes (POMDPs) can be used. In this case, after each action is executed, an observation is made and the value of the state is determined. [Russell and Norvig 1995]

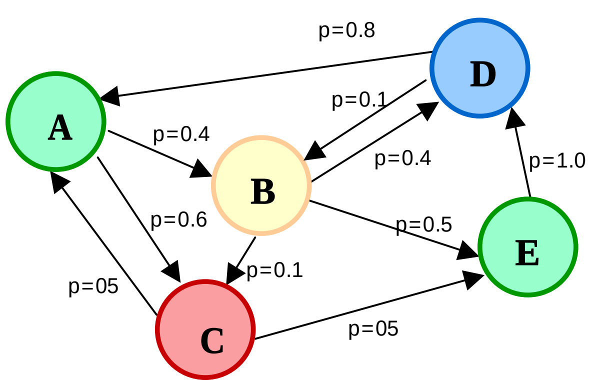
MDPs and POMDPs are frequently used in conjunction with learning, to determine which action is best to take under each circumstance. Generally, iterative methods explore various state-action pairs in order to determine the likely results using the values assigned to the states. Over time, this leads to a reasonable estimate of which actions are likely to produce the desired results, and a policy to maximize the value of the entire process. [Russell and Norvig 1995]

History

Markov first proposed the Markov Decision Problem in 1913, in the context of letter-sequence analysis; this formulation used the Hidden Markov Model. Work on inferring the underlying structure of a Markov Model based on data was begun in 1966 by Baum and Petrie, and later applied to speech by Baker (1975) and Jelinek (1976). Smallwood and Sondik published the first exact solution to POMDPs in 1973. Use of POMDPs in planning problems was done by Cassandra et al. in 1994. [Russell and Norvig 1995].

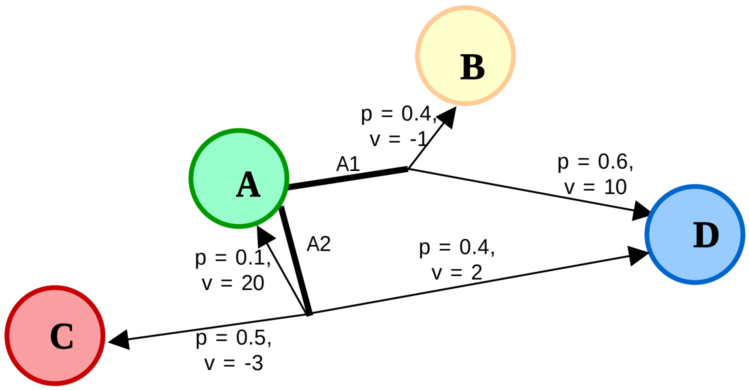
Architecture

Unlike Bayes Networks, in Markov Models, nodes represent states of the system and the connections between nodes represent actions. A Markov Model is shown in the figure below. The transition probabilities are represented on each transition.



An example of a Markov Model. Arrows represent transitions between states, and each is assigned a probability of occurring.

In contrast to the Markov Model, a Markov Decision Process is shown in the following figure. In the MDP, some control over transitions is allowed through action, though the results of the actions are probabilistic. In the figure, state A is the current state. Only those actions and values immediately relevant are shown for simplicity. There are two action choices: A1 may lead to state B with probability 0.4 and value –1 or to state D with probability 0.6 and value 10; A2 may lead to state D with probability 0.4 and value 2, state C with probability 0.5 and value –3 or back to state A with probability 0.1 and value 20.



A sample Markov Decision Process. State A is the current state. For clarity, only the actions available from state A are shown. There are two available actions and four possible outcomes.

Applications

Markov Models have been used frequently for modeling speech, beginning with Markov himself in 1913. HARPY [Lowerre and Reddy 1980] was a system produced for DARPA to represent meaningful word sequences.

Due to their ability to handle probabilistic state transitions, Markov models have been used to model robot control systems for control and for planning. Unlike Bayes Networks, which aid in making inferences about the state of a system (which can be used as inputs to a control system), MDPs can be used directly as control systems. Examples of such types of applications are presented in Russell and Norvig [Russell and Norvig 1995].

2.4 Expert Systems

An *Expert System* is a system designed to mimic an expert's ability to make decisions for real-world applications in specialized domains. In general, expert systems use rules/heuristics and inference to determine the best course of action. The approach is based on how human experts approach problem solving.

The expert providing knowledge and rules to the system and the ability to describe the knowledge in an appropriate manner limit the capabilities and success of expert systems. [Rolston 1988, Russell and Norvig 1995, Schneider et al 1996].

2.4.1 History

The concept of an expert system was first proposed in the 1950's, when programming languages were being developed to do symbolic reasoning. Several early systems were developed in the 1960's. Stanford developed DENDRAL in 1965 [Feigenbaum et al 1971]. Its purpose was to interpret raw mass spectrometer data in order to infer molecular structure. The Massachusetts Institute of Technology developed MACSYMA in 1965 as a system to provide complex mathematical analysis [Martin and Oxman 1988]. Carnegie Mellon University produced Hearsay in 1965 as a tool for natural language interpretation. [Erman and Lesser 1983].

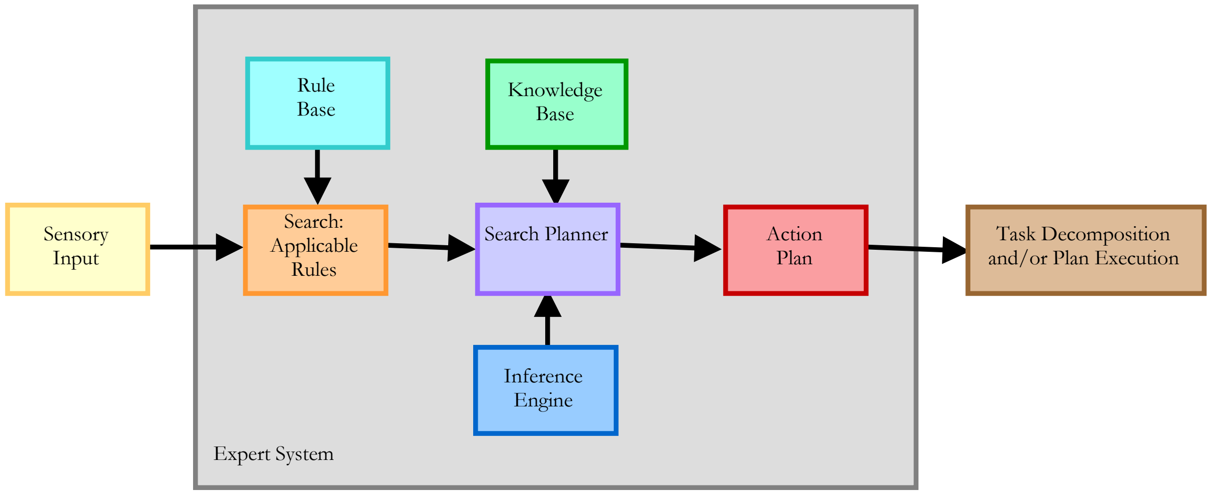
2.4.2 Principles

Process

The basis of an expert system is its knowledge base. The knowledge base is used for data interpretation and is highly specific to the problem domain. In addition to the knowledge base, a set of rules and heuristics is used to lead to procedures for problem solving within the domain. Search methods are applied to the rule base in order to determine which rules are applicable to the current situation. Once the appropriate rules have been identified, symbolic processing uses inference and the knowledge base to produce a procedure for solving the problem or obtaining the goal. [Rolston 1988, Schneider et al 1996].

The solution to a problem is generally presented as a model: a list of rules to be followed, a network of states and goals with appropriate transitions specified, or by an explicit mathematical model. Expert systems are expected to be able to explain the reasoning behind and the method for achieving the solution. [Rolston 1988, Schneider et al 1996].

The basic architecture is shown below:



A schematic illustration of the expert system architecture, receiving input from the sensors and reporting its derived plan to t he plan execution system.

Rules and Knowledge Bases

Two types of rules are generally recognized: procedural and heuristical. Procedural rules are generally specific rules that are invariant [Rolston 1988], such as the example given previously of "if the engine warning light goes on, turn the engine off." Heuristical rules, in this context, are rules based on human expert experience and hunches. These are often approximate and can include fuzzy logic [Schneider et al 1996].

The rules and knowledge base are generated either by human experts or through learning. Human experts provide the basic information about how they solve problems, which is translated into the rule and knowledge bases. Learning can occur by example or by analogy. In learning by example, classified examples of procedures or data are shown to the system or experienced by the system and then incorporated appropriately. In learning by analogy, the system uses data or a rule similar, but not identical, to the circumstance and adapts it, adding the new rule or data to its program. [Schneider et al 1996].

2.4.3 Fuzzy Logic

Since human experts often use degrees and relative values to make decisions, fuzzy logic is often necessary to fully describe expert system rules in order to approximate them more closely. In general, fuzzy logic is applicable whenever degrees of truth are required or useful. One system that will be using fuzzy logic in space is Earth Observing-1 (see also the Systems, Orbital section of this document). EO-1 will be using fuzzy logic to determine and maintain appropriate satellite constellation formations.

Principles

Fuzzy can be interpreted as "vague" or "ambiguous." In many instances, the applicability of a class to an object is neither true nor untrue, but true to a degree. The degree of truth, or relative degree compared to that of another object, is often important. Fuzziness allows objects to belong to classes to differing degrees and determination of relative relationships among different objects. [Kosko 1992, Schneider et al 1996].

Fuzzy classes, or fuzzy sets, are commonly used by humans. The set of "people who are 50 years old" generally includes those who are almost 50 as well as those who not quite 51, not just those people who are exactly 50. The set of "short people" would include people who are 1.5 meters tall and those who are 1.6 meters tall, but those are 1.5 meters tall are *shorter* than those at 1.6. Fuzziness is an attempt to duplicate these types of classification.

Fuzziness is unlike probability. Probability is useful in non-deterministic systems where you cannot definitively predict occurrences. In such systems it is how likely an event is to occur that is important. Fuzziness is a means of interpreting events or objects once after they have occurred. Probability can tell you how likely it is that a tall person walks into a room, but only fuzzy theory can tell you *how* tall that person is after he or she arrives. [Kosko 1992, Schneider et al 1998].

Fuzzy Logic is mathematical logic that allows vagueness in symbols, symbol matching to determine rule applicability, and quantifiers. A traditional logic rule might read:

If the distance to the target is greater than 10 meters, move forward at 10 meters per second.

For a fuzzy logic system, this rule might be rewritten as:

If the distance to the target is very large, move forward quickly.

The system must know to what degree different distances are "large" and to what degree different speeds are moving "quickly." Fuzzy symbol matching would allow distances close to 10 to find this rule applicable. The degree to which the rule is applicable is determined by the degree to which the distance matches "large." The result, the speed matched to "quickly" may also be determined by the degree of applicability of the rule. The use of "very" in this rule is an example of a vague quantifier, quantifiers such as this will alter the level of match between objects (such as distance in this example) to the symbol ("large"). [Kosko 1992, Schneider et al 1998].

A system based on fuzzy logic begins with the definitions of degrees of truth. The degrees of truth for various conditions, or the degree of belonging to each fuzzy set, are defined for a series of examples. Interpolation between these defined examples determines the degree to which new data belongs to the fuzzy set, which is in turn used for mathematical logic using the data. [Kosko 1992, Schneider et al 1996].

While fuzzy logic allows for vague matching of symbols and rules, the rules of basic logic and math sets apply. Combining rules and applying them to data allows for making implications and inferences. New sets can be generated from the old ones by set unions or set intersections. The logical conjunctions AND and OR also apply; for an AND you look at the minimum degree of truth (since it will be the most constraining); and for an OR you look at the maximum degree of truth (since only one must match the criteria). [Kosko 1992, Schneider et al 1996].

History

Fuzzy logic was inspired by the Heisenberg uncertainty principle, which brought indeterminacy to physics. Its formulation is also based on how humans think in degrees and vague concepts without assigning specific values to everything. [Kosko 1992].

The first step toward fuzzy logic was multi-valued logic proposed by Rosser and Turquette in 1952 and independently by Rescher in 1969. In multi-valued logic, the logical world is no longer binary (0 and 1, or TRUE and FALSE). Instead, degrees are assigned that fall between these values. [Kosko 1992].

Lotfi Zadeh first proposed the concept of a fuzzy set in 1965. An element is not only included or excluded from a fuzzy set, but its degree of belonging to the set is specified. "Tall people" is such a fuzzy set. [Kosko 1992].

2.4.4 Applications

Several types of problems have been solved with the expert system approach including data analysis, medical diagnostics, fault diagnostics, and design.

Control and Planning

Expert systems such as that on Deep Space 1 can be used to plan and execute sequences of actions in order to achieve mission goals (see Deep Space 1 in the Systems chapter).

Data Analysis

Stanford's DENDRAL used an expert system to interpret raw data from a mass spectrometer and infer molecular structure in 1965 [Feigenbaum et al 1971]. Prospector, a product of the Stanford Research Institute in 1972, identified minerals in samples for predicting the presence of molybdenum deposits. [Duda et al 1978].

Medical diagnostics

MYCIN, created by Stanford in 1972, used the results of blood tests to diagnose blood diseases and recommend appropriate treatments [Shortliffe 1976]. The University of Pittsburgh produced Caduceus in 1975 to perform internal medicine diagnostics [Rolston 1988]. Puff was capable of pulmonary disease diagnostics in the 1980's; the Casnet and Internist expert systems to assist in medical diagnostics were also developed in the 1980's [Matin and Oxman 1988].

Fault diagnostics

SOPHIE was a system for troubleshooting electronic circuits used in the 1970's that was built using training rather than complete pre-programming [Matin and Oxman 1988]. Jet X diagnosed military aircraft engine faults starting in 1988 [Shah 1988]. AITest debugged computer programs, also in 1988 [Ben-Bassat 1988]. PERF-EXS was a system designed to monitor manufacturing plant functions to detect malfunctions and determine their causes. [Matin and Oxman 1988].

Other

ASDEP was an expert system developed for power plant design [Matin and Oxman 1988]. VIPS was a general planner to achieve goals using Petri Net representation [Matin and Oxman 1988]. The EXPLAIN expert system aided non-experts in using image processing algorithm packages [Tanaka and Sueda 1988]. Carnegie Mellon's R1 configured DEC computer systems [McDermott 1982]. Soar (Newell 1990) is a general-purpose expert system architecture that has been used in many recent applications.

2.4.5 Tools

Several types of support tools have been developed for the generation of expert systems. Some tools assist in the building of expert systems while others aid in the automatic translation of data into a knowledge base. Examples of tools used for assisting in the development of expert systems include Stanford's AGE [Nii and Aiello 1979], Rand Corporation's Rosie (1978) and Carnegie Mellon's OPS5 (1974) [Rolston 1988]. Teiresias (Stanford, 1972) was developed to transform expert knowledge into a computer knowledge base [Rolston 1988].

3 Component Technologies

In this section, various component technologies applicable to space robotics are discussed. In this context, a component technology is defined as specific aspects of robot design or operations. Each aspect of robot operations, such as control, can be approached in several different ways, as can the aspects of design, such as architecture and interfaces. These approaches (as applicable to space robotics) are discussed here. These technologies are often based on the underlying technologies discussed in the previous section.

The aspects of robot operations discussed in this section include real-time control, fault detection and diagnosis, fault tolerance, task planning and scheduling, and navigation. Aspects of design discussed here include robot architectures, human-machine interfaces, and visualization methods.

3.1 Architectures

Autonomous systems are typically quite complex. Well-designed software architectures can provide concepts, constraints and tools that make it easier to design, implement, and debug such systems. While no single architecture is best for all applications, this section presents some of the commonly used architectures and describes their advantages and disadvantages.

The term architecture actually encompasses several different notions. Of particular interest are architectural structure and style. Architectural structure refers to how a system is divided into subsystems, the behaviors of those subsystems, and the interfaces between subsystems. The traditional "boxes and arrows" diagrams are often used to represent architecture. Architectural style refers to computational concepts that underlie a given system. For instance, one system might use a publish-subscribe message passing style of communication, while another system may use a more synchronous client-server style.

Architectures can provide support in the design, implementation, and debugging of autonomous software. From a software design perspective, a well-defined architectural structure enables systems to be developed in a modular, distributed fashion. Architectural style also greatly influences how components behave and interact with one another. For instance, a publish-subscribe style can make it easier for components to react to changes in other parts of the system, and can lessen the need to replicate certain aspects of the internal state. From an implementation perspective, there exist standardized languages, code libraries, and tools to support various architectural styles. This can make it easier to implement systems using those styles, and can provide increased confidence in the systems, since they use well-tested infrastructure. Examples include architectures based on finite-state automata, such as ControlShell [Schneider et al 1998] or Orrcad [Coste-Maniere and Turro 1997], that come with tools and a large amount of software support.

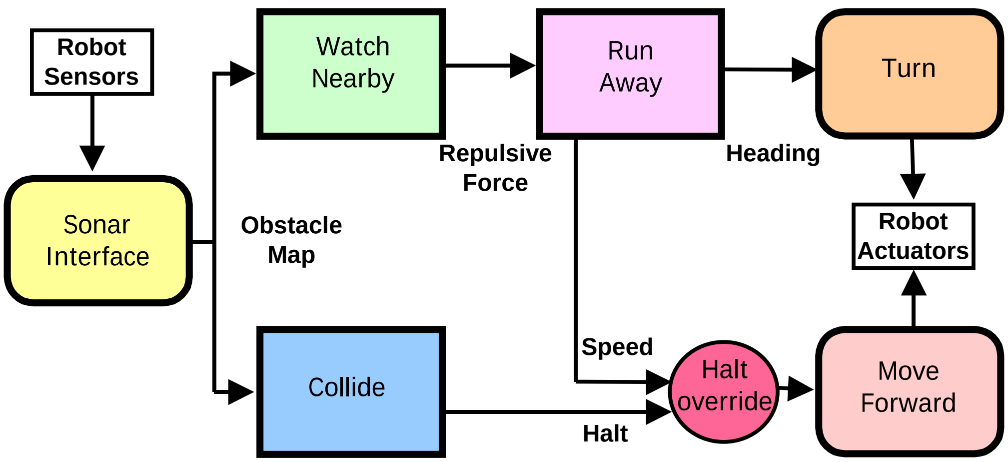
Finally, the architecture can make it easier to test and debug systems. A well-designed architectural structure helps in enabling components to be tested in isolation and replacing components by other implementations (such as simulators) that adhere to the same specifications. Architectural style can help as well. For instance, in an asynchronous publish-subscribe style, less internal state is needed, which means that there are typically fewer things that can be out of sync. On the other hand, a synchronous approach to communication can be more predictable. Which style is chosen depends on the specifics of the domain. Many architectures come with visualization tools that help in understanding how a system performs. Finally, there has been some work in using formal verification techniques to prove properties of different architectures. These techniques often depend on specific constraints and features of the architectures that make verification tractable [Lowry et al 1997, Musliner et al 1993, Pecheur and Simmons 2000].

3.1.1 Architectural Styles

Autonomous systems have two main objectives -- to achieve high-level tasks (goals), and to react in real time to changes in the environment. Most of the architectures for autonomous systems can be classified into three categories of style: behavioral, hierarchical, and hybrid. The categories differ largely in how they handle task achievement and in their reactivity.

Behavioral Architecture

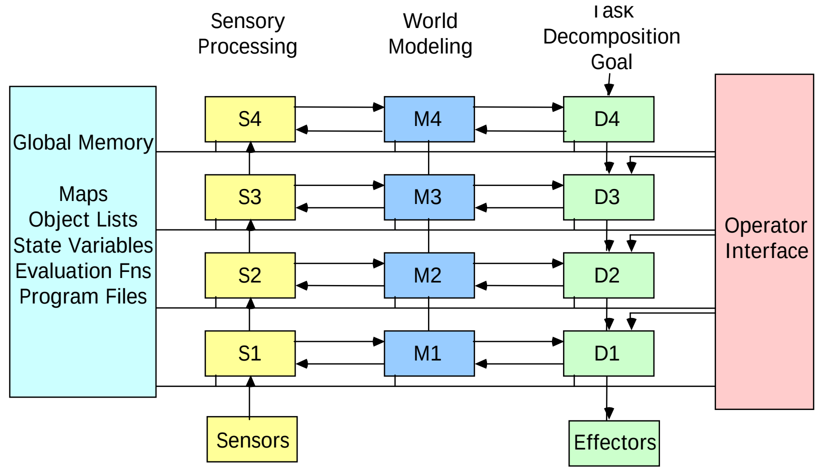
Behavioral architectures adopt a bottom-up approach. This style uses groups of software modules known as *behaviors* that run concurrently and interact through communication and through the environment [Brooks 1986, Gat 1992, Schneider et al 1998]). A key issue with behavioral architectures is how to arbitrate amongst concurrent, potentially conflicting, behaviors. Some architectures handle this by blocking the output of a subset of the behaviors, either through overrides, as in Subsumption [Brooks 1986], or by dynamically selecting which behaviors will run, as in 3T [Bonasso et al 1997]. Other architectures address this problem by combining the outputs of multiple behaviors to form a single output command, as in Aura [Arkin 1989] and DAMN [Rosenblatt and Thorpe 1995]. The tradeoffs are that blocking outputs typically leads to more predictable and computationally efficient systems, but combining outputs tends to be more flexible, since all behaviors have a potential effect on the chosen output. In general, behavioral architectures facilitate reactivity, but it is often difficult to use them to achieve non-trivial goals.



Example of the Subsumption architecture, in which the outputs of some behaviors can suppress or affect the outputs of other behaviors. Rounded blocks indicate modules that interact directly with the environment.

Hierarchical Architecture

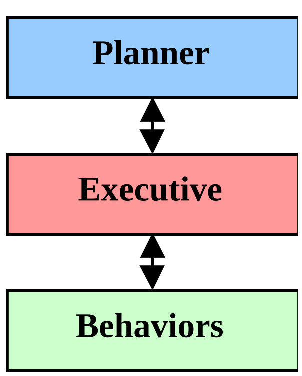
The hierarchical style adopts a top-down approach. It highlights the supremacy of high-level control and restricts low-level horizontal communication [Albus 1991, Albus et al 1988]. The architectures work by decomposing high-level, abstract goals into successively more detailed subgoals. Typically, each level in the hierarchy contains some degree of planning, world modeling, task management, and monitoring (shown in the figure below). In essence, each layer in the hierarchy can be thought of as a high-level control loop, with the layer above supplying the reference signal and the layer below acting as a "virtual machine." An issue in hierarchical architectures is how to decompose tasks -- spatially, functionally, or temporally. In general, hierarchical architectures are able to handle complex tasks with many interactions, but have trouble handling many sensors in tight reactive and reflexive loops.



The RCS hierarchical architecture.

Hybrid Architecture

Hybrid architectures are the most recent [Bonasso et al 1997, CIRCA, RA, Bellingham and Consi 1990, Borrelly et al 1998]. The hybrid style combines the best of both reactive and deliberative control in a heterogeneous architecture. Such architectures facilitate the design of efficient low-level, reactive control with a connection to high-level planning and reasoning. An issue with hybrid architectures is how to connect the deliberative and reactive components. Many hybrid architectures favor a three-level approach in which an executive component mediates between the low-level behaviors and high-level planning. An example is shown in the following figure. The three layers use very different types of representations and algorithms. In particular, the executive component typically contains constructs for task decomposition, task scheduling and synchronization, execution monitoring, exception handling, and resource management [Coste-Maniere and Turro 1997, Firby 1987, Gat 1996, Georgeff 1987, Simmons 1994, Simmons and Apfelbaum 1998]. It accepts goals from the planner, decomposes them into executable primitives, dispatches these to the behavioral part of the system, and monitors the execution. Hybrid systems have been used fairly extensively in the last few years to control increasingly complex systems [Bonasso et al 1997, Muscettola et al 1997]. An issue with hybrid architectures is that the different layers can be tricky to integrate, and must be carefully designed and implemented to provide the right mix of reactivity and deliberation.



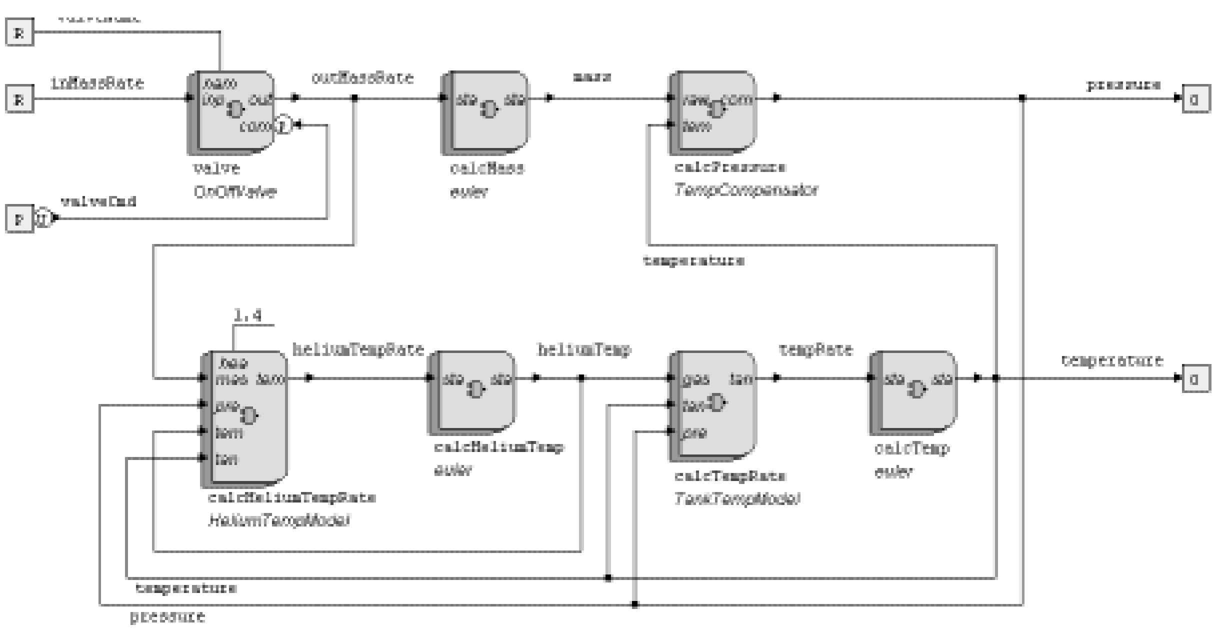
Example of a three-tiered hybrid architecture.

3.1.2 Examples of Architectures and Architectural Components

Much research has focused on the *executive* layer of architectures. Most have similar functionality, and they are mainly distinguished by whether they are self-contained languages [Firby 1987, Georgeff 1987], function libraries [Simmons 1994], or extensions of existing languages [Gat 1996, Simmons and Apfelbaum 1998]. The latter are particularly useful, since they facilitate easy integration with existing software. In particular, features of the language can be used only when needed.

<u>ControlShell</u>

ControlShell is a commercial product (from Real-Time Innovations) that provides a graphical programming environment for real-time and task-level control. For real-time control, ControlShell provides methods and run-time support for creating data-flow components that represent transfer functions and real-time control loops. Inputs and outputs of the data-flow components can be connected graphically, and then code is automatically generated that will run those components at designer-specified frequencies. For task-level control, ControlShell provides a graphical programming interface to develop hierarchical Finite State Automata. As with the data-flow diagrams, code is automatically generated for the FSA and run in real-time. The FSA and data-flow components communicate via NDDS, a flexible inter-process communication package. In addition, ControlShell, through its Stethoscope tool, enables users to view internal variables of an executing real-time system. [Schneider et al 1998].



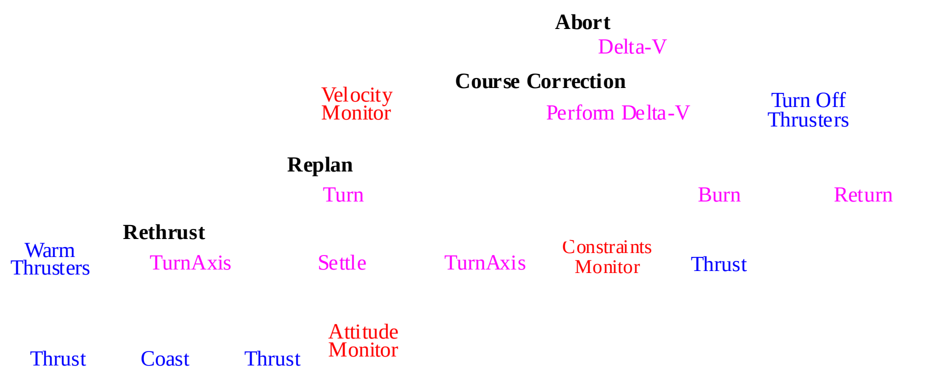
A graphical representation of the data flow through the ControlShell system.

Execution Support Language (ESL)

ESL (Execution Support Language) is an extension of Lisp, developed at the NASA Jet Propulsion Laboratory, that was designed specifically to help develop "smart executives" [Gat 1996]. Its primary use was as part of the DS1 Remote Agent [Muscettola et al 1997]. ESL is implemented as a macro library that makes use of many of the features of multi-threaded Lisp. In particular, it has syntactic constructs to achieve goals, synchronize tasks, handle exceptions, and manage discrete resources using the "property lock" mechanism. ESL also includes a simple back-chaining logical database (a sort of "mini Prolog") that enables it to perform real-time inference. In addition, the Remote Agent Executive that is built using ESL includes capabilities for executing plans produced by the planner, and doing resource and configuration management.

Task Control Architecture (TCA)

The Task Control Architecture (TCA) provides similar executive-level capabilities [Simmons 1994]. TCA is built around *task trees*. A task tree is a hierarchical representation of the achievement of a high-level goal by decomposing it into progressively finer subgoals. In addition, task trees encode temporal constraints between tasks and hierarchical exception handlers. The figure below represents such a task tree. TCA is a library-based system: The executive functions are implemented as C code plus calls to the TCA library, which uses message passing to send data to a centralized task management process. This process is responsible for dispatching tasks in accordance to the temporal constraints imposed by the system. TCA builds the task trees dynamically, and exception handlers can deal with exceptional situations by terminating sub-trees, re-invoking tasks with new parameters, adding new nodes, or adding new temporal constraints. TCA has been used in about a dozen autonomous systems, including a prototype autonomous spacecraft.



A Task Tree: decomposition of a task into subgoals. Temporal constraints are shown as horizontal arrows and hierarchical exception handlers are shown in bold black.

Task Description Language (TDL)

The Task Description Language (TDL) is based on TCA. TDL is an extension of C++, and includes explicit syntax for task decomposition, task synchronization, execution monitoring and exception handling (an example is shown in the figure to the right). It supports a richer set of temporal constraints than TCA, including both metric and relational constraints. TDL is implemented using a compiler that transforms TDL code into pure C++ code plus calls to a Task Control Management (TCM) library, written in C++. The philosophy behind TDL is that simple task-level control concepts should be simple to express in the language, but that it should not preclude one from doing complex things. TDL is being used in several autonomous robot projects, and is being extended to coordinate multiple, heterogeneous autonomous systems. [Simmons and Apfelbaum 1998].

```
Goal deliverMail (int room)
  Exception Handler noDelivery
{
  double x, y;
  getRoomCoordinates(room, &x, &y);
  spawn navigateToLocn(x, y);
  spawn centerOnDoor(x, y)
    with sequential execution previous,
    terminate in 0:0:30.0;
  spawn speak("Xavier here with your mail")
    with sequential execution centerOnDoor,
   terminate at monitorPickup completed;
  spawn monitorPickup()
    with sequential execution centerOnDoor;
}
```

An example of the Task Decomposition Language.

3T

Probably the most well known example of hybrid architectures is NASA's 3T [Bonasso et al 1997]. 3T consists of behavioral, executive, and planning levels; it is similar to the example architecture previously shown in the Hybrid Architecture section. The behavioral level is implemented using *skills*, which are essentially data-flow components whose inputs and outputs can be dynamically hooked up together. The executive level is implemented using RAPs [Firby87], which is a self-contained executive language implemented in Lisp. RAPs are used to decompose tasks into subtasks. At the bottom of the hierarchy are RAPs that interact with the skill layer, enabling and disabling skills and setting parameters. Skills interact with the RAPs layer by sending messages to update the symbolic RAP database, which in turn is used by RAPs to monitor the state of the world. Finally, an *adversarial* planner is used create high-level plans that are then passed on to the executive. 3T has been used to control manipulators and mobile robots, as well as in a number of space-relevant applications, including a free-flying camera (Aercam II) and a life support system (Bioplex).

3.2 Real-Time Control

Real-time control is controlling a system at a speed appropriate to the speed of the system (including sensing and motion) and environmental changes. Real-time control allows a system to effectively react to dynamic environments. [Brogan 1991].

Many powerful techniques in current research require minutes (or more) to determine the appropriate reaction to an event or sensor input and translate this into specific control inputs for the system. Such slow response can be disastrous in dynamic environments and lead to:

- Damage or loss of system: driving off a cliff before you can decide what to do about it.
- Damage or loss of manipulated objects: crushing the rock you wanted to investigate before you realized it was gripped.
- Damage or loss of features in the environment: driving over an interesting target before sensing it.
- Loss of opportunity: driving or flying past an interesting target before sensing it.

Real-time control capabilities exist for many applications, including trajectory tracking (for navigation and system configuration). Limitations in processing speed and representation prevent the ability to perform real-time control with symbolic manipulation.

3.2.1 Approaches

Several approaches are commonly used for real-time control. Two of these approaches, neural networks and expert systems, are derived from the artificial intelligence community; both of these approaches are defined in depth in the Underlying Technologies chapter of this document. Adaptive control is a branch of traditional control theory.

Neural Networks

Neural Networks can be used with great success for real-time control. Neural nets can be trained to respond to specific tasks, and the nature of the simple mathematical computations made by neural nets allow the system to make decisions quickly, even if systems are highly nonlinear and difficult to model. Neural nets can also learn by allowing on-line training, which allows the system to adapt to new situations or to changing system parameters. [Narendra and Parthasarathy 1990].

The Multimode Proximity Operations Device at the University of Maryland's Space Systems Laboratory (see also the Systems, Orbital section of this document) uses a neural network control system to guide a neutral-buoyancy craft.

Fuzzy Expert Systems

Expert systems, provided with an appropriate knowledge and rule base for the task, can also be used for real-time control. With the addition of fuzzy logic, such systems can quickly determine appropriate responses to situations that are similar to, but not identical to, situations covered in their rule base. While very large expert systems can be slow due to their dependence on inference, for specific control applications, real-time control is achievable. [Schneider et al 1996].

Adaptive Control

Adaptive control, rooted in traditional control theory, is the control of a system by updating the model of the system on-line. As system parameters change, the relationship between input and output also change. Monitoring such changes can allow a system to adapt its control parameters and achieve more accurate control. Like expert systems, this allows for handling of unexpected changes in the system itself. Since most adaptive control methods are computations based on relatively simple mathematical models, computing expected results in order to determine appropriate control inputs can occur quickly. [Hagan 1991].

3.3 Fault-Tolerant Distributed Robotics

In the extreme, hazardous environments encountered in space exploration, military operations, fire fighting, and nuclear cleanup, the likelihood that robots will be injured is amplified. In many situations, the danger posed is so great that a single robot that is expected to perform adequately in these scenarios must be designed to mitigate every conceivable circumstance. Clearly, this task is either very difficult or impossible for most operations.

A promising alternative approach is to use tightly coordinated groups, or colonies, of smaller, simpler robots to perform tasks in these dangerous locales. The fundamental advantage of this approach is fault tolerance through redundancy. If managed properly, the loss of single robots, although detrimental, need not be catastrophic; task execution capabilities will degrade gracefully across multiple robot failures. While very promising, the implementation of these ideas into a working system is fraught with the difficulties inherent in any highly redundant system. This section provides an overview of the class of redundant robotic systems that exhibit fault tolerance. Characteristics of these systems typically include homogeneity and scalability, while individual elements coordinate using cooperative or competitive strategies. This overview concludes with some sample applications and the most promising avenues for pursuit of fault tolerant robotic systems.

3.3.1 Definitions

Robot Colony: A coordinated group of robots working towards a common goal or task and that exhibit "spatially distributed parallel and concurrent perception and action." [Arkin and Bekey 1997].

Fault Tolerance: The construction of functionality (hardware or software) from redundant components in parallel and concurrent operation [Marciniak 1994]. For example, in N-version programming, voting is used to reconcile the outputs of different versions and instances of programs trying to accomplish the same task [Neuman 1995].

Cooperation: Barnes and Gray define cooperation as "joint collaborative behavior that is directed toward some goal in which there is common interest or reward." Cooperative agents typically share resources so that group performance is maximized. [Barnes and Gray 1991].

Competition: Busuioc defines competition as the attempt of an individual to "maximize its own interests." Interaction among competitive agents typically involves negotiation to resolve conflicts and bidding for tasks as the mechanism for competition. [Busuioc 1996].

3.3.2 Coordination Architectures

Two principle types of coordination architectures exist: *cooperative* and *competitive*. Cooperative architectures use top-down design principles to enable a group of agents to coordinate joint efforts. Competitive architectures rely on pressure from individuals to organize teams that cooperate within a competitive framework. Coordination in competitive systems results from increased incentive to work with other agents, for example, financial gain. This section explores the key tenets of each.

Cooperative Multi-Agent Architectures

Cuo identifies two principal key components of a cooperative system: the task and the form of cooperation. Cooperative agents typically rely on task decomposition as a means to provide a sequence of subtasks that can be contracted out to a multi-agent system such that parallel execution of subtasks is achieved at some level. As an example application, the task of mapping an area can be decomposed into subtasks, each concerned with mapping individual portions of the total area. In this manner, individual agents might construct a sub-map that is integrated with other sub-maps (from other agents) to complete the original task. [Cuo et al 1997].

The infrastructure for cooperation is most often based on communication. Cuo outlines three types of communication: environmental, sensor-based, and direct. In environmental communication, agents utilize features of the environment to enable a primitive form of communication. Sensor-based communication relies on the ability of an agent to distinguish between environmental features and other agents. This form of communication is similar to the "Dance" languages employed by honeybees were information about food-stores is communicated by the geometry and frequency of movement [von Frisch 1967]. The final method is direct communication, which involves the definition of a protocol, or language, that allow agents to exchange information directly.

In cooperative systems, groups of agents attempt to execute a task(s) using a type of communication to coordinate group behavior. These systems are not directly adversarial, in that there is no competition for the most desirable subtasks or conflict caused by behaviors (such as territoriality, etc.). Nevertheless, conflicts may arise as agents attempt to simultaneously access a group resource. Cooperative systems typically employ a rule-based conflict resolution mechanism that distributes shared resources. For example, in time–shared systems, each agent is given an equal slice of time to access a common resource. The key advantage of cooperative versus competitive multi-agent systems is the lack of background competitive pressure in cooperative systems. This allows the cooperative system to deliver harmonious and highly coordinated solutions to problems that maximize group, rather than individual, efficiency.

Competitive Multi-Agent Architectures

Coordination may arise from a competitive field. Examples of such systems tend to be economic or behavioral systems in which highly self-determined agents compete for survival by maximizing their gain in economical or biological terms. For an economic analog example, a company typically makes strategic partnerships with key vendors and customers to gain a market edge. Thus, cooperation is achieved in competitive arenas because individual agents "do not have sufficient capabilities or resources to complete problem solving alone" [Faratin et al 1998]. This is in direct contrast to the top-down framework for coordination that is exhibited by the cooperative robotics systems in which individuals have little intrinsic control over their network of collaborators or the terms of collaboration. In competitive systems, agent coordination is driven from the bottom-up in that individuals determine both the collaborators and the terms of collaboration.

In economy-based multi-agent systems, the concept of market price for goods and services forms the key component of the infrastructure for the circulation of task assignments and resources among the individuals [Wellman 1995]. In cooperative systems, conflicts between agents usually only arise over access to shared resources; conflict abounds in competitive architectures, as success is a function of a finite supply of incentive [Castelfranchi 1998]. The mechanism of choice for conflict resolution in competitive multi-agent architectures is negotiation. Negotiation is defined as "a process by which a joint decision is made by two or more parties. The parties first express contradictory demands and then move towards agreement by a process of making concessions and searching for new alternatives" [Pruitt 1981]. Thus, in spite of the initial pressure to maximize personal gain, economic systems have exhibited a large degree of coordination among individual participants. The promise of competitive over cooperative systems is reliability and efficiency of resource utilization that is not possible using a rigid coordination strategy.

3.3.3 Approaches

Several approaches to distributed robotics for fault tolerance have been investigated. Generally, these have been modeled on biological systems. Several of these approaches are discussed here.

CEBOT

Cellular Robotics System (CEBOT) is based on the organization of biological systems. In CEBOT applications, a group of distributed agents is arranged in a reconfigurable, hierarchical architecture. In CEBOT, master agents (master cells) are responsible for peer communication (talking to other masters) and coordination of subordinate cells. The network of agents that comprise a CEBOT system can be reconfigured such that the connectivity of the network changes in response to dynamics from its environment. Task assignments are delivered to master cells that are responsible for the coordination of subordinate cells for executing subtasks. [Fukuda and Nakagawa 1990].

SWARM

SWARM is an architecture for distributed systems with a large contingent of agents. Individual behavior is modeled as a local phenomenon stimulated by the actions of neighboring agents. Thus, SWARM architectures are primarily focused on networks of simple agents that require some degree of self-organization. The ultimate goal of SWARM architectures is to evolve a form of intelligence known as *SWARM intelligence*, similar to that seen in some insects. In this view, intelligent behavior can be synthesized from a group of non-intelligent agents. In SWARM architectures, communication is often modeled after the chemical communication systems exhibited in many insect species. Here, low-bandwidth pheromone signals trigger a spatially proximal response among interacting insects. As signals propagate from insect to insect, high-priority signals can achieve geometric growth in the number of notified insects. Thus, through local interaction a group of simplistic agents can achieve complicated behaviors and interactions with the environment. SWARM architectures attempt to endow robotic systems with similar properties. [Jin et al 1994].

DIRA

The Distributed Root Architecture (DIRA) aims to develop fundamental capabilities that enable multiple, distributed, heterogeneous robots to coordinate tasks that cannot be accomplished by the robots individually. The basic concept is to enable individual robots to act independently while still allowing for tight, precise coordination when necessary. Individual robots will be highly autonomous, yet will be able to synchronize their behaviors, negotiate with one another to perform tasks, and advertise their capabilities. [CMU a].

This architecture supports the ability of robots to react to changing situations and previously unknown conditions by replanning and negotiating with one another if the new plans conflict with previously planned cooperative behaviors. The resulting capability will make it possible for teams of robots to undertake complex coordinated tasks, such as assembling large structures, that are beyond the capabilities of any one of the robots individually. Emphasis will be placed on the ability of the system to reliably monitor and deal with unexpected situations and on the flexibility to dynamically reconfigure as situations change and robots join or leave the team. [CMU a].

Meta-Map

Meta-Map is a distributed mapping system designed for homogenous robotic agents tasked with urban reconnaissance and surveillance. The foundation of this application is a free-market architecture for coordination of a distributed robot colony [Dias and Stentz 1999]. In this architecture, the planner decomposes tasks into a series of subtasks and offers each to the colony members, promising reward upon completion of each contract. Individual robots compute each subtask's cost based on the relative amount of effort required to perform the subtask. Cost is a composite indicator of the physical expense a robot will incur in the execution of the subtask (distance traveled, estimated computational effort, power requirements, etc.). Robots will only bid on jobs that have a sufficient profit margin, subtasks for which the cost is less than the reward offered by the planner. Negotiation between the bidding agents and the planner results in the award of a contract that assigns subtask (with reward) to the robot agent with the winning bid [Faratin 1998]. Thus, the coordination of the colony can be modeled as the interaction of participants in a market economy; as such, it fosters both competition and cooperation among agents in an ad hoc fashion. Homogenous agents with spatial proximity will tend toward competition, while agents with complementary abilities will team to form a more effective unit.

3.4 Fault Detection and Diagnosis

This section provides an overview of the work that has been done in the area of automated fault detection and diagnosis. When the behavior of a system deviates from "normal" behavior,' the system is said to be in a fault state. Detection is the process of determining that a fault has occurred and diagnosis is the process of identifying what type of fault has occurred. It also includes determining where and when the fault occurred and the magnitude of the fault. This information is then used in taking recovery actions to return the system to normal operation.

3.4.1 Definitions

IFAC Technical Committee SAFEPROCESS proposed the following definitions to standardize the terminology used in this field, and we use these in this document. [Isermann and Balle 1997].

Fault: A non-permissible deviation of at least one characteristic property or parameter of the system from the acceptable, nominal, or standard value.

Failure: A permanent interruption of the system's ability to perform a required function under specified operation conditions.

Symptom: A deviation of an observable parameter from its nominal value.

Monitoring: A continuous, real time task of determining the conditions of a physical system, by recording information, recognizing and indicating anomalies in behavior.

3.4.2 Stages in the process

Detection

The detection process needs to determine that the system is in a fault state. The following methods are typically used for monitoring the state of the system, though other approaches have been applied.

Limit checking

This is a simple method that tracks whether or not key parameters are within allowable operating ranges. Although simple, this method suffers from the problem that these ranges might change as the system ages or when the operating conditions are different. For example, the normal current draw for a robot during locomotion is different when it is on flat ground versus when it is on inclined terrain.

Model Based

These methods maintain mathematical or analytical models of the system and compare the current state of the system against the model. If the current behavior deviates from what is expected by the model, the system is determined to be in a fault state. These methods are more flexible than limit checking methods, but are only as good as the model of the system.

Diagnosis

The diagnosis process needs to determine the description of the fault. This includes determining the type, location, time, and magnitude of the fault. Many methods of diagnosis have been developed. Several of those commonly applied are listed here. Details on many of these approaches appear in the Underlying Technologies chapter of this document.

Bayes Networks

When using Bayes Networks in diagnosis, the input nodes represent hypothesized faults and the leaf nodes represent symptoms. Intermediate nodes in the network represent tests that may be done to further isolate the faults. The conditional probabilities, which may be learned or obtained from expert knowledge, represent the causal relationships. These systems are used when one needs to determine the sequence of tests required to diagnose a fault. [Russell and Norvig 1995

Neural Networks

Neural Networks can be trained to recognize faults in a system. During the training phase, a set of corresponding symptoms and faults are used to train the network. The objective is to adjust the weights of the network so that when a set of symptoms is given to the trained system, it can provide the fault corresponding to it. Neural Networks work well in diagnosis problems since they can handle noisy and complex sensor data. The disadvantage of these systems is that the information stored in the net is very hard for humans to interpret or augment with expert knowledge. More about Neural Networks can be found in the Underlying Technologies chapter of this document.

Expert Systems

Expert systems store knowledge obtained from experts in the field as a set of rules. These rules capture causality relationships between symptoms and faults and use deductive reason to diagnose the faults. These systems can provide a reasoning process for the diagnosis that is understandable to humans. The complexity of these systems tends to grow considerably with the number of rules. The two most common techniques for building diagnostic expert systems are rule-based reasoning and set-covering. The rule-based approach uses a set of heuristic problem-solving rules, similar to those a human expert might use, to detect and isolate faults. It detects failures by matching actual performance data against failure symptoms contained in the rules and performs linear regression and correlation analysis on temporal data to detect incipient faults. Once a fault is detected, the rule-based system chains backwards through the rules, from the source to the symptoms, to isolate the problem. The set-covering technique contains a database, generated from the failure modes and effects analysis, that links known system failures to their known symptoms. It detects symptoms using rule-based classifiers, then searches the symptom-failure database to find the solution that best accounts for, or covers, the observed faults. While expert systems can quickly diagnose faults for which they have explicit prior knowledge, they are often unable to respond to conditions that were not foreseen when they were designed. More on Expert Systems can be found in the Underlying Technologies chapter of this document.

Model-Based

Model-based techniques use knowledge of the system's underlying behavior to diagnose faults. While expert systems rely primarily on heuristic knowledge, model-based systems contain design knowledge indicating how a system should behave. A model-based system typically cannot isolate faults with the speed of a set-covering or rule-based expert system, but it can isolate failures that were not anticipated when the system was built. Since all failures cannot possibly be accounted for at design time, a method for reacting to unforeseen failures is critical for autonomous space applications. Some model-based techniques model the behavior of the failed component, although most model the input versus output relationships of system components and systematically relax these assumptions to isolate failed components. More about Model-Based systems can be found in the Underlying Technologies chapter of this document.

Fuzzy Systems

Fuzzy Systems, instead of using binary logic as used in the expert systems above, use fuzzy logic which allows them to deal with imprecise data. More about Fuzzy Systems and Fuzzy Logic can be found in the Underlying Technologies chapter of this document.

3.4.3 Example applications

Livingstone

Livingstone is a model-based system developed at NASA Ames that was used to autonomously control the New Millennium Deep Space One Probe (DS 1) for part of the mission (see also the Systems, Deep Space/Heliocentric section of this document for information on Deep Space 1). Livingstone accepts a model of the components of a complex system such as a spacecraft or chemical plant and infers from it the overall behavior of the system. Livingstone also notes which commands are being given to the system and what observations are available. From this, Livingstone is able to monitor the operation of the system, diagnose its current state, determine if sensors are giving impossible readings, and recommend actions to put the system into a desired state even in the face of failures. [Williams and Nayak 1996, Williams and Nayak 1997].

MARPLE

MARPLE is an expert system that uses model-based reasoning. It has functions for capturing behavioral knowledge, a reasoning engine that implements a model based technique known as constraint suspension [Russell and Norvig 1995], and a tool for generating user interfaces. It has been demonstrated to work for the NASA LRC Space Station Freedom (SSF) power system test-bed. The primary objective for the power system's operation was to generate and dispatch electric power to the loads and maximize SSF's productivity without violating any constraints. [Roumeliotis et al 1998a, Roumeliotis et al 1998b].

The model-based system is constructed hierarchically. It isolates a problem at a certain level and then invokes an expert system for further diagnosis or to arbitrate between specialized expert systems that monitor limited areas.

Constraint suspension views the system to be monitored as a network of black boxes which are modeled as components in the MARPLE system. Constraints are placed on the behavior of each box. Both forward and reverse constraints calculate values for each node in the network, so that several values are calculated or each node. If the values at a particular model do no agree, MARPLE begins the constraint suspension isolation process. The components in the network are suspended, one at a time until a component is found that can account for all the inconsistent values at the nodes. The fault is then said to be isolated within that component. In MARPLE, rules may be defined to guide the model-based search. [Roumeliotis et al 1998a, Roumeliotis et al 1998b].

Envelope Learning and Monitoring via Error Relaxation (ELMER)

Traditionally, fault detection has been done by checking whether the values of certain critical parameters fall within acceptable ranges. These redline limits are often made wide to reduce the number of false detections, which can result in faults being detected only when the problem has become too critical to solve. Also, these bounds don't work well as the performance of the system degrades with age or the environment changes. ELMER is an algorithm that incrementally generates successively tighter bounds on acceptable ranges for these values. It adapts the bounds over the lifetime of the system based on incoming data. In its current implementation, ELMER works in a batch mode to adapt to system degradation. However, it doesn't yet detect when the operational conditions have changed to such an extent as to make the bounds learned from previous data useless. [Decoste 1997, Decoste 2000].

Multiple Model Adaptive Estimations (MMAE)

Kalman filtering is a well-known technique for state and parameter estimation. It is a recursive estimation procedure using sequential measurement data sets. Prior knowledge of the state (expressed by the covariance matrix) is improved at each step by taking the prior state estimates and new data for the subsequent state estimation. In MMAE, each estimator is a Kalman filter with a specific embedded failure model. The filter bank also contains one filter which has the nominal model embedded within it. The filter residuals are post-processed to produce a probabilistic interpretation of the operation of the systems. The output of the system at any given time is the confidence in the correctness of the various embedded models. [Fesq and Stephan 1989].

3.5 Navigation

Extreme environments, such as space and the surface of planets, pose special challenges for autonomous robots. Not only must the robot navigating in such environments avoid colliding with obstacles, such as rocks, it must also avoid falling into a pit or ravine and terrain that would cause it to tip over. Vast areas often have open spaces sparsely populated with obstacles where a robot might travel freely. However, the range of obstacles that can interfere with the robot's passage is broad; the robot must still avoid rocks as well as go around hills. Large areas are unlikely to be mapped at high resolution a priori, and the robot must explore as it goes, incorporating newly discovered information into its database. Hence, the solution to successfully navigating in such environments must be incremental by necessity.

Another challenge in navigation is dealing with a large amount of information and with complex vehicle dynamics. Taken as a single problem, so much information must be processed to determine the next action that it is not possible for the robot to perform at any reasonable rate. Planetary rovers carry very limited computing and thus it is important that sensor processing and decision making are as efficient as possible. One way to deal with this issue is use of a layered approach to navigation. That is, the complex navigation task can be decomposed into two simpler subtasks: local path planning (obstacle avoidance) and global path planning (achieving a goal location). The job of local planning is to react to sensory data as quickly as possible, avoiding hazards of various kinds. A more deliberative, global process, operating at coarser information resolution, determines how to steer the vehicle such that it can get to the goal, sometimes deciding to temporarily move away from the goal to reach the final destination safely.

3.5.1 Local Path Planning

Local navigation in such environments poses an additional challenge. Since the surface might be very cluttered, it is important for the system to discriminate between obstacles that must be avoided at all costs versus those that can be traversed but should be avoided if there is a choice. A conservative planner that regards all detectable objects as obstacles will not exploit the ability of the rover to drive over some obstacles, and may cause the rover to become trapped or go far out of its way unnecessarily.

Nearly all the research in local obstacle avoidance has been conducted for indoor robots, and much of the work in outdoor vehicles has used the assumption that the world is composed of obstacles and free space [Arkin 1987, Hebert 1997, Matthies et al 1995]. As described above, this assumption has serious consequences for navigation in rugged terrain, where sometimes the best (or only) choice is to surmount low obstacles, or to travel a short distance through somewhat rough terrain, rather than taking a long detour.

Recently, however, some researchers have begun investigation of using continuous measures of traversability in order to enable vehicles to make reasonable decisions. The Ranger algorithm at Carnegie Mellon University [Singh et al 2000, Kelly 1995] performed a simulation of a vehicle driving through the terrain. It analyzed the roll, pitch and high-centering of the vehicle as it moved along predetermined arcs. We have moved to a more statistical analysis of traversability, in part to better deal with sensor noise and in part to facilitate the merging of data over time; by merging traversability maps rather than terrain maps we reduce the effects of dead-reckoning error. The algorithm fits plans to blocks of the terrain computes the difference (residual) between the terrain and plane. The residual provides an estimate of terrain roughness; high residuals indicate rougher terrain. The residuals are computed at two levels of resolution, and can thus differentiate roughness at two different scales.

The work of Seraji uses linguistic terms (for example, "passable" or "highly-impassable") to represent traversability measures [Seraji 1999]. Fuzzy logic, in the form of expert-system-like rules, is used to decide how to move and turn to avoid obstacles. Several methods for producing the traversability measures from actual data are suggested, but it is not clear how the system (which was tested in simulation) actually calculates these terms. Gennery has proposed a method of traversability analysis for a planetary rover very similar to that of Carnegie Mellon [Gennery 1999]. His method also fits planes to small terrain patches and uses the plane parameters and plane-fit residual to estimate slope and roughness, respectively. The major difference is that Gennery's algorithm is that it is iterative, and thus more computationally complex.

3.5.2 Global Path Planning

Global navigation is the task of moving the rover from some initial location to a goal location. For the considered application of planetary navigation, the environment is presumed to be unknown or only partially known. Given the limitations on prior information, the rover cannot pre-plan a path that is optimal and guaranteed to reach the goal. Instead, the rover must acquire sensor information about the environment as it navigates and modify its plan accordingly.

One approach is to combine directed navigation with undirected exploration to learn a good route to the goal [Korf 1987, Pirzadeh and Snyder 1990, Thrun 1995]. This approach is most appropriate when environments can be traversed multiple times since the rover discovers better routes over time through trial and error. A second approach is to attempt to drive directly to the goal, locally circumnavigating obstacles along the way [Laubach 1998, Lumelsky and Stepanov 1986]. This approach requires little state information and is easy to implement. However, the approach does not make use of prior information, it assumes a binary world (i.e., obstacles and free space), and it can produce non-optimal paths. A third approach is to plan an initial path using all known information, making assumptions about parts of the environment that are unknown. As the rover acquires new information about the environment, the assumptions are updated with correct information, and the path is replanned [Singh et al 2000]. Generally, this approach works quite well for rover navigation, since it is able to make use of continuous cost information and prior map data, it works well in environments that are traversed a single time, replanning is needed only when the initial assumptions are proved invalid.

3.6 Planning and Scheduling

Planning is the process of constructing a set of actions that can enable an agent, or a system, to reach high-level goals. *Scheduling* is the process of ordering a set of actions so that it adheres to various temporal and resource constraints. Both planning and scheduling are needed to enable autonomous spacecraft to achieve complex tasks with limited resources.

Many planning and scheduling systems use similar basic representations. A *state* is a symbolic representation of the state of the world, including the agent, at a given point in time. A *goal* is a partial description of a state of the world that one wants to achieve. An *operator* is a representation of an action that can change the state of the world. Operators are typically defined in terms of their preconditions (what conditions must hold in the current state for the operator to be applicable) and post-conditions (the effects an operator has when it is applied). In addition, operators can specify the resources they need, including the time they will take to execute, and goals can specify their priority. A *plan* is a (partially ordered) set of actions that can achieve the given goals when executed.

3.6.1 Planning and Scheduling Paradigms

Even for the simplest operator and state representations, planning and scheduling is computationally costly (NP-complete) [Erol et al 1995]. Thus, planning cannot run in a real-time loop, since its performance cannot be guaranteed (see section on Architectures, this chapter). To make planning and scheduling tractable, several different paradigms have been developed. The most popular are methods based on refinement, transformation, and decomposition.

Refinement Planning and Scheduling

In refinement planning and scheduling, constraints are added to a plan or schedule incrementally, until eventually all goals are achieved, or all tasks are scheduled. Since planners based on refinement methods only add information, backtracking is needed in case the planner makes bad decisions. The advantage of refinement planning and scheduling is that the algorithms are usually fairly simple, and it is usually fairly straightforward to prove formal properties about the algorithm (such as soundness and completeness). Refinement-based scheduling is sometimes called *iterative repair* [Rabideau et al 1999, Chien et al 2000]. Within the refinement paradigm, algorithms differ along several dimensions. One dimension is whether the algorithm searches forward from the initial state towards the goals [Blum and Furst 1997, Kautz and Selman 1996] or backward from the goals towards the initial state [Fikes and Nilsson 1972, Penberthy and Weld 1992]. Another choice is whether to maintain a total order on the plan [Veloso et al 1995] or a partial order [McAllister and Rosenblitt 1991, Penberthy and Weld 1992]. Still another choice is whether to use a least commitment approach [Weld 1994] or to commit early to constrain the plan or schedule [Chien et al 1999]. No one choice is best for all domains -- the different choices have tradeoffs between the complexity of the algorithm, the time needed to generate a plan or schedule, the time needed to evaluate whether a plan or schedule is correct, and the flexibility in executing the plan.

Transformational Planning and Scheduling

In transformational planning and scheduling, constraints are both added (as in the refinement paradigm) and removed in the course of solving a problem [Simmons 1988, Hanks and Weld 1992, Beetz and McDermott 1994]. For instance, while refinement planners can add orderings between steps in a plan, transformational planners can reorder steps. While refinement planners can achieve the same effect by backtracking, in many situations it is much more efficient to change constraints directly. Thus, transformational planners and schedulers are more flexible, but this comes at a price: They are also more complicated algorithms and can have worse performance if not carefully controlled.

Another use for transformational planners is in *case-based* planning [Hammond 1989, Veloso and Carbonell 1993]. In case-based planning, a library of previously used plans is maintained. When a new set of goals is presented, the algorithm searches the library for a similar problem that it has solved in the past. The plan retrieved is then modified, using transformational methods, to achieve the new set of goals (and the new plan is then added to the library).

Hierarchical Decomposition Planning and Scheduling

Most practical planning and scheduling systems use hierarchical decomposition, which decomposes higher-level, more abstract operators in terms of sets of lower-level, more concrete operators [Wilkins 1988, Currie and Tate 1991]. Hierarchical decomposition (also called hierarchical task net, or HTN, planning) is a very powerful technique for solving complex problems because it enables the developer to indicate the way that goals should be achieved. For instance, a spacecraft engineer might know that in order to fire a thruster, the thruster must be warmed up, and then a valve must be opened, and then a bus command given. By using such procedural ("how to") knowledge, the planning algorithm can solve fairly complex problems by breaking them into simpler subproblems. This paradigm works well if the subproblems are relatively independent of one another (that is, solving one does not undo another). Most deployed planning and scheduling systems make use of hierarchical decomposition in one way or another.

Planning and Scheduling with Uncertainty

Recent research in planning and scheduling has dealt with handling uncertainty [Weld 1999]. Until recently, planning and scheduling systems assumed that the world was deterministic. Plans and schedules were simple sequences of actions. Any uncertainty was left to be handled by the executive and behavioral components (see section on Architectures, this chapter). More recently, planning and scheduling algorithms have been developed that represent operators using probabilistic representations [Kushmerick et al 1994, Blum and Langford 1999]. In addition, work has been done on creating *conditional* plans that branch [Drummond and Bresina, 1990, Peot and Smith 1992, Draper et al 1994] and *policies*, which are state/action mappings that indicate the optimal action to perform in every possible state that the system could be in [Schoppers 1987]. Planners based on Partially Observable Markov Decision Problems (POMDPs) combine probabilistic representations and policy-based plans to create very powerful and flexible systems [Kaelbling et al 1998]. Unfortunately, such planners are also very computationally expensive. Much research is involved in finding ways to solve POMDP problems in a tractable way [Parr and Russell 1995, Kaelbling et al 1998, Roy et al 1999].

3.6.2 Examples

Relatively little work in planning and scheduling has been used in space flight missions. The SPIKE scheduler was used to develop long-range schedules for the Hubble Space Telescope [Johnson and Adorf 1992] and SPSS does detailed planning of short-range segments. SPIKE is able to create multiyear schedules with as many as 5000 observations in less than an hour, including requests that are periodic. A similar system is Plan-ERS1 [Fuchs et al 1990], which does observational planning for the European Space Agency. Plan-ERS1 is based on O-Plan [Currie and Tate 1991], a hierarchical decomposition planner that has a rich vocabulary of resource constraints, including discrete, continuous, consumable, and renewable constraints.

The Remote Agent used a planner/scheduler based on the HSTS planner (also originally developed for the Hubble Space Telescope). The HSTS planner/scheduler uses hierarchical decomposition and iterative repair. It supports a very rich set of temporal and resource constraints. This makes it very expressive, but at a cost of planning complexity. HSTS represents high-level state variables as time lines and uses *tokens* to represent a constant (often symbolic) value of the time line over an interval. The algorithm tries to place the tokens such that all the resource and temporal constraints are met. Another feature of HSTS is that it represents the start and end points of tokens using ranges called *time windows*. For instance, it can say that some token starts between 12:00 and 12:05. This gives the system a lot of flexibility in both creating and executing schedules.

ASPEN (Automated Planner/scheduler has been used in a number of NASA applications, including planning for image analysis [Chien et al 1997] and scheduling of the Deep Space Network. Like HSTS, ASPEN uses iterative repair. Unlike HSTS, ASPEN does not use hierarchical decomposition because the problems it is designed to handle are not easily decomposable into independent subproblems. The design philosophy in ASPEN is to separate the domain knowledge ("how the system works") from search control knowledge ("what to do").

Most current planning and scheduling systems for space use work in batch mode, where the algorithm works to create a plan to a given horizon in the future, and then hands it off to an executive to be executed. In contrast, CASPER (Continuous Activity Scheduling Planning Execution and Replanning) is a continuous planner, which can replan very quickly (on the order of 5-10 seconds) in response to changes in the environment or changing goals. To make this concept work, CASPER is being more tightly integrated in with the executive component, blurring the distinction between them [Chien et al 1999].

Additional details on SPIKE, SPSS, ASPEN, and CASPER can be found in the Systems, Tools section of this document.

3.7 Human-Machine Interfaces

Human-machine interfaces are the mechanisms through which operators can interact with machines or with virtual machines. They operate as translators from the human world (typing, motion, voice, etc.) to the machine world (motion, sensing, etc.). Interfaces are required for any form of communication with machines, be it interaction with and control of a far-removed robot or simulated control of and feedback from a design for testing. For the ease of the humans interacting with the machines, interfaces are desired to be "natural" in some way, presenting information that can be interpreted intuitively and allowing the user to respond similarly. As a result, today visual and speech interfaces are replacing the keyboard and raw data. For the area of space robotics, focus has been placed on visualization; these visual methods of interfacing between humans and machines are the focus of this section.

3.7.1 Visualization

Visualization is the process of converting computer data into a visual domain so that a user can readily interpret it. Visualization has become the predominant mode of human-machine interface, as it is both the most natural means for humans to interpret an environment and the foundations for visual feedback (in the form of text and pictures on a screen) are well in place.

The level of complexity of visualization has drastically expanded in recent years. Construction and rendering of three-dimensional models from stereo imagery have increased the level of telepresence achievable and assisted science that can be performed on downlinked robot data [Stoker and Zbinden 1999]. Concepts from the schools of human-computer interaction, graphic design, and psychology have drastically increased the usability of visual user interfaces. The result of these advancements is a broad range of visual interfaces and a broad range of applications.

3.7.2 Control

For space systems, control of systems can be via teleoperation, remote control, or high-level commands. Visualization can be useful in different means for each of these. For deep space missions, where direct teleoperation is not feasible, visualization can aid planners in monitoring the health and state of the craft and in selecting high-level goals to send to the robot. For closer operations, such as on Mars or the Moon or in orbit [Kuester and Lane, 1995], teleoperation can be used.

Teleoperation and remote control of vehicles and spacecraft requires reliable knowledge of the state of the vehicle and its environment. Visualization of the craft and its environment can allow the user natural access to such information in order to increase reliability and speed in controlling the vehicle. In particular, a three-dimensional model of the robot slaved to the actual robot motion can give the operator a better understanding of robot state.

Examples of visualization for robot control include control of manipulators, spacecraft, and rovers. DLR's Rotex is a robot arm that is controlled with an image of the arm presented to the user. WITS is a three-dimensional simulation of a craft and its environment that allows direct control of the craft by moving it in the image. DartsShell models spacecraft dynamics in a visual way to test designs or predict outcomes for planning. Viz is another three-dimensional visualization tool that can be used for interactive control of space vehicles. Details on each of these systems are presented in the Systems chapter of this document.

3.7.3 Data Integration

Users can get a much better understanding of a set of images from a probe if they are integrated into a single mosaic. Similarly, a more complete picture of an environment can be obtained if the data from several different instruments can be combined into a single image.

The ability to combine multiple images into a single panoramic mosaic was well demonstrated with the ground control software of the NASA Mars Pathfinder mission in 1997. The 3-D terrain models generated by Viz from a combination of stereo sensors on the lander and the Sojourner rover provided a much more intuitive interface than raw images, for both scientists and engineers on the mission. Shown is a photo-realistic terrain model of the Mars Pathfinder landing site generated by this method. [ARC a, Glombek et al 1997].

A simulated view of the Martian terrain and the Pathfinder lander on Mars generated by Viz.

Big Signal Antarctica 2000 (also discussed in the Systems, Tools section) is an example of a system that combines the data from several different sensors into a single visual representation to be studied by observers. WITS also allows for data integration by combining the state of the robot and local terrain map information into a single simulated image.

3.7.4 System Design

Computer-assisted drafting (CAD) techniques can be extended to allow moving parts to be modeled. For example, when designing a manipulator arm, the user can interactively set joint angles to determine angle limits and the effective workspace. One example of commercial software in this area is Deneb Robotics' ENVISION package. Johnson Space Center's Engima has also been used for design of robotic arms; more information on Enigma is provided in the Systems, Tools section.

3.7.5 Simulation of Sensor Data

Many robots rely on visual sensing. In order to simulate closed-loop control of these robots, the simulator must be able to generate images corresponding to the current world state. This technique is used for multi-robot visual servoing on the Distributed Robotic Architectures project at Carnegie Mellon University, as discussed in the Systems, Tools section. In the process of generating images, many graphics packages also calculate depth information, which can be used to simulate a range sensor. WITS and Enigma can also simulate sensor data when provided with a model of the system and its sensors. Viz provides simulation of sensor data as well, by combining information from different images and simulating what can be seen in a computer image.

3.7.6 Future Work

In the future, humans and robots will work together to accomplish tasks. This scenario presents a very different set of requirements than telerobotic interfaces. When humans work alongside robots in space (such as during orbital operations or EVA), intuitive and non-invasive interfaces are required. These human-robot interfaces must distill a humanly manageable quantity of high-level, relevant, and naturally understandable explanations from a much larger set of raw, technical, robotic data. The human user will need to attend to his or her task as well as communicate with a robot; so vigilant monitoring of the robot's state is unacceptable. Important technologies include wearable computers, speech recognition, natural language processing, expert systems, and automated presentation techniques. Robots working alongside humans also should able to recognize and learn from gesture [Lee and Xu 1996, Voyles and Khosla 1999] and other forms of nonverbal communication.

4 Systems

Systems relevant to the field of space autonomy are discussed in this chapter. In this context, *system* is used as a general term to include spacecraft; planetary surface systems (rovers and landers); intelligent software for space systems; and spacecraft, lander, and rover system prototypes. The systems discussed in this chapter are organized into sections according to type: deep space and heliocentric spacecraft, planetary orbital and surface systems, systems for human assistance, and software tools. Deep Space missions include missions to visit asteroids, comets, more than one planetary body or moon, or to leave the solar system. Heliocentric missions are those craft that maintain an orbit around the sun rather than a planet or other body. Planetary missions include those that orbit a single body or land on a surface, including those missions to Earth's Moon. Orbital missions include those that remain within Earth's orbit. Human Assistance projects are those which are provided to extend the abilities of humans for such things as EVA, inspections, and other typically human space activities. The Tools section includes software tools designed to assist in various aspects of space missions.

The included deep space, heliocentric, and planetary systems are meant to include all such systems past, current, and future, as well as some proposed. Directly controlled systems in these categories are discussed here to demonstrate the pervasiveness of the direct control approach and to include current missions of interest. Orbital and Human Assistance systems are included if they possess a level of autonomy (teleoperated or higher); many other such systems with direct remote controlled operation, such as common science data collection satellites operating in Earth's orbit and communications satellites, are not specifically discussed. The Tools section is limited to recent tools that exhibit autonomous or semi-autonomous functionality and that are specifically designed with space robotics applications in mind.

Each section is subdivided by the system's level of autonomous activity. Systems that are highly or completely self-reliant are considered *Autonomous*. Systems with some level of self-reliance but that depend regularly on human input are considered *Semi-Autonomous*. *Teleoperated* is the designation for systems that are primarily guided by humans, but with some level of autonomous assistance built-in (such as obstacle avoidance, script execution, or joint control); this designation is primarily reserved for complex mechanisms such as planetary surface systems. The lowest level is considered *Directly Controlled*, where an operator gives a system. It is worth noting that the predominant mode in spacecraft is still remote control because of its relative lack of complexity, low cost, and reliability.

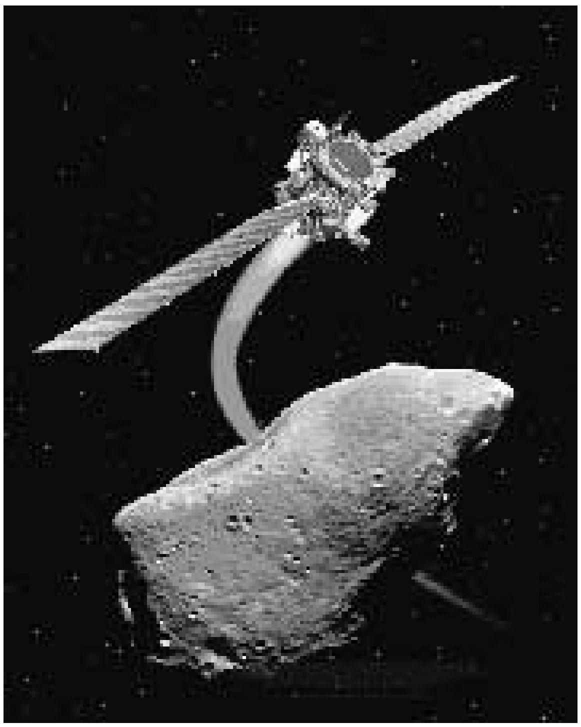
4.1 Deep Space and Heliocentric

4.1.1 Autonomous

Deep Space 1 (DS-1)

NASA's DS-1 is the first highly autonomous space mission. An autonomous celestial localization module named AutoNav determines DS-1's location in space; this is the first spacecraft with the capability to self-localize in space. Updates to the craft's position estimate are computed weekly. The Remote Agent planner autonomously plans course corrections in order to achieve goals and deploys or enables science instruments at locations appropriate to the specified science targets. It also enables the spacecraft to work around many problems that arise or request help, when needed. Deep Space 1 also uses a method of autonomous fault detection, a system named Livingstone. More information on AutoNav and Remote Agent can be found in the Systems, Tools section; Livingstone is also discussed in the Component Technologies, Fault Detection and Diagnosis section.

Deep Space 1 was launched in October of 1998. It is primarily intended as an autonomous technology demonstrator, as well as a demonstrator of the new ion propulsion system. Its goal is to perform scientific investigations of asteroids and comets in deep space, at distances that minimize the ability for intervention from Earth. The first target for Deep Space 1 was the asteroid Braille, which it encountered in July of 1999. During the flyby at a distance of 26 miles, the spacecraft took images and conducted studies of mineral composition, size, shape, and brightness. Additionally, it searched for changes in the solar wind resulting from interaction with the asteroid in order to determine the presence or absence of a magnetic field. The next targets are the comet Wilson-Harrington and comet Borrelly, where it will take close-up images, determine the size and shape of their nuclei, study their comas (the cloud of water and gases that surrounds a comet's nucleus), examine the relationship of the surface features of the comet's nucleus to its dust jets, and study the interaction of the comet with solar wind.



Artist conception of Deep Space 1 making a close fly-by of an asteroid.

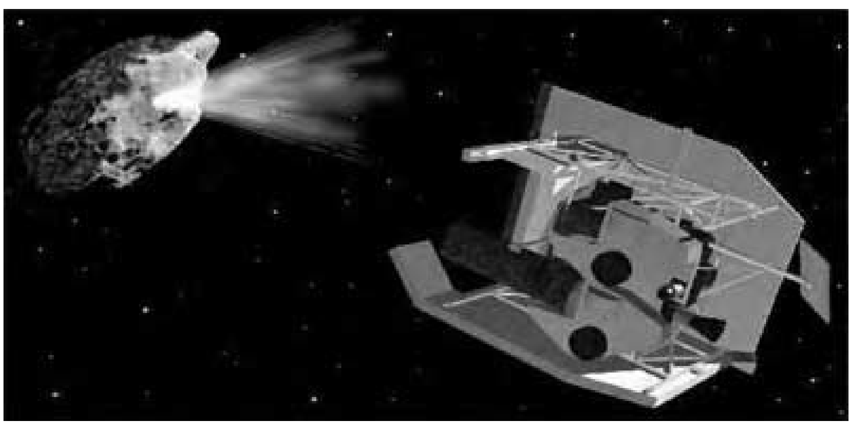
4.1.2 Semi-Autonomous

Deep Impact

Deep Impact was selected to be the eighth flight mission in the Discovery Program, a NASA program seeking to enhance our understanding of the solar system through exploration of planets, moons and other small bodies. Deep Impact is a joint effort involving many universities and companies, with primary collaboration between the University of Maryland and JPL. Deep Impact, scheduled for launch in 2004, will view deep into the interior of a comet nucleus. The spacecraft will fly by the asteroid and release a smart impactor. The craft is protected from faults with large margins and substantial redundancy.

The impactor released by Deep Impact is equipped with an autonomous active guidance system that will choose an impact site on the sunlit side of the comet surface and steer it to that site for impact. The impactor will produce close-up images of the comet's surface prior to impact and relay them to the spacecraft. Upon impact, it will excavate a deep crater in the cometary nucleus.

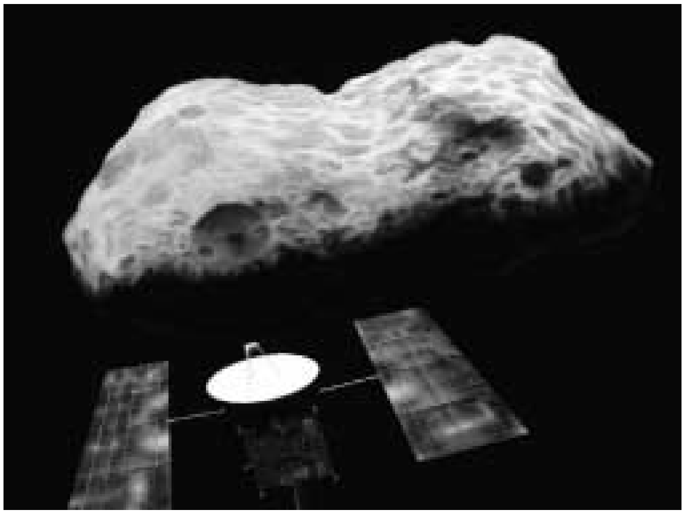
The spacecraft, equipped with optical and infrared instruments, will spectrally map the impact and resulting crater. It will observe how the crater forms, document the final state of the crater, measure the composition of the hot ejecta from the crater, and determine the changes in natural outgassing produced by the impact. All data from the spacecraft and the impactor will be transmitted to Earth by the spacecraft, which will take up to ten hours.



Artist's conception of Deep Impact as it approaches a comet for impact.

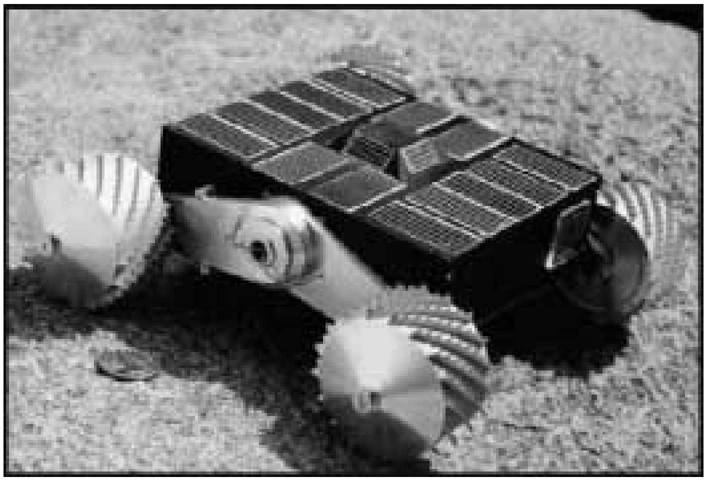
MuSpace Engineering Spacecraft and Nanorover (Muses-C/Muses-CN)

The MuSpace Engineering Spacecraft and Nanorover will incorporate the most autonomous planetary operations to date with autonomous navigation for landing and autonomous science data collection on a rover platform. The small near-Earth asteroid 4660 Nereus is the target of the Muses-C mission, the world's first asteroid sample return. The mission is a collaboration between ISAS and JPL. Muses-C is the ISAS spacecraft and Muses-CN is the JPL rover. Muses-C will acquire a sample of the asteroid and return it to Earth for analysis. The Muses-CN nanorover, which is dropped just before touchdown, will move around the surface of the asteroid. The launch is scheduled for January 2002 and is anticipated to arrive in April 2003.



Artist's conception of the Muses spacecraft as it approaches 4660 Nereus and prepares to land.

Muses-C will provide the first space flight demonstration of several new technologies and techniques. It will use solar electric propulsion to travel to the asteroid. At the moment of touchdown, the Muses-C spacecraft will fire a small pellet into the asteroid and collect ejecta thrown off through an inverted funnel, storing it in the sample-return capsule onboard.



Photograph of the nanorover for the Muses-C mission, shown next to a US quarter for scale.

The Muses-CN nanorover autonomously senses its environment and controls its operations. With a mess of 1 kg, it is the smallest rover ever to fly on a space mission. It is mobile and able to navigate in low gravity environments. Nanorover is equipped with an imager and instruments that observe in both the visual and near-infrared wavelengths to provide information on the elemental composition of the asteroid's surface. The nanorover is designed to survive and actuate in temperature ranges of –125°C to +125°C without thermal enclosure or control. The Muses-CN rover prototype is also being modified to enable operation on other targets such as comet nuclei, moons around other planets, and Mars.

Space Technology 3 (Deep Space 3)

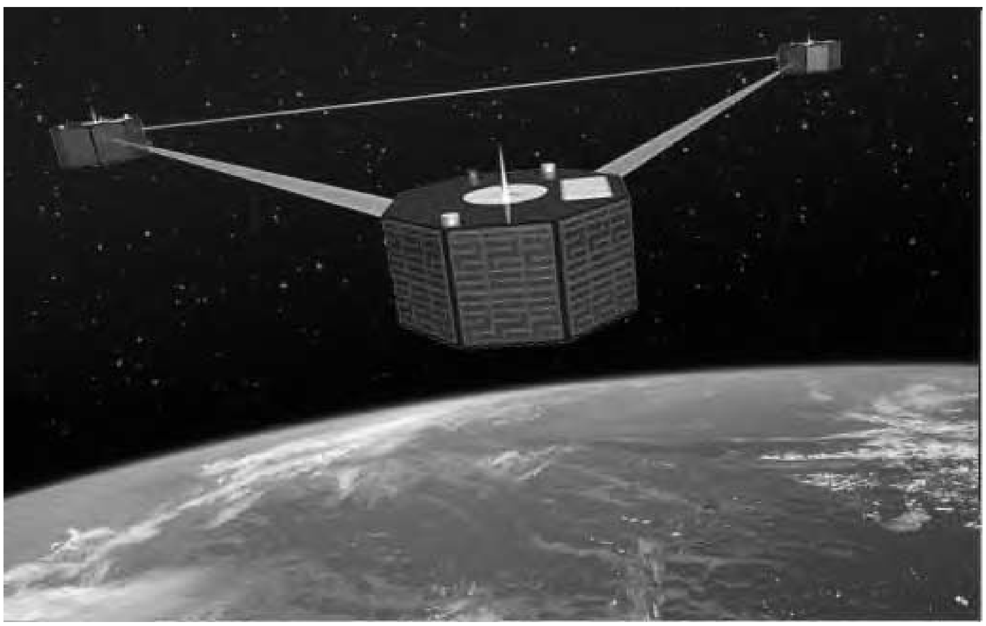
JPL's Space Technology 3, part of NASA's New Millenium program, will validate advance technologies for future spacecraft and instruments, including autonomous maintenance of satellite constellation formations. In order to find and ultimately study Earth-like planets around nearby stars, it is necessary to separate an interferometer's collecting apertures by large baselines of hundreds of meters to thousands of kilometers; thus, this first spaceborne stellar interferometer will consist of two spacecraft flying in formation. Control at the nanometer level demands precision spacecraft controls, active optics, metrology and starlight detection technologies. To date, some of these technologies have been demonstrated only in ground applications with baselines on the order of a hundred meters. Space operation will require a significant capability enhancement. It is anticipated that some part of the Remote Agent developed for Deep Space 1 will be employed.

Artist's conception of the Space Technology 3 telescope satellites flying in formation to perform stellar interferometry.

The mission's primary science targets are Be stars, Wolf-Rayet stars and cool M-dwarf stars. The Space Technology 3 mission will be launched into space in the year 2003 aboard a Delta 7325 rocker. The revolutionary technologies demonstrated during the experimental flight will be used by other sophisticated interferometer NASA missions planned for the next few decades. Both ST3 spacecraft will be launched from a single launch vehicle into a heliocentric orbit to trail behind the Earth in late 2003.

Space Technology 5 (Deep Space 5)

GSFC's Space Technology 5 is part of NASA's New Millennium Program. It will attempt to fly three miniature spacecraft in formation high above the Earth with autonomous design and maintenance of constellation formations. Each of the "nanosatellites" is about the size of a birthday cake and will be used to test methods for operating a constellation of spacecraft as a single system. The three nanosatellites will perform coordinated movements, communication and scientific observations as if they were a single larger spacecraft. The nanosatellites will have to autonomously stay in constant contact with each other, sharing information and reconfiguring onboard instruments and systems to behave as a single unit to achieve the complex communication path from a constellation of spacecraft in flight high above Earth to communication stations on the ground. An autonomous ground station will automatically operate the set of spacecraft and determine its orbit. While primarily a mission for technology demonstration, Space Technology 5 will collect scientific data to study particles in the Earth's magnetic field and the Earth-Sun interaction.



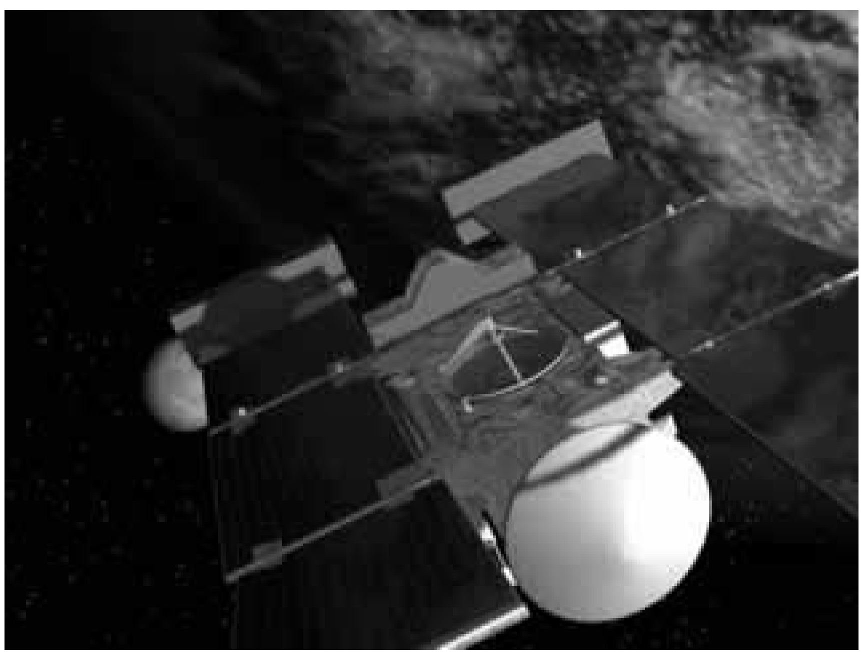
Artist's conception of the Space Technology 5 nanosatellites flying in formation to study Earth's magnetic field.

Other technologies that Space Technology 5 will test that will enable future generations of satellite constellation missions include: communications components for small spacecraft; multi-functional structures for the (4.0) satellite technologies; ultra low-power demonstration; variable emittance coatings for thermal control; propulsion systems components; and lithium-ion power system for small satellites. The mission is planned for launch in 2003 as a secondary payload on an expendable launch vehicle. Results from the mission will then be used to fly larger constellations of nanosatellites in future missions.

Stardust

Photograph of the sample return capsule from the drop tests.

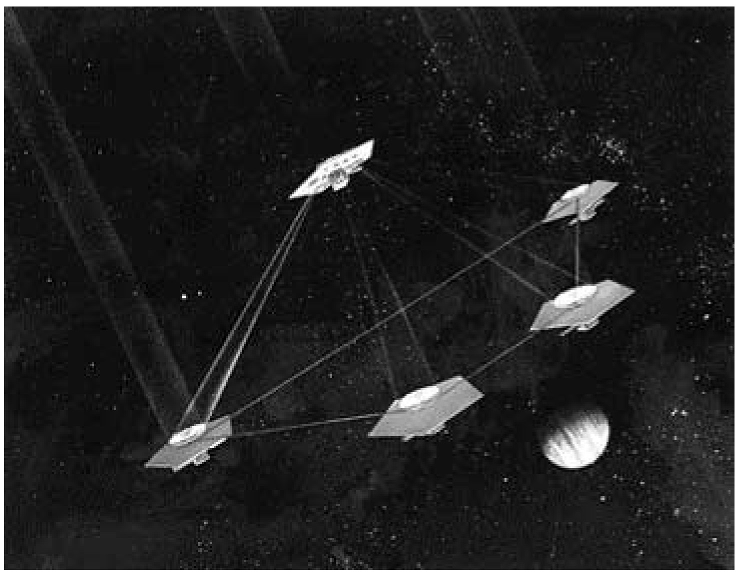
Stardust is a semi-autonomous NASA Discovery collaborative mission among JPL, University of Washington, and Lockheed Martin that will fly close to a comet bringing back to Earth cometary material and interstellar dust sample return. It is the first robotic return of extra-terrestrial material from outside the orbit of the Moon. Its primary goal is to collect dust and volatile samples during a planned close encounter with comet Wild 2. Additionally, the Stardust spacecraft will bring back samples of interstellar dust including the recently discovered dust streaming into the solar system from the direction of Sagittarius. Flybys of the comet will be accomplished by on-board optical navigation; images to monitor the trajectory will be transmitted at intervals that gradually change from weekly to hourly as the comet is approached. On-board systems also will deploy the dust-collection instrument autonomously and keep the camera and mirror pointed at the comet. Stardust was launched on February 7, 1999 and should encounter Wild 2 in January 2000.



Artist's conception of the Stardust spacecraft as it leaves Earth's orbit.

Terrestrial Planet Finder (TPF)

NASA's TPF will use multiple spacecraft to study all aspects of planets including their formation and development in disks of dust and gas around newly forming stars, the presence and features of those planets orbiting the near stars, the numbers at various sizes and places, and their suitability as an abode for life. The mission would include multiple spacecraft in autonomous formation flight. By combining the high sensitivity of space telescopes with the sharply detailed pictures from an interferometer, TPF will be able to reduce the glare of parent stars by a factor of more than one hundred-thousand to see planetary systems as far away as 50 light years. By combining the sensitivity of the Next Generation Space Telescope with detailed imaging, TPF will be able to study the winds from dying stars that enrich the material between the stars with life-enabling heavy atoms (like carbon and nitrogen) and will be able to see the cores of quasars and the black hole at the center of Earth's Milky Way. The specified spacecraft formations will be reached and maintained using the Global Positioning System (GPS).



Artist's conception of the Terrestrial Planet Finder satellites working in formation.

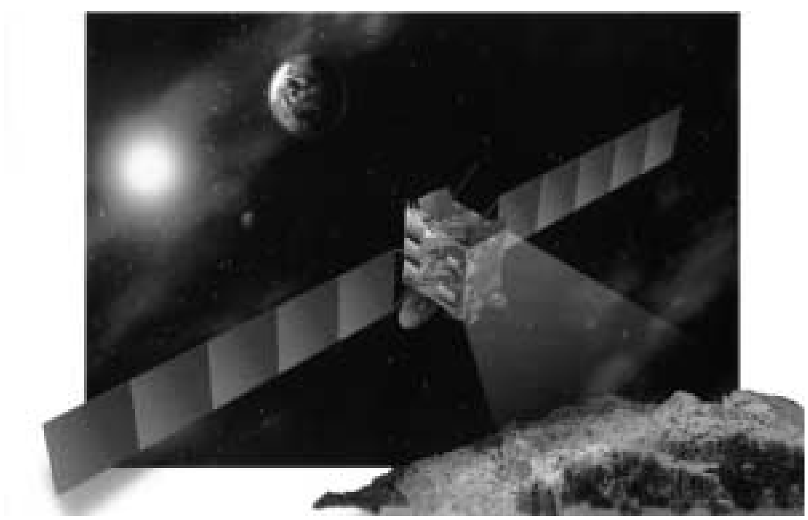
4.1.3 Teleoperated

Rosetta

Rosetta is ESA Horizon 2000's third cornerstone mission to rendezvous with comet 46 P/Wirtanen and perform remote sensing investigations. Rosetta will conduct flybys of two asteroids, 4979 Otawara and 140 Siwa on its way to the comet, where it will release a probe to land on the comet's surface and perform in situ measurements. The spacecraft has autonomous self-monitoring and maintenance. Rosetta's on-board instruments will be autonomously pointed toward the targets and be able to carry out a variety of experiments and examinations on the comet. It will study the appearance of the comet's surface, its composition and temperature distribution and analyze the gas and dust emitting from its nucleus. It will determine dust and gas emission rates, and investigate the interaction with the solar wind.

The Rosetta Spacecraft, with solar panels stowed.

Rosetta is scheduled for launch in January 2003 on board an Ariane-5 rocket. It will take eight years to reach Comet Wirtanen and then will orbit it for the following two years.

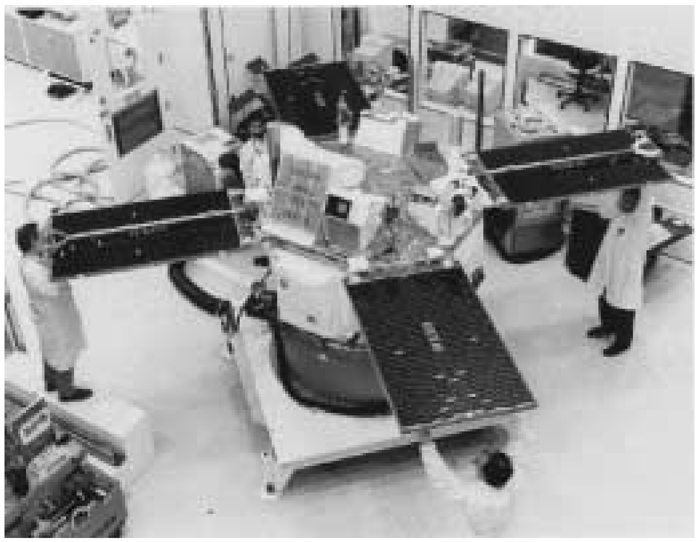


Artist's conception of the Rosetta spacecraft approaching the comet, where it will release a lander.

4.1.4 Directly Controlled

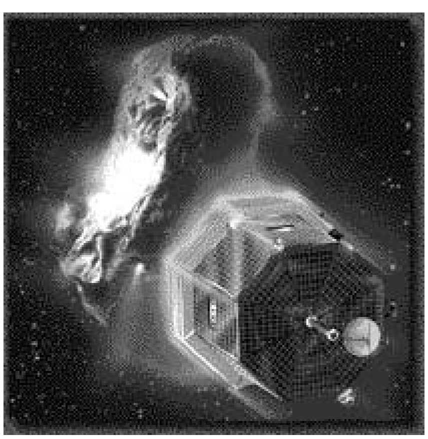
Advanced Composition Explorer (ACE)

ACE was developed for GSFC by UMD and JHU APL. It is a spacecraft that carries six high-resolution sensors and three monitoring instruments and will sample low-energy particles of Solar origin and high-energy galactic particles with a collecting power 10 – 1000 times greater than in past or planned experiments. The mission was launched in August 1997 and is still in operation.



The Advanced Composition Explorer spacecraft, with solar panels deployed.

Comet Nucleus Tour (CONTOUR)



NASA's GSFC, APL, and Cornell University are the primary participants in developing the Comet Nucleus Tour, CONTOUR, a mission that is intended to greatly expand what is known of comet nuclei and to assess their diversity through a series of flybys of three comets, Encke, Schwassmann-Wachmann-3, and d'Arrest. The CONTOUR mission is capable of being retargeted to approach an unforeseen cometary visitor. At each comet flyby, the spacecraft will take high-resolution pictures. The launch date is scheduled for July 2002.

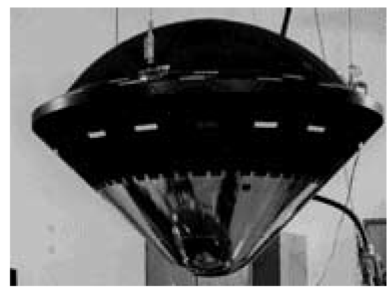
CONTOUR flying by a comet, artist's conception.

Galileo

The Galileo spacecraft deployed from the Shuttle for studies of Jupiter and its moons.

Galileo is a JPL spacecraft on a flyby mission to Jupiter, Europa, and Io for imaging and science data. When launched from the Space Shuttle in 1989, the spacecraft consisted of an orbiter and a probe. The two parts of the spacecraft made the journey to Jupiter together. For both the probe and the orbiter, trajectories and data collection (including imaging) are controlled from Earth. All atmospheric scientific experiments are conducted passively; collected data is returned to Earth for processing and analysis.

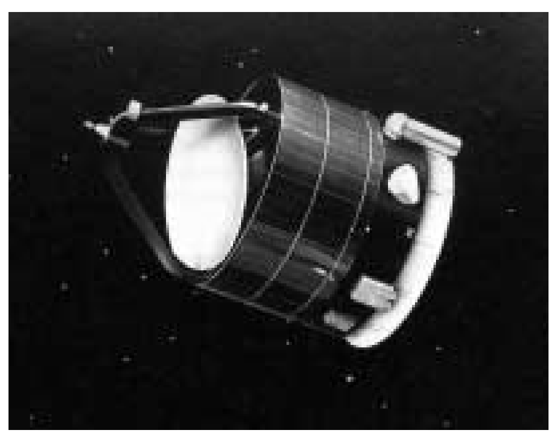
Upon arrival at Jupiter on December 7, 1995, Galileo released a small science probe on a trajectory into Jupiter's atmosphere to determine the atmosphere's composition. The Galileo orbiter entered into an orbit around the planet that allows it to encounter the major satellites. The orbiter is expected to continue its studies beyond 2000.



The Galileo probe in testing.

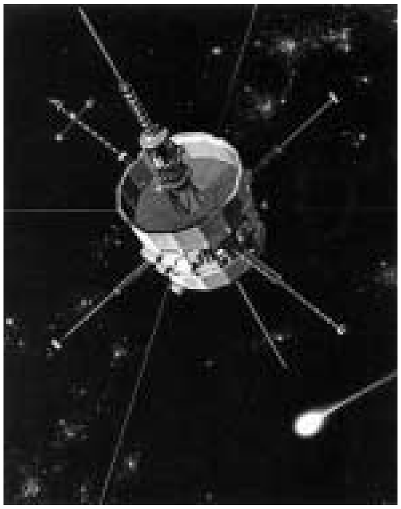
Giotto

Giotto was the ESA's first deep space mission. Along with a suite of probes from the Soviet Union, Europe, and Japan, Giotto conducted a flyby mission to perform scientific observation and imaging of Halley's comet during its closest passage to the Sun. The mission was launched in July 1985. It was the first spacecraft to encounter two comets and the first deep space mission to change orbit by returning to Earth for a gravity assist. It provided the first close-up images of a comet nucleus.



Artist's conception of the Giotto craft to Halley's comet.

International Sun-Earth Explorers (ISEE, ICE)



ISEE and ICE were a group of three spacecraft developed by Goddard Space Flight Center and ESA to study the interaction of the solar wind with the Earth's magnetosphere. Two of the spacecraft (ISEE 1 and 2) were in Earth-like heliocentric orbits, and the third (ICE) moved between the Lagrange point and the Earth's orbit for large-baseline measurements of solar wind and the Sun-Earth relationship. The first two spacecraft were launched in 1978.

The International Sun-Earth Explorer spacecraft, artist's rendition.

Laser Interferometer Space Antenna (LISA)

The JPL and ESA Laser Interferometer Space Antenna (LISA) mission will investigate gravity waves. Three Earth-controlled spacecraft, separated by 5,000,000 km, will fly in triangular formation in heliocentric orbits at 1 AU. The three spacecraft will perform as a very-large-baseline interferometer for measuring the distortion of space caused by passing gravitational waves. Effects due to sunlight variations will be reduced using a proof mass that is shielded from direct sunlight and laser interferometry the will measure the distance between the shielded proof masses in different spacecraft. Launch is planned for 2008.

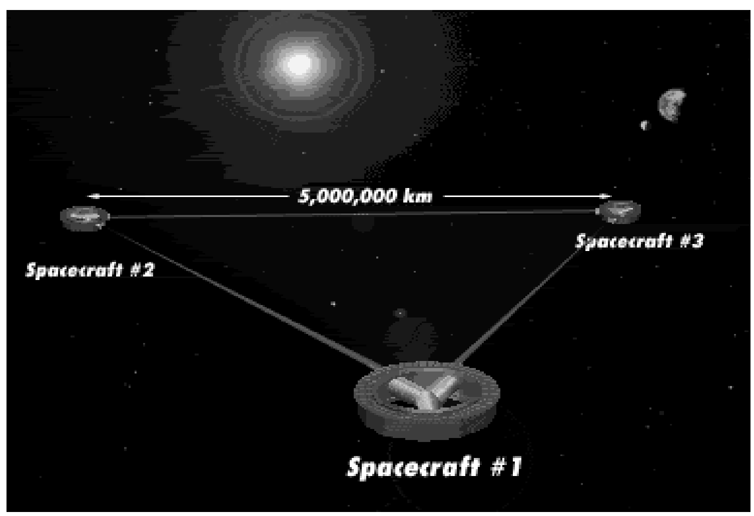


Illustration of the LISA multi-spacecraft laser interferometer.

Near Earth Asteroid Prospector (NEAP)

SpaceDev's Near Earth Asteroid Prospector (NEAP) is the first planned deep-space mission to be wholly defined, executed, and funded by commercial entities. The craft will orbit and land on the Nereus asteroid and other targeted asteroids with the intent to claim one for commercial resource prospecting. NEAP is designed to carry a mix of science, engineering, and "novelty" payloads as several attached and two ejectable packages. NEAP is scheduled for launch in the fall of 2001.

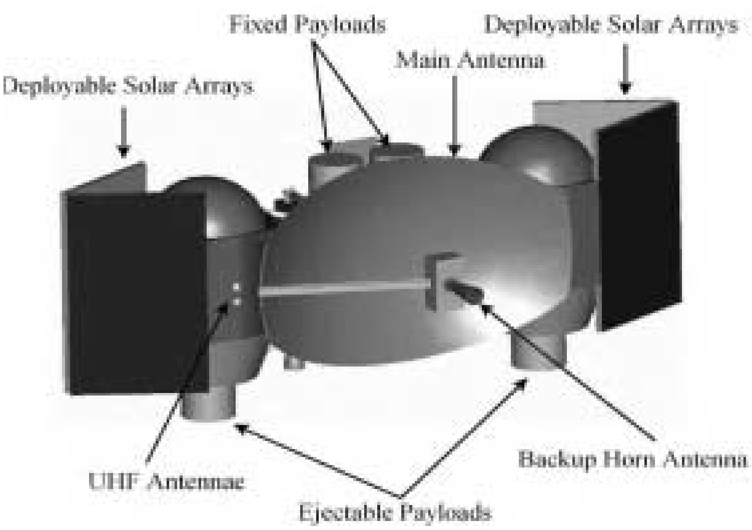


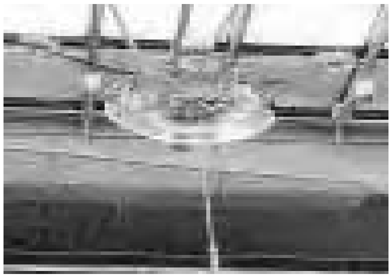
Illustration with components of the NEAP, the first commercial deep space craft..

Near Earth Asteroid Rendezvous – Shoemaker (NEAR)

Artist's conception of NEAR craft for asteroids investigation.

The Near Earth Asteroid Rendezvous (NEAR) mission was the first of NASA's Discovery missions, and the first mission to go into orbit around an asteroid. JPL, APL, and GSFC are collaborating on this flyby mission of the Eros asteroid for imagery and spectroscopy to determine the nature and composition of the asteroid. The spacecraft is equipped with an X-ray/gamma ray spectrometer, a near-infrared imaging spectrograph, a multi-spectral camera fitted with a CCD imaging detector, a laser altimeter, and a magnetometer. It was launched in February 1996.

Pioneer

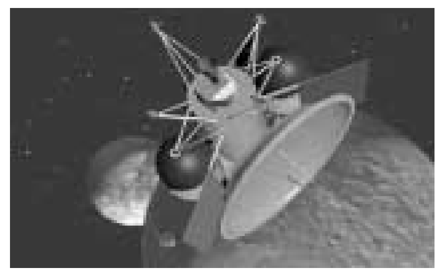


Pioneer 10 spacecraft.

NASA ARC's Pioneer series were the first spacecraft to perform close flybys of the asteroid belt, Jupiter, and Saturn. Pioneer 10 obtained close-up images and data on high-energy particles and magnetic fields on asteroids before continuing on and performing similar experiments at Jupiter. It was the most remote object ever made at the time of launch, March 1972. Launched in April 1973, Pioneer 11 followed Pioneer 10 to Jupiter. It continued on to make the first direct photographic and scientific observations of Saturn, and studied energetic particles in the outer heliosphere.

Pluto-Kuiper Express

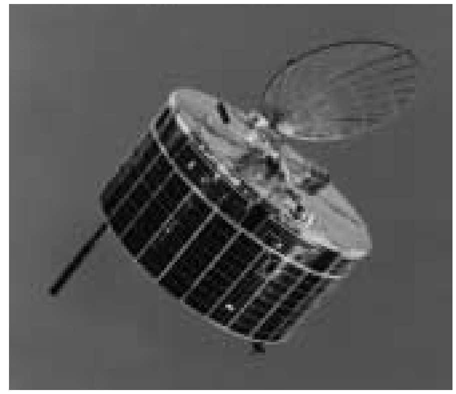
Pluto-Kuiper Express is a JPL and GSFC mission currently under investigation that is designed to fly by and study Pluto and its satellite Charon. The studies of Pluto and Charon will include imaging, mapping, and compositional determination. It will then fly on to encounter one or more of the large bodies in the Kuiper asteroid belt, which lies beyond the orbit of Pluto, and conduct similar experiments. The spacecraft will use lightweight advanced-technology hardware components and advanced software technology. It is intended for launch in December 2004.



Pluto-Kuiper mission to study Pluto and the Kuiper asteroid belt, artist's conception.

Sakigake (MS-T5, Pioneer)

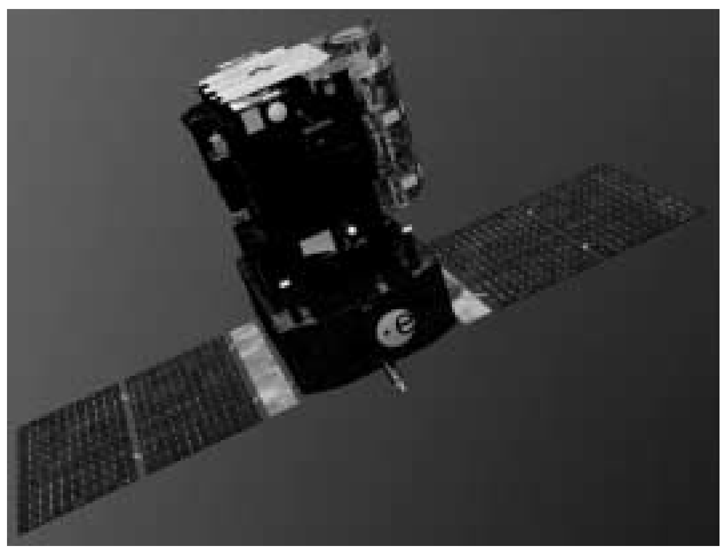
Sakigake was an ISAS spacecraft, a heliocentric orbiter that collected data on plasma wave spectra, solar wind ions, and interplanetary magnetic fields. It was a technology demonstrator aimed to prove the performance of Japan's new launch vehicle and test the schemes of the first ISAS escape from Earth's gravitation. It was launched in January 1985. The spacecraft was identical to that used for Planet-A/Suisei, also launched in 1985.



Artist's conception of the Sakigake Solar orbiter.

Solar and Heliospheric Observatory (SOHO)

SOHO is a project undertaken by ESA and ASA to study the Sun, from its deep core to the outer corona and the solar wind. The goals of the SOHO mission are to reach a better understanding of the structure and dynamics of the Solar interior using techniques of helioseismology, to gain better insight into the physical processes that form and heat the Sun's corona, and to investigate the solar wind and its acceleration processes. SOHO was launched in December 1995 and is still in operation.



Artist's conception of the Sakigake Solar orbiter.

Solar Probe

JPL is developing Solar Probe to operate near the Sun's surface in order to investigate solar wind and coronal energy and to make density maps. The science package for the Solar Probe is not yet finalized, but will most likely consist of passive instruments. Data collected by these instruments will be transmitted back to Earth for analysis. Technology that is being developed for use on the Solar Probe includes X2000 avionics, thermal shielding to protect the craft at up to 2100°C, solar arrays, and an integrated instrument package. The Solar Probe would approach as close as 3 solar radii from the surface to enable close-up measurements of the Sun. The mission is scheduled for 2007.

Artist's conception of the Solar Probe..

Solar Terrestrial Relations Observatory (STEREO)

NASA and JHU's APL are developing STEREO to provide a new perspective on Solar eruptions by studying Earth-Sun relationships, plasma dynamics, and weather from heliocentric orbit at 1 AU. Two spacecraft will carry clusters of telescopes. Telescopic images will be combined with data from observatories on the ground or in low Earth orbit, the buildup of magnetic energy and the lift off, and the trajectory of Earthward-bound coronal mass ejections can all be tracked in three dimensions. To provide the images for a stereo reconstruction of Solar eruptions, one spacecraft will lead Earth in its orbit and one will be lagging. The mission is scheduled for launch in 2004.

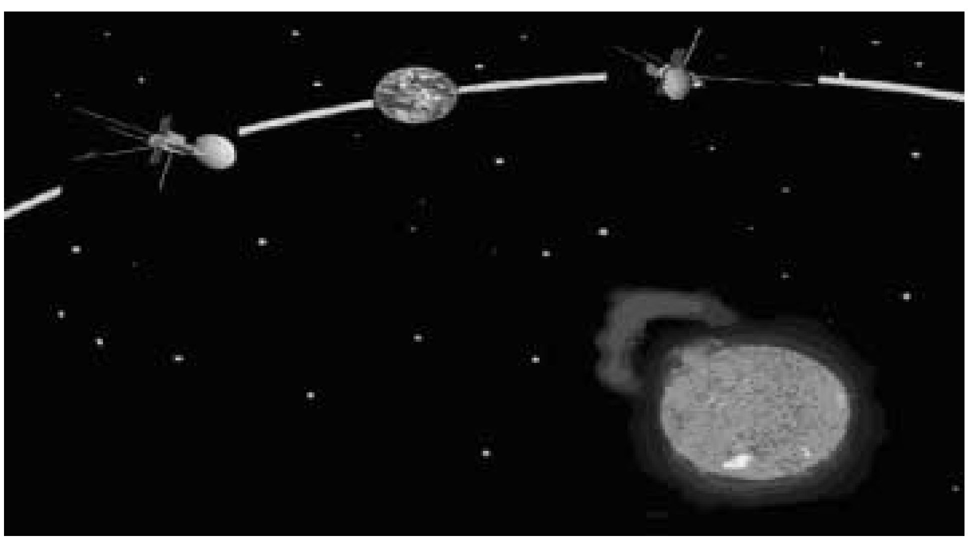
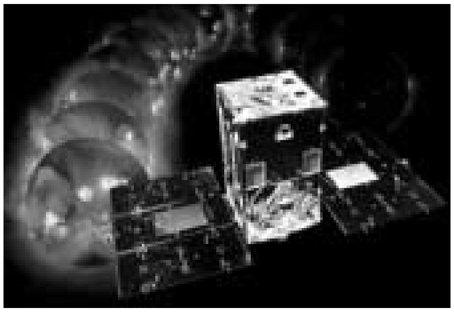


Illustration of the STEREO observatory satellites, leading and lagging Earth in its orbit.

Solar-A (Yohkoh)



Artist's rendition of Yohkoh/Solar-A.

Solar-A, also known as Yohkoh in Japanese, is a heliocentric orbiter developed by ISAS with support from NASA and BNSC. The Solar-A mission goal is to investigate the Sun's corona and study the Solar cycle using X-ray imaging. Yohkoh was launched in 1991 and still continues to produce epoch-making X-ray observations. Yohkoh has played a significant role in the development and construction of a global observation network, a mission which emphasizes cooperation among many members of the international community for ground-based and space-based scientific observations.

Solar-B

Solar-B is ISAS's follow-up mission to the Solar-A mission. The mission consists of a coordinated set of optical, extreme ultraviolet and X-ray instruments that will apply a systems approach to investigating the interaction between the Sun's magnetic field and its corona. The result will be an improved understanding of the mechanisms which give rise to Solar magnetic variability and how this variability modulates the total Solar output and creates the driving force behind space weather. This will provide for the first time quantitative measurements of the full vector magnetic field on small enough scales to resolve elemental flux tubes. Launch is scheduled for the summer of 2004.

Illustration of Solar-B's orbit around Earth as it observes the Sun.

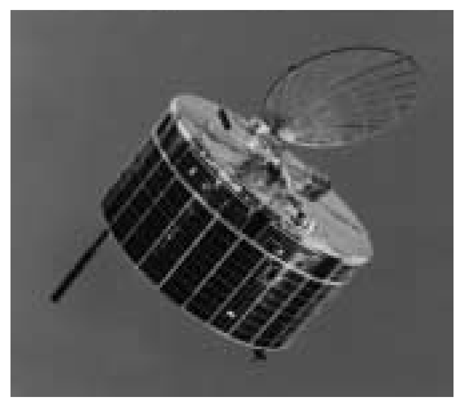
Space Interferometry Mission (SIM)

Artist's conception of the SIM optical interferometer craft.

SIM is part of the Interferometry Technology Program at JPL. SIM will be an optical interferometer operating in an Earth-trailing heliocentric orbit. SIM must meet and overcome the following technological challenges: nanometer level control and stabilization of optical element positions on a lightweight flexible structure; sub-nanometer level sensing of optical element relative position over meters of separation distance; and overall instrument complexity and the implications for interferometer integration and test and autonomous on-orbit operation. It is anticipated that some part of the Remote Agent technology developed for Deep Space 1 may be employed. The spacecraft is scheduled for launch in June 2006.

Suisei (Planet-A)

Suisei, the Japanese Planet-A, was a heliocentric satellite set in an orbit to enable a Halley's comet flyby as the comet made its closest approach to the Sun. The mission's main objective was to take UV images of the comet's hydrogen corona for about 30 days before and after it crossed the ecliptic plane. Solar wind parameters were measured for a much longer time. Developed by the ISAS, Suisei was part of the "Halley Armada" with Vega, Giotto, ICE, and Sakigake. The spacecraft was identical to that used for the Sakigake mission, and was launched in August 1985.



Artist's conception of the Solar-A Solar orbiter.

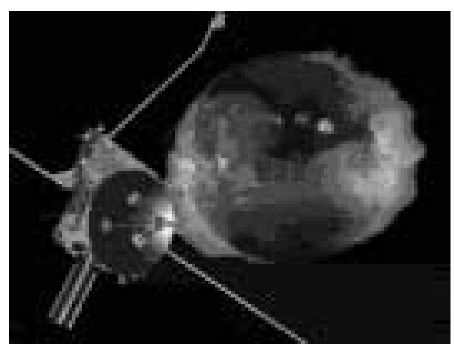
Transition Region and Coronal Explorer (TRACE)

TRACE is a heliocentric orbiter designed by GSFC and Lockheed Martin Solar and Astrophysics Laboratory for the quantitative study of the connections between fine-scale magnetic fields and the associated plasma structures on the Sun. TRACE observes the photosphere, the transition region, and the corona. With TRACE, these different temperature domains are observed nearly simultaneously with a spatial resolution of one arc-second. The orbiter was launched in April 1998 to allow joint observations with SOHO during the rising phase of the Solar cycle to sunspot maximum.

The TRACE spacecraft for Solar astrophysics.

Ulysses (International Solar Polar Mission)

Ulysses is an orbiter developed by ESA and JPL that is charting the unknown reaches of space above and below the poles of the Sun. Ulysses is studying the solar wind which carves the heliosphere. It is providing the first complete map of the heliosphere from the equator to the poles. Its scientific instruments can detect and measure solar wild ions and electrons, magnetic fields, energetic particles, cosmic rays, natural radio and plasma waves, cosmic dust, interstellar neutral gas, Solar X-rays and cosmic gamma-ray bursts. Ulysses was launched by the Space Shuttle Discovery in October 1990.



Artist's conception of Ulysses in orbit around the Sun.

Vega

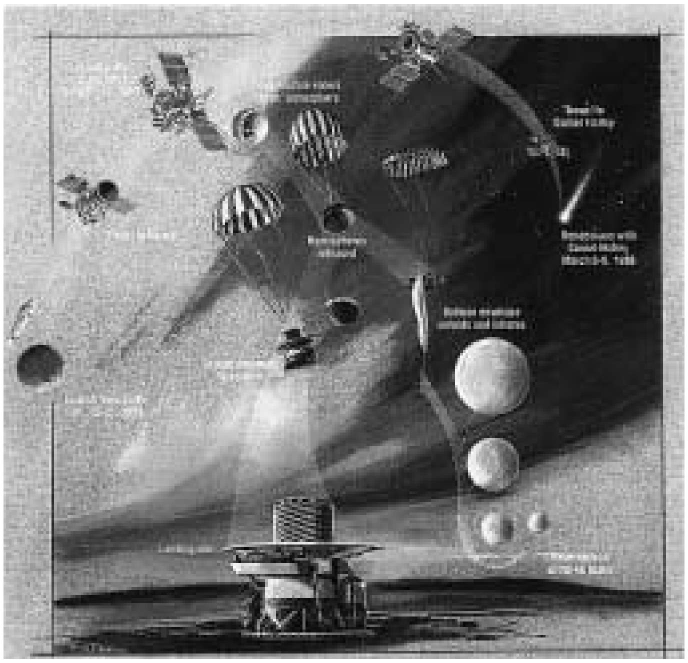


Illustration of the lander and balloon aerostat launched from Vega 2 to study Venus' atmosphere and crust.

Developed by IKI, the Vega missions were studies of Venus. Both identical Venera-class crafts combined a Venus swingby and a Halley's comet flyby. The crafts carried probes, released in the vicinity of Venus, to perform scientific experiments on the atmosphere. After releasing the probes, the spacecraft were retargeted, using a gravity field assist from Venus, to intercept Halley's comet. The spacecraft were part of the international fleet to Halley's comet that included Giotto, Sakigake and Suisei. Vega 2 additionally released a Venus lander and a balloon aerostat for further studies of the atmosphere and crust. Observations of the comet by Vega 1 and Vega 2 were used to target the Giotto spacecraft for a close encounter with the comet's nucleus. The spacecraft were launched in December 1984 and completed their missions in 1985.

Voyager

The 1997 Voyager series of two identical spacecraft, developed by JPL, performed a series of planetary flybys to perform imaging and scientific analysis, and were the first spacecraft to leave the solar system. Both Voyagers performed flybys of Jupiter and Saturn, and Voyager 2 also flew by Uranus and Neptune. Upon leaving the solar system, the Voyagers continued to investigate plasma, cosmic rays, radiation and magnetic field, relaying data to Earth. Participants in the project included GSFC, MIT, APL, and the University of Iowa.

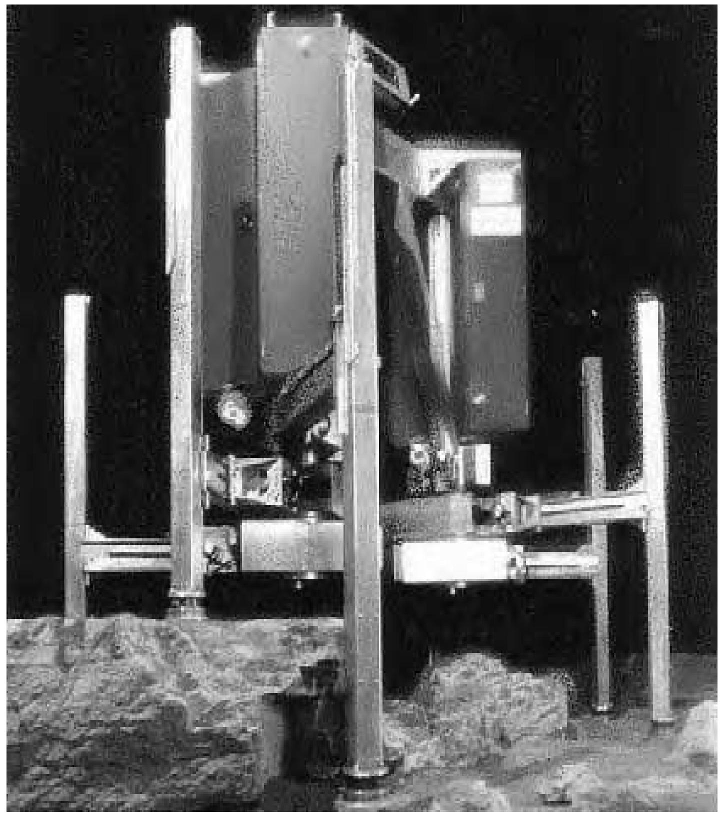
The Voyager 1 spacecraft, sent to study Jupiter and Saturn.

4.2 Planetary

4.2.1 Autonomous

Ambler

Ambler is a large autonomous rover prototype developed by Carnegie Mellon University for the NASA Space Telerobotics Program. It was designed for walking under the particular constraints of planetary terrain where there are meter-sized boulders, deep crevices, and steep slopes. An autonomous gait planner determines the foot placement and sequence required to follow a prescribed trajectory. The gait planner takes into account terrain constraints, its own walking capabilities, the reach and extension of the robot's legs, how far the robot's body can stray from its center of gravity, where the robot can move each leg without colliding into another leg, and how it can place its legs so that its body has a clear path to move forward. Ambler also autonomously builds detailed maps while walking.



Photograph of the large planetary rover prototype, Ambler, in a simulated Martian environment.

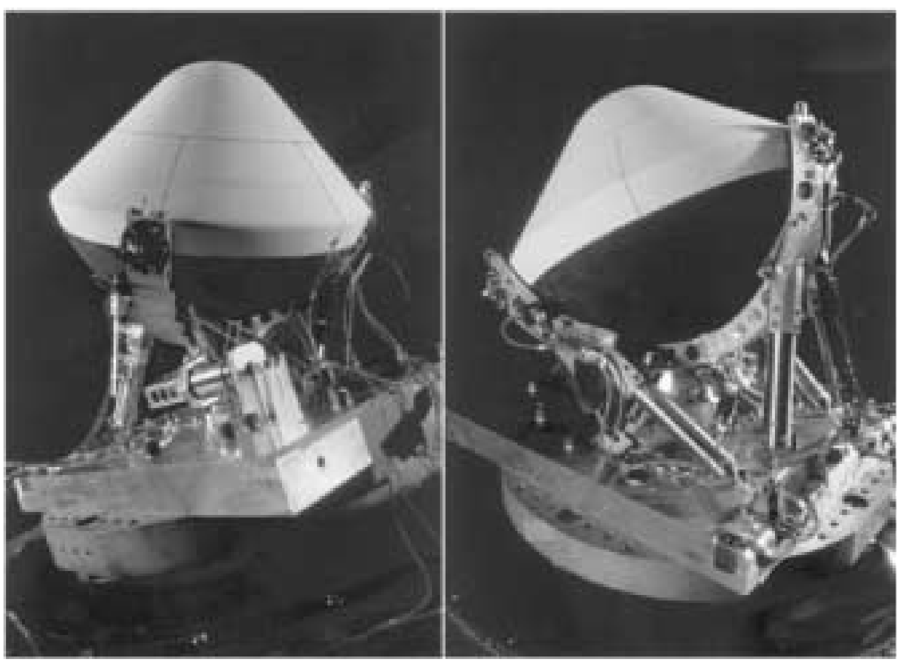
Stepping with any leg in any sequence, the six-legged Ambler has the capability to move its rear-most leg past all other legs in order to travel efficiently over extreme terrain. In extensive tests from 1987 through 1990, the Ambler traveled thousands of meters, took thousands of steps and negotiated terrain other robots even today could not.

Deep Space 2

Artist's conception of Deep Space 2 just before impacting Mars.

The Deep Space 2 probes are the first completely integrated and miniaturized systems built by JPL for space flight. The primary contractor for building the probes was Lockheed Martin, and the science team is represented additionally by NASA ARC, University of Northern Arizona, and the University of Arizona. Both microprobes were designed to test technologies to prove they work in space. The microprobes were designed to perform simple autonomous science experiments, including in-situ subsurface sample collection and analysis for water content. To slow their descent through an atmosphere and land safely, typical spacecraft need parachutes and rockets as well as an aeroshell. The Deep Space 2 probes are the first to use only an aeroshell, a system designed to shatter on impact and deploy the probe. This system not only makes the landing and penetration entirely passive, requiring no human intervention, but also makes the probes lighter and less expensive. They can survive a high-speed impact and operate successfully in extremely low temperatures. Upon impact and release from the Aeroshell, the force of collision will force the forebody of the probe into the Martian soil several centimeters. Once underground, the probe can begin autonomous sample collection and analysis.

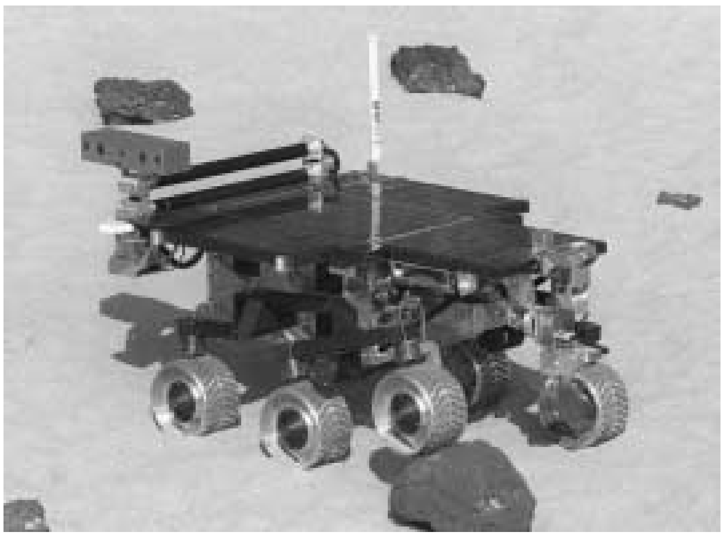
These Mars penetrators were launched in 1999 from the Mars Polar Lander during descent for water detection. Currently, Deep Space 2 is lost.



Photographs of the Deep Space 2 probes in the aeroshell.

Field Integrated Design and Operations (FIDO)

The FIDO rover is the first planned technology product of JPL's Long Range Science Rover program, an offshoot of the Exploration Technology Rover task. The Long Range Science Rover project designs, integrates and carries into simulated field operations advanced concept terrestrial rovers supporting NASA's Mars Surveyor Program. It draws on technology innovations in the areas of autonomous vehicle navigation, planetary mobility mechanization, robotic manipulation, 3D machine perception, miniature science instrumentation, and interactive ground user interfaces. Rover system issues and their experimental analysis include speed and reliability of autonomous terrain traverse, remote visual designation and acquisition of science targets, selective analysis of these targets once acquired with minimal remote interaction, control of robotic arms and coring devices, and caching.



Photograph of FIDO rover prototype in the JPL Mars yard.

Immediate engineering contributions of the ET Rover task to the ongoing Mars Surveyor Program mission development include delivery of the FIDO rover mobility platform for use in flight rover software development and validation. Later developments of the task include field trials of enriched instrumentation and broader engineering scope (longer range, higher speed, longer sequences of autonomous operations, and greater on-board intelligence), and integration and demonstration of sample return functions supporting the Mars Sample Return objectives. The first fully instrumented FIDO prototype has been completed and full-scale field trials are planned for the Spring of 2000.

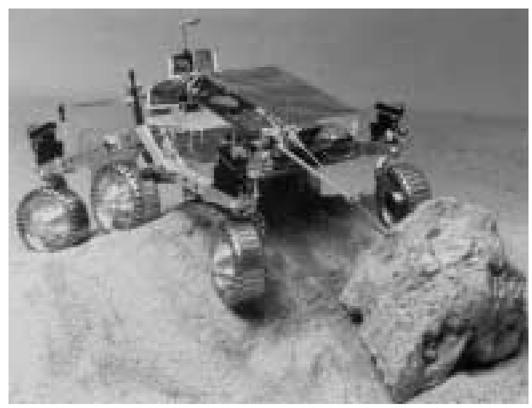
Long Range Science Rover (Rocky)

Rocky 3 with arm deployed.

The JPL Long Range Science Rover task is designed to expand the current science enabling capabilities of small robotic rovers (microrovers) for planetary surface exploration while increasing their range of operation.

The first successful models in the Rocky series of science rovers were Rocky 3 in 1990 and Rocky 4 in 1992. Rocky 3 used a laser light stripe for obstacle detection and used behavioral control. It performed autonomous rock finding and served as a platform for rover-mounted drilling experiments. It could perform visual localization by using a colored cylindrical marker (intended for identifying a lander). Rocky 4 was the prototype for the Sojourner rover used on the Mars Pathfinder mission, and used many of the same technologies demonstrated by Rocky 3. Rocky 3.2 followed in 1997.

The current generation designed by JPL is Rocky 7, a long-range planetary science rover testbed. Operational since 1997, Rocky 7's work includes exploring new or improved methods of mobility, manipulation, sensing, computation and control. Other research in rover technology consists of sensing, perception and control for navigation and manipulation. The task also will research the integration and operation of real science instruments, autonomous sequencing for nominal and contingency operations, and data reduction. Experiments in planning are being conducted by integrated the ASPEN planner. Finally, the operator interfacing portion of the task will research graphical programming of rover operations, fused display of collected science data, and World Wide Web access.



Rocky 4 testing science instrument deployment on a Mars Yard rock.

Rocky 7 employs Sojourner flight spare wheels and steering struts and is approximately the same size, but all other mechanical designs are new. The project is demonstrating, on Earth, new technology concepts for use in a long-range traversal across Mars. Current flight designs, scheduled for missions early this decade, are based on FIDO.

Rocky 7 with arm deployed, testing in the Mars Yard.

Robotic Antarctic Meteorite Search/Nomad

NASA and CMU's Robotic Antarctic Meteorite Search project develops robots for autonomous search of Antarctic meteorites and demonstrates advanced perception, control, navigation and scientific search technologies as a terrestrial analog to robotic exploration of Mars and the Moon. The project designed Nomad, a four-wheeled robot that can traverse planetary analogous terrain, as a planetary rover analog. Nomad has the capability to perform autonomous coverage of an area while performing autonomous remote scientific exploration and classification.

Photograph Nomad performing autonomous scientific analysis of a rock in Antarctica.

Nomad uses a transforming chassis, internal body averaging and in-wheel propulsion to provide greater mobility, stability and control. In 1997, Nomad navigated over 200 kilometers of the Atacama Desert in Chile to prove that its design is capable of traversing terrain similar to that of Mars and the Moon. During the trek, researchers in North America were able to perform science remotely with the use of Nomad's sensors. Nomad performed over 50 kilometers of autonomous patterned searches and its sensors allowed discovery of in situ meteorites. During the 1999-2000 field season in Antarctica, Nomad autonomously explored portions of a blue-ice field and identified five in-situ meteorites which were independently verified as meteorites by an expert.

4.2.2 Semi-Autonomous

Aerobot

Artist's conception of an Aerobot operating in the Martian skies.

The Planetary Aerobot Program is part of JPL. Planetary Aerobots are semi-autonomous, unmanned, scientific exploration vehicles designed to float like balloons for up to several months in the atmospheres of planets and are equipped to conduct sophisticated observational programs from their unique vantage points. To operate successfully on distant planets, a planetary Aerobot cannot rely on constant guidance from Earth. It must do some or all of the following autonomously: determine its position, altitude and velocity; acquire scientific data; actively control its altitude; and land at designated surface sites.

Planetary Aerobots can study the environments of other worlds from higher vantage points than autonomous wheeled vehicles and may be able to move up and down within an atmosphere to help control their flight paths. Planetary Aerobots can change their altitude and ride the winds in the atmosphere studying both meteorology and atmospheric chemistry. Also, they can explore the surface much closer to the ground than orbiting satellites and cover much more territory than planetary rovers. Aerobots were first tested on Earth in 1993. Testing is continuing and the Mars Aerobot Technology Experiment (MABTEX) and/or the Mars Solar Montgolfiere are projected to be launched in 2003.

Cassini

Photograph of the Cassini spacecraft prior to launch.

Cassini is a joint mission involving NASA's Kennedy Space Center, JPL, the European Space Agency, and others. Cassini's principal objective is to send a suite of instruments to Saturn to collect scientific data about Saturn, its rings, its satellites, its field and particle environments and its interactions between them. The Saturn and Titan probe, developed in JPL, has been designed to have a high degree of autonomy. It provides general services for each of twelve science investigations including forwarding commands to the instrument, collecting and transmitting instrument telemetry, orienting the spacecraft to desired targets and providing attitude stability, power, and thermal control. The spacecraft is flown with sufficient margins to allow the instruments to operate fairly independently from each other, but still allow for collaborative, synergistic collection of data. The spacecraft has an onboard data system that features instruments with computers capable of instrument control and data handling. Spacecraft sequences use a combination of centralized commands and instrument commands. Scientists are aided by Ames Research Center's COSMO, a heuristical planner with a graphical interface that takes mission and engineering constraints into account for the user.

Artist's conception of Cassini in Saturnian orbit

Ground operations are centralized at JPL and during the Saturn tour, JPL intended to incorporate science operations being conducted all over the world into command sequences. Scientists from around the world could operate instruments from their home institutions easily and with minimal interaction to conduct their observations. Since most of the Cassini instruments are body-fixed, Cassini observes the Saturnian system for about 12-15 hours a day and once a day points its high gain antenna to Earth for 9-12 hours and transmits the science data collected while continuing to gather fields, particles and waves data. Operational modes have been designed to balance the science return with the need to keep operational complexity and cost under control in planning sequences. Cassini was launched late in 1997. The mission requires a nearly seven year cruise to get the spacecraft to Saturn and is designed for a four year tour in orbit around Saturn.

Dante

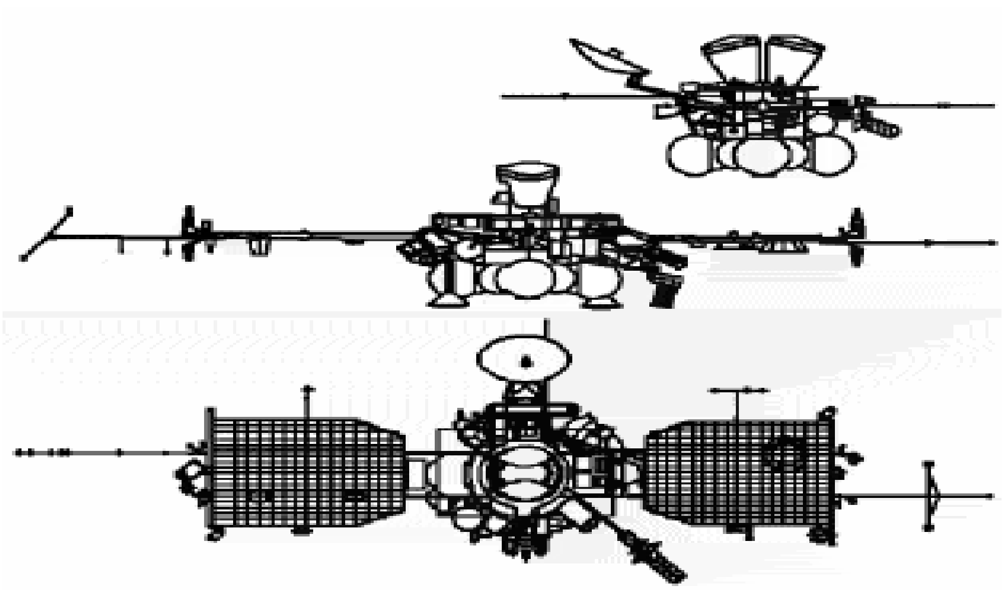
The Dante series includes two large, tethered, eight-legged walking robots. While tethered to off-board computers for data processing, the Dante systems operate semi-autonomously to achieve pre-defined scientific objectives. The gait and foot placement are autonomously determined. Both Dante robots are equipped with cameras, range-finders, and several scientific instruments. Dante is also typically operated in a teleoperation mode.

Dante II descending down the slope of Mt. Spurr, Alaska, to investigate a volcano.

Dante I was sent in 1992 to explore the Mt. Erebos volcano in Antarctica, where it demonstrated the ability to survive in both extreme heat and extreme cold conditions, but failed to demonstrate the ability to navigate rugged terrain and repel down steep cliffs. While at Erebos, it was intended to collect scientific data on one of the few known magma lakes. Dante II was designed and tested in 1994. It explored Mt. Spurr, a volcano in Alaska, enduring high temperatures and rugged slopes. It collected scientific data on the volcano, including gas samples.

Mars 96

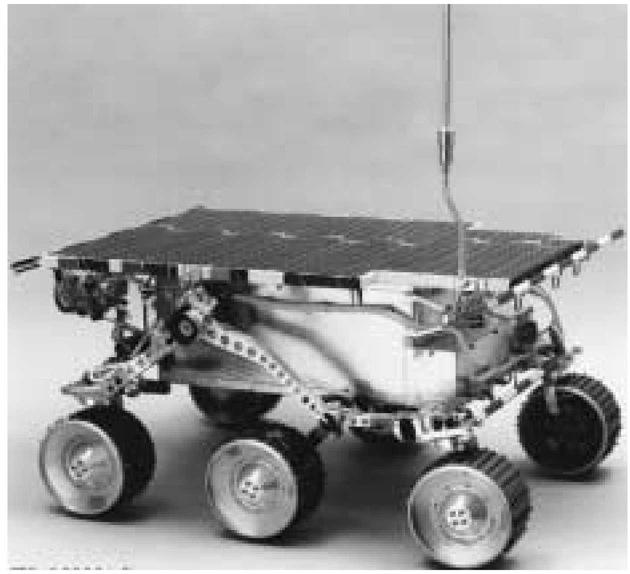
The Mars 96 mission was a group of Martian surface penetrators developed by the Russian Space Institute, IKI, to conduct autonomous science experiments. The spacecraft consisted of an orbiter equipped with several passive science instruments, two small autonomous science instrument stations that were to land on the surface of Mars, and two penetrators that would analyze the underlying surface layers. The objectives of the mission were to investigate the evolution and contemporary physics of Mars, and to perform studies of past and present physical and chemical processes. Mars 96 was to study Mars' inner structure, its atmosphere and its plasma envelope. The spacecraft was launched in November 199, but the mission failed to reach Mars.



Schematic drawing of the Mars 96 spacecraft, designed for orbital and subsurface studies of Mars.

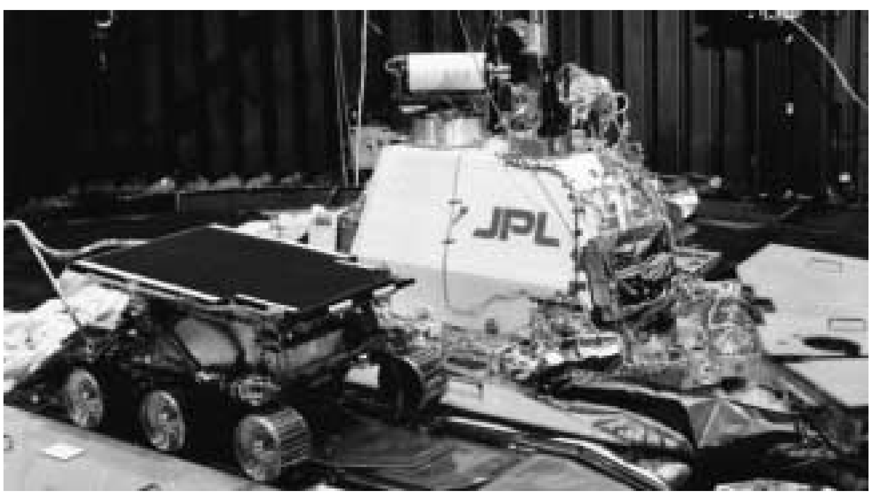
Mars Pathfinder/Sojourner

The Mars Pathfinder mission was launched in 1996, and arrived on Mars in 1997. Pathfinder included a lander and the first successful rover landing on Mars. The Sojourner rover was a small vehicle which traversed distances up to ten meters from the lander, and covered up to 100 meters during its mission. The rover was guided by waypoint teleoperated navigation via a three-dimensional reconstructed view from the rover's stereo pair of cameras. Sojourner detected obstacles with a camera and laser system and could turn to avoid them; after the obstacle was passed the rover would return to the desired heading, toward the next waypoint.



The Sojourner Mars rover.

The primary objective of the mission was to perform in-situ scientific analysis of geological samples. Several spectrometers and images were used to allow scientists on Earth to analyze data that was collected through teleoperation of the science instruments.



The Mars Pathfinder lander and Sojourner rover in landing configuration.

Marsokhod

The Marsokhod (or Mars rover) is a six-wheel-drive, Russian-built robotic planetary rover prototype. It was developed by IKI with aid from NASA. The rover has integrated advanced sensors and computer intelligence with the chassis to make it capable of handling more complex missions. Research has concentrated on a variety of architectures for intelligent mechanisms, including software architectures, advanced processors, sensor processing (including vision, tactile, and proximity sensors), user interfaces, and machine learning. The expertise is in the end-to-end development of complete systems tested under field conditions. Marsokhod is also equipped with a robot arm.

Marsokhod (right) and Koala (left) during 1996 field tests in the Arizona desert.

Several field experiments and tests with the Marsokhod vehicle have been performed since 1993. An early experiment successfully made the link between a virtual environment control system and the rover control system, reconfigured the rover control software, and began to operate the rover by sending commands via the virtual reality-based graphical user interface. Tests in the desert of Arizona in 1996, in which teams participated in the real-time operations, tested the use of the vehicle to do geologic science and have been instrumental in the design of the operator interface and in the selection of on-board science instruments; a second robot, Koala of the Swiss Robotics Institute (EPFL) was independently tested along side Marsokhod. Other tests have demonstrated a planetary exploration capability that can be used for Mars or Lunar missions and tested new control software, vision-based navigation techniques and a new arm end-effector carousel. The rover will continue to be used to test a variety of new sensors and operational capabilities.

Mars Surveyor 2003/2005

The Mars 2003 and 2005 missions will include a lander and a rover. The rover, Athena, is being developed by NASA and Cornell University and is teleoperated with autonomous obstacle avoidance. It is the first space project designed to return rock samples from another planet.

Artist's conception of the Mars Surveyor lander and rover on the Martian surface.

Following the Mars Surveyor 2001, in 2003 and 2005, Athena will continue where it left off, taking the same instruments and more on board a much more capable rover to observe Martian rocks and soil at two different landing sites. Each of these missions will be investigating geology and chemistry with scripted control of the science instruments. Currently, the navigation and science technologies are being tested by on the FIDO terrestrial prototype at JPL. Higher levels of autonomous navigation are under investigation on FIDO and may be incorporated into the Athena architecture for future missions.

The FIDO testbed for the Mars Surveyor 2003 and 2005 missions.

Selenological and Engineering Explorer (Selene)

Selene is a Japanese Moon orbiter mission jointly prepared by ISAS and NASDA. Selene will demonstrate a soft landing technology that autonomously avoids obstacles and locate flat terrain to land near a pre-selected site on the surface of the Moon. Selene will communicate with Earth through a satellite that it will carry and deposit in Lunar orbit. The satellite will additionally map the Lunar surface from orbit. Additional technology includes a propulsion module that aims to demonstrate the thermal-control and energy-storage technology necessary to survive on the Moon. The mission's objectives are to obtain scientific data on Lunar origins and evolution, and to develop technology for future Lunar exploration. It will be launched in 2003 or 2004.

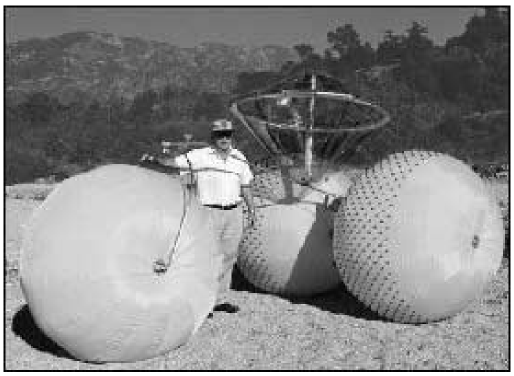
Artist's conception of the Japanese Selene at the Moon.

Artist's conception of the Selene lander on the surface at the Moon, with the communications and science satellite overhead.

4.2.3 Teleoperated

Inflatable Rover

NASA's JPL Inflatable Rover Program focuses on developing a large-wheeled, light-weight, remote-controlled vehicle for transport of instrument payloads on distant planets and moons. The rover uses large, inflatable wheels to climb over rocks instead of traveling around them so that it can autonomously traverse more rugged terrain than other planetary surface explorers. This technology enables robotic outpost development, transportation of astronauts, and long distance transfer of heavy equipment or in situ resources. New technologies that are being developed include more rugged, ultra-lightweight inflatable tires; a compressible chassis to fit into small planetary entry capsules; and revised autonomous control algorithms that allow much larger distances to be traversed.

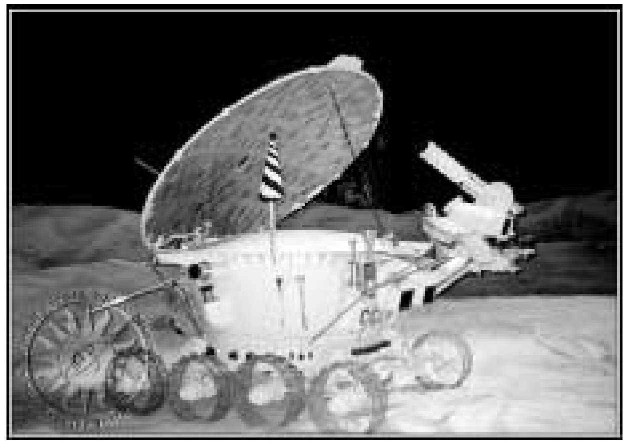


One model of an inflatable rover, JPL's design for traversing rugged terrain.

The rover has been successfully tested in a wide range of conditions including on giant sand dunes in the Mojave desert; in very rugged, rocky canyons simulating Martian terrain; and on calm lakes simulating liquid methane seas anticipated to exist on Saturn's moon. The continuing project began in 1996.

Lunokhod

Two Soviet rovers, the Lunokhods, landed on the Moon in November 1970 and January 1973 to perform imaging and science. The rovers, designed by IKI, were teleoperated roving vehicles that carried television cameras and instruments to measure the physical and chemical properties of the Lunar soil. They were part of the Soviet Luna probe series, Luna 17 and Luna 21) which began in the 1950s. Lunas were the first man-made objects to attain escape velocity; to impact on the Moon; to photograph the far side of the Moon; to soft land on the Moon; to retrieve and return Lunar surface samples to the Earth; and to deploy a Lunar rover on the Moon's surface.



A Lunokhod vehicle in a simulated Lunar environment.

Sketch of the Luna 17 lander, which carried a Lunokhod rover to the Moon.

Mars 2/Mars 3/PROP-M

the Mars 2 and Mars 3 missions were developed by IKI. Each consisted of an identical craft, which included a directly controlled orbiter, a teleoperated lander, and a teleoperated rover. The orbiters assisted in landing with autonomous altitude control. The purpose of the orbiters was to take images of the surface, study topography, and analyze the upper atmosphere of Mars. the landers were intended to perform a soft landing and perform scientific analyses of the atmosphere, weather, and soil properties. Additionally each lander would deploy a PROP-M rover with a manipulator arm. The rovers walked on skis and were tethered to the lander for power and communications. The rovers were intended to assist by taking measurements in the landers' images. Mars 2 failed to land successfully on Mars. Mars 3 did conduct the first soft landing on Mars, but subsequently lost communication with Earth.

A photograph of the Mars 3 lander.

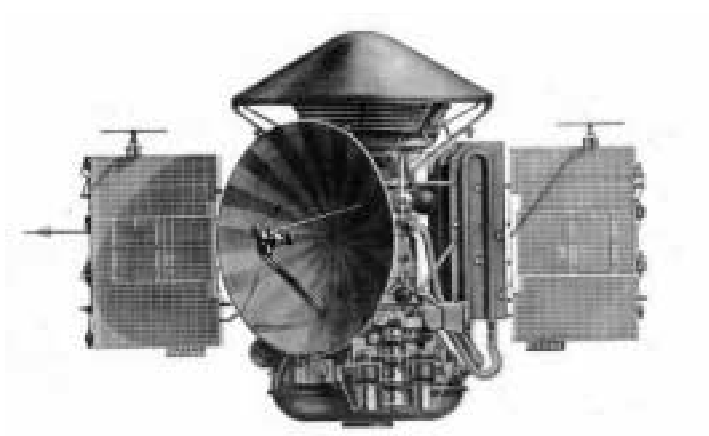


Illustration of the Mars 2 and Mars 3 orbiters.

Mars Express (ASPERA-3)

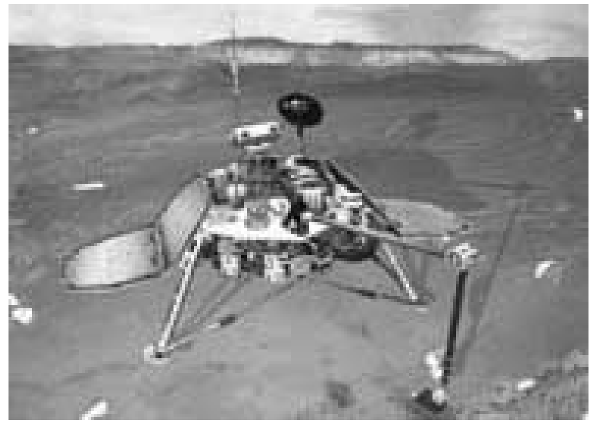
Artist's conception of the ASPERA-3 spacecraft in orbit over Mars.

Mars Express is an orbiter and a Beagle 2 lander being developed by the European Space Agency (ESA) with support from JPL and Russia. The orbiter has seven onboard scientific instruments that will probe the planet's atmosphere, structure and geology looking for, among other things, evidence of hidden water. The main spacecraft will release a small lander that will be guided via teleoperation to gather and test rock and soil samples on the surface. The other instruments will make observations from the main spacecraft in polar orbit which will allow it to gradually cover the whole planet during the primary mission's planned life of two years. Scientific data collected passively by the orbiter and through teleoperation of the lander will be relayed back to Earth for analysis. Additionally, the lander and orbiter will be capable of relaying communications from other spacecraft to Earth. The eleven day launch window opens on June 1, 2003. Mars Express is expected to arrive at Mars in 2003.

The Beagle 2 lander prototype, which will be delivered to Mars by ASPERA-3.

Mars Polar Lander

Mars Polar Lander is part of a series of missions in a long-term program of Mars exploration managed by JPL for NASA's Office of Space Science. The lander is designed to perform a semi-autonomous landing. To achieve the primary goal of the mission, the investigation of water and other volatiles near the soil surface of Mars, the lander is equipped with a robotic arm capable of collecting samples and transferring them to one of several science instruments for analysis. Sample collection and analysis is done by teleoperation and remote initiation of scripted functions. In addition to landing on the surface, while in orbit around Mars the Polar Lander released the Deep Space 2 probes, which were intended to impact the surface and search for water at locations distant from the landing site.



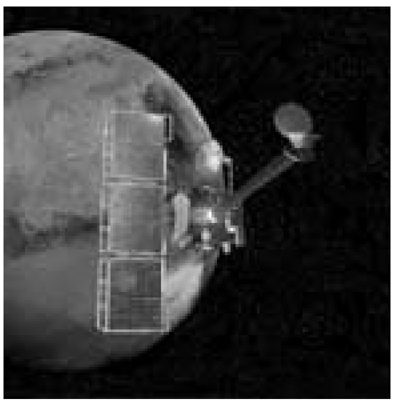
Artist's conception of the Mars Polar Lander on the Martian surface in search of water and other volatiles.

The Mars Polar Lander lost contact with Earth shortly before landing and communication was never reestablished.

Mars Surveyor 2001/Athena Precursor Experiment (APEX)

The next step in the NASA Mars Surveyor program, Surveyor 2001 is tentatively scheduled to deploy an orbiter, a lander, and a rover on Mars. the original mission concept included a rover, Marie Curie, is a small rover similar to Sojourner, designed to traverse small distances and perform teleoperated, scripted science experiments. This mission was considered the Athena Precursor Experiment (APEX) because the science package to be deployed on the Athena rover as part of future Mars Surveyor missions will be tested on Marie Curie.

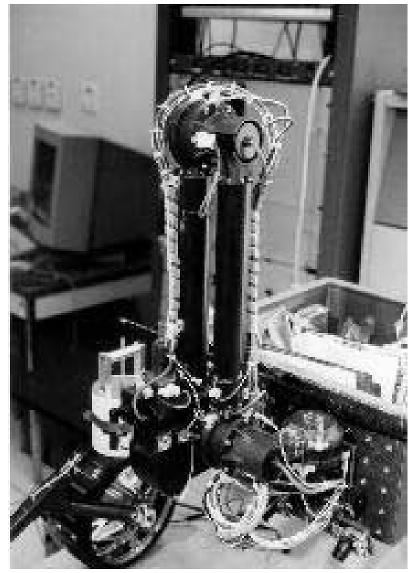
The lander and rover portions of the Mars Surveyor 2001 mission, intended as the Athena Precursor Experiment, have, as of this writing, been cancelled and only an orbiter is scheduled. The orbiter will be studying radiation as well as mineralogy and morphology of the Martian surface. It is also expected to work in conjunction with the Surveyor 2003 mission and to support communications for 2003 as well.



Artist's conception of the Mars Surveyor 2001 orbiter in orbit around Mars.

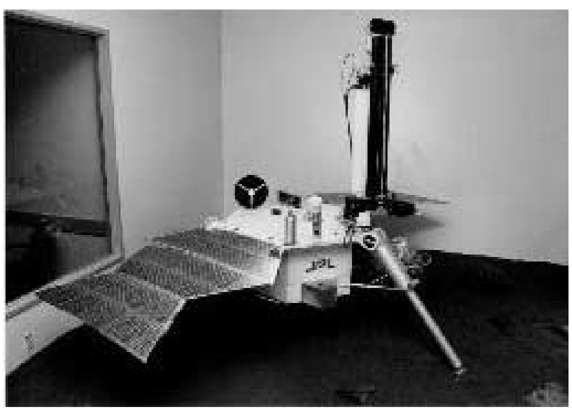
Planetary Dexterous Manipulator (PDM)

Planetary Dexterous Manipulators (PDM) develops new teleoperated and autonomous robotic system concepts and technologies for planetary surface and near-surface science. PDM is a task in the NASA Space Telerobotics Program conducted by JPL. When these dexterous manipulators are integrated with future lander platforms and roving vehicles, technology products of this task will enable scientists to dexterously view, probe, freshly expose, acquire and containerize surface and near-surface samples. A major part of this task is autonomous sample acquisition and science instrument placement software.



Fixed-base prototype of a planetary dexterous manipulator.

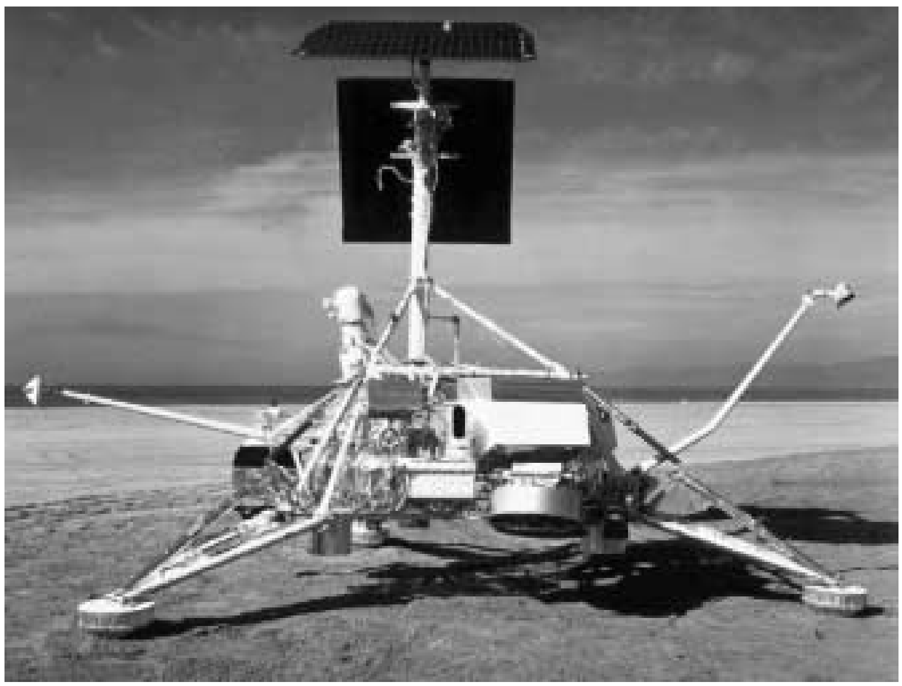
In 1998, the PDM task demonstrated autonomous sample acquisition and science instrument placement software in collaboration with the Long Range Science Rover task and delivered manipulator arms to the Exploration Technology Rover Task. The first flight model was incorporated into the Mars Polar Lander, but the mission failed. The PDM task currently is developing software that will be used in operations analysis for the MVACS robot arm. Also, the PDM task is developing autonomous sample acquisition capabilities that are targeted toward the Mars Surveyor 2003 and 2005 missions.



Prototype of a Martian lander with a deployed planetary dexterous manipulator.

Surveyor

JPL followed the Ranger series of the 1960s with the Surveyor spacecraft intended to soft-land on the Moon. Seven Surveyors were launched from June 1966 through January 1968. The science instruments varied from flight to flight, but included cameras, surface samplers and soil analyzers. The Surveyors returned nearly 88,000 high resolution pictures of the Moon's surface and performed the first soil analysis. Surveyor 3 brought a new teleoperated robotic tool, the scratcher arm, into use on the Moon. Apollo 12 astronauts removed the arm and other parts of the spacecraft so scientists could study their condition after nearly four years of exposure to the space environment.



Photograph of a Surveyor landing craft similar to those sent to the Moon in the 1960's.

4.2.4 Directly Controlled

Apollo

NASA's Apollo was a series of manned and unmanned Lunar orbiting and landing missions launched from Kennedy Space Center. The series included a manned rover, that were launched from 1968 to 1972. This series included the first and only human presence on another planetary body.

Lunar rover with astronaut Jim Irwin, Apollo 15.

Lunar landing module, Apollo 9

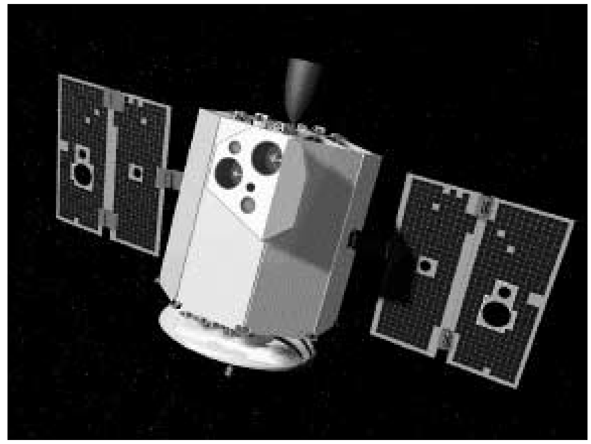
BepiColombo

BepiColombo is a Mercury orbiter to collect scientific data that is being developed as part of ESA's Horizon 2000 plan. The original mission objective is being re-examined. Currently, the possibility of performing a Mercury orbiter mission using electric propulsion as main system and appropriate trajectories is being evaluated. The scientific mission would include imaging, ion and electron analysis, electric and magnetic field analysis, and x-ray and gamma radiation. A possible launch date is set for 2009.

Artist's conception of BepiColombo in orbit around Mercury.

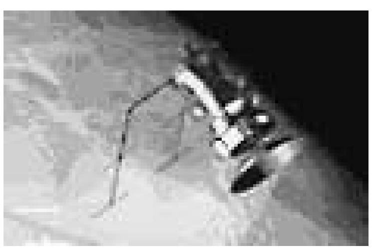
Clementine (Deep Space Probe Science Experiment/DSPSE

Clementine was a NASA-coordinated project with participation from the Naval Research Laboratory, Goddard Space Flight Center, JPL, the Ballistic Missile Defense Organization and Lawrence Livermore National Laboratory. Its goal was to test sensors and spacecraft components under extended exposure to the space environment and to make scientific observations of the Moon and the near-Earth asteroid 1620 Geographos. The orbiter performed radar mapping of most of the Moon and observations including imaging at various wavelengths such as ultraviolet and infrared, laser ranging altimetry and charged particle measurements. Geographos observations were not made due to a malfunction in the spacecraft. Clementine was launched in January 1994.



Computer image of the Clementine satellite.

Europa Orbiter

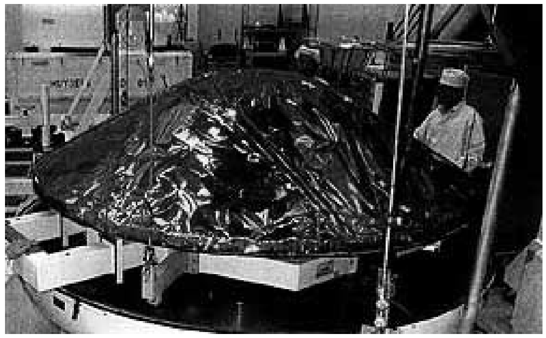


Artist's conception of Europa Orbiter.

The Europa Orbiter is being developed by JPL. It is an orbital explorer that will use a radar sounder to study the icy surface of Europa, Jupiter's fourth largest satellite, and attempt to determine the thickness of the ices and whether liquid water exists below the ice. Other instruments to study the surface and interior will include an imaging device with multiple filters to map the surface at a resolution of 100 meters and a laser altimeter to measure the topography and characterize the tidal response of the surface. The mission is scheduled for launch from the Space Shuttle in November 2003.

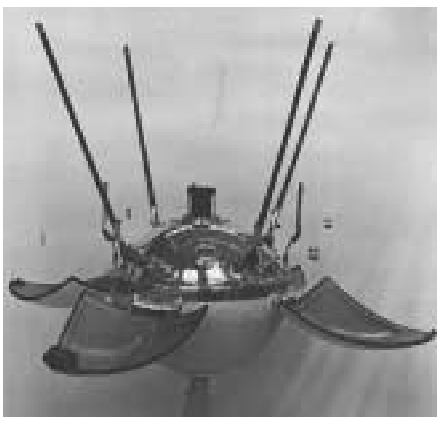
Huygens

Huygens is a joint NASA/ESA mission to explore the Saturnian system, including Saturn's atmosphere, rings and magnetosphere, and some of its moons including Saturn's biggest moon, Titan, and the icy satellites. Huygens' remote-sensing instruments will use visible, ultraviolet and infrared light and radar to record details of Titan's chemical make-up, its weather and clouds, and its surface. Huygen's was launched in October 1997 as part of NASA's Cassini mission. Cassini will launch the probe when it reaches Saturn in 2004.



The Huygens Saturn probe in preparation for launch.

Luna (Lunik)



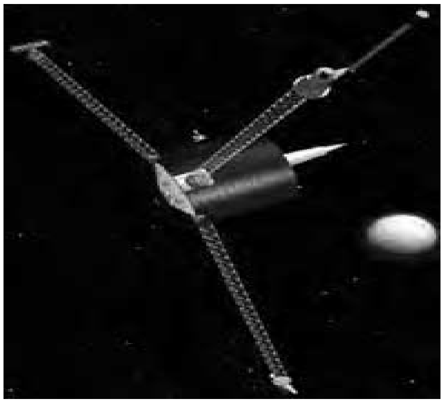
The successful Luna 9 lander of 1966.

The Soviet Lunar program had 20 successful missions to the Moon and achieved several Lunar firsts. One of the successful series of Soviet probes was Luna. Luna consisted of a series of 24 Lunar missions from 1959 through 1976. The missions included Lunar flybys, impacts, orbiters, landings, and rovers (Lunokhod 1 on Luna 19 and Lunokhod 2 on Luna 21). Luna 20 and Luna 24 included Lunar sample returns. The Lunar landers obtained close-up images of the surface of the Moon for use in Lunar studies and determination of the feasibility of manned Lunar landings. Passive scientific investigations were conducted, particularly imaging, of the Lunar surface.

Lunar Orbiter

GSFC launched five Lunar Orbiter missions from 1966 through 1967 to map the Lunar surface before the Apollo landings. All five missions were successful and 99% of the Moon was photographed with a resolution of 60 meters or better. The first three missions flew at low inclination orbits and imaged 20 potential Lunar landing sites, selected based on Earth-based observations. The fourth and fifth missions were devoted to broader scientific objectives and were flown in high-altitude polar orbits.

Lunar Prospector

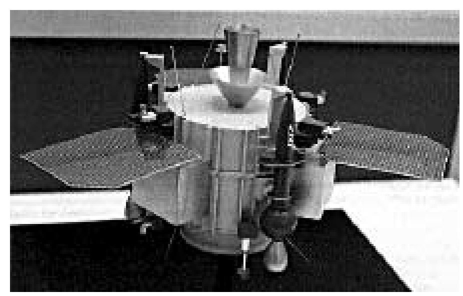


Artist's rendition of the Lunar Prospector craft.

The Lunar Prospector, part of NASA's Discovery Mission Program to further involve non-government agencies in the US space program, is a simple and reliable spin-stabilized spacecraft. Primary participants included the Lockheed Martin Missiles and Space Corporation. It was designed to perform low polar orbit investigation of the Moon, including mapping of surface composition and possible polar ice deposits, measurements of magnetic and gravity fields, and studies of Lunar outgassing events. In particular, the Lunar Prospector was looking for evidence of water ice on the Moon, and successfully identified the presence of high concentrations of hydrogen near both Lunar poles. It was launched in January 1998 and impacted the Moon in late 1999 in an unsuccessful effort to produce water vapor observable from Earth.

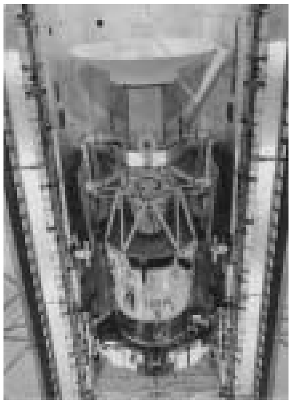
Lunar-A

ISAS is developing the Lunar-A spacecraft, which will image the surface of the Moon to monitor moonquakes, to measure the near-surface thermal properties and heat flux, and to study the Lunar core and interior structure. Lunar-A will carry a mapping camera and two surface penetrators that are equipped with seismometers and devices to measure heat flow. Launch is scheduled for 2003.



The Japanese Lunar-A spacecraft prototype.

Magellan

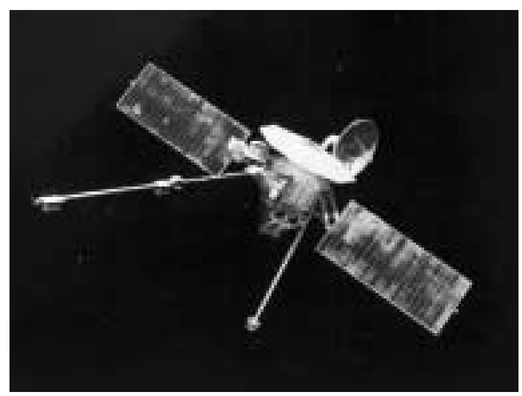


During its four years in orbit around Venus, JPL's Magellan spacecraft, built with the aid of SAIC, used a sophisticated imaging radar to make the most highly detailed maps of Venus ever captured. Magellan also made global maps of Venus' gravity field. In addition, flight controllers tested a new maneuvering technique called aerobraking, which uses a planet's atmosphere to slow or steer a spacecraft. Magellan was the first planetary spacecraft to be launched by a Space Shuttle when it was carried aloft by the Shuttle Atlantis from Kennedy Space Center in Florida in May 1989. The mission ended in late 1994.

The Magellan spacecraft in preparation for launch to Venus via the Space Shuttle.

Mariner

Mariner 2, a JPL mission, was the world's first successful interplanetary spacecraft. It was a series of Venus flybys that sent back new information about interplanetary space and the Venusian atmosphere including recording the temperature at Venus for the first time as well as measuring the density, velocity, composition and variation over time of the solar wind. It was launched in August 1962. Mariner 5 carried a complement of experiments to probe Venus' atmosphere with radio waves, scan its brightness in ultraviolet light, and sample the solar particles and magnetic fluctuations above the planet. It was launched in June 1967.

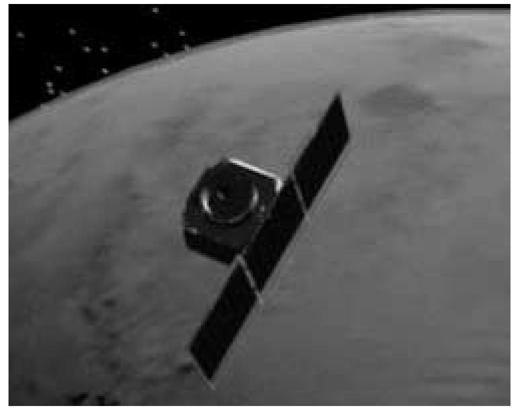


Mariner 10 spacecraft photograph.

Mariners 4, 6, 7 and 9 were part of NASA's Mariner series of flybys for imaging. The spacecraft were interplanetary probes designed to investigate Mars, Venus and Mercury. Mariner 4 gave scientists their first glimpse of Mars at close range. It carried a television camera and six other science instruments to study interplanetary space between the orbits of Earth and Mars and in the vicinity of Mars. It was launched in November 1964. Mariners 6 and 7 were designed to fly over the equator and southern hemisphere of Mars. The pair of spacecraft studied Mars' atmosphere and profiled its chemical composition. Mariner 6 was launched in February 1969 and Mariner 7 a month later. Mariner 9 was the first spacecraft to orbit another planet. It circled Mars twice each day for a full year photographing the surface and analyzing the atmosphere with infrared and ultraviolet instruments. It was launched on May 30, 1971.

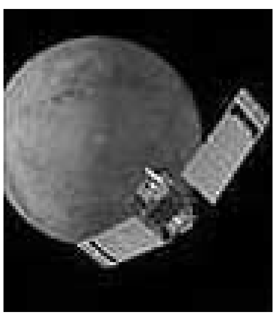
Mars Climate Orbiter

The Mars Climate Observer was developed by JPL to collect data on atmospheric composition and imaging, and act as a relay station for five years assisting in data transmission to and from the Mars Polar Lander as well as the 2001 Lander mission. The orbiter carried two instruments. The Pressure Modulator Infrared Radiometer was to provide detailed information about the atmospheric temperature on Mars, dust, water vapor and clouds. The Mars Color Imager consisted of two cameras that were to observe the Martian atmosphere and interaction between the atmosphere and the surface of the planet. The mission was launched in 1998 and ended in 1999 when the spacecraft apparently exploded.



Artist's rendering of the Mars Climate Orbiter, lost in 1999 in route to Mars.

Mars Global Surveyor (MGS)

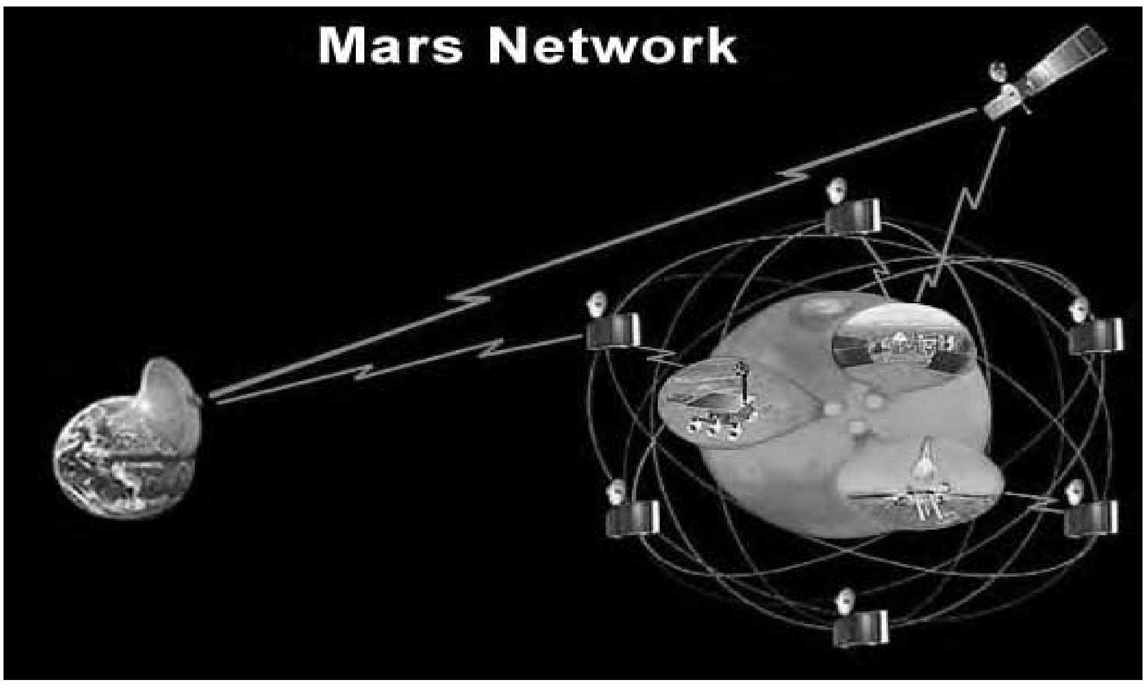


Artist's conception of Mars Global Surveyor at Mars.

The Mars Global Surveyor spacecraft was developed by JPL and Stanford University to study the atmosphere, topography, geology, mineralogy, gravity and magnetic field of Mars for a Martian year. It was intended to help future mission planners select landing sites for robotic and, ultimately, manned expeditions to the Martian surface. The MGS radio science team is using ultrastable radio transmissions from the orbiting MGS spacecraft to probe the Martian atmosphere and uncover the keys to the internal structure of that atmosphere and to the Martian climate. MGS was launched in November 1996 and the main mapping phase of the mission began in March 1999. It is still in operation.

Mars Network

Mars Network is being studied at JPL as a possible future element of NASA's Mars Surveyor Program, designed to support Mars global reconnaissance, surface exploration, sample return missions, robotic outposts, and human exploration. The network will enable these missions by developing a communications capability to provide a substantial increase in data rate, connectivity from Mars to Earth, and developing an in-situ navigation capability (similar to Earth's global positioning system) to enable more precise location information on approach and at Mars. The Mars Network constellation of Mars orbiters would enable greater information flow to the public through a constellation of microsatellites, or Microsats, and one or more Mars Aerostationary Relay Satellites, or MARSats, creating a "Mars Internet." Deployment of a prototype Microsat is tentatively scheduled for 2003.



Artist's conception of the Mars Network constellation of communications satellites in orbit around Mars.

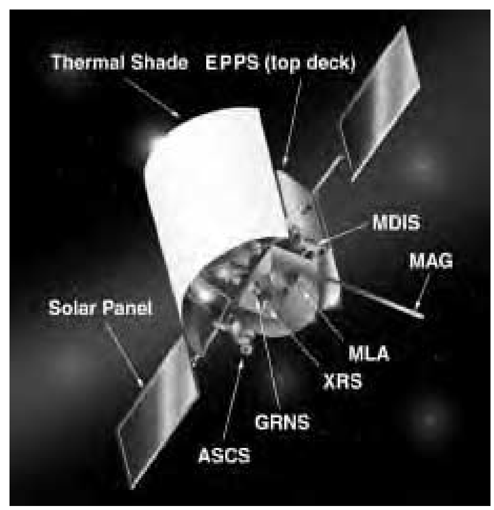
Mars Observer

The Mars Observer, developed by JPL, was the first of the Observer series of planetary missions. It was designed to study the geological science and climate of Mars. It was launched in September 1992 and was lost in August 1993.

Artist's conception of the Mars Observer in orbit around Mars.

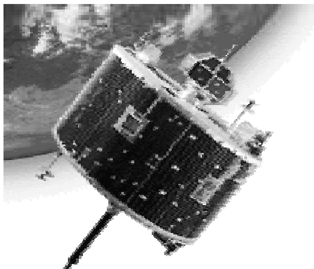
Mercury Surface, Space Environment Geochemistry and Ranging (MESSENGER)

MESSENGER is a Mercury orbiter being designed by the JHU/APL and GSFC. It will investigate the core and polar compositions, density, magnetics and geological history using an optimized set of miniaturized instruments. Data passively collected by these instruments will be transmitted to Earth for analysis. It will provide multiple flybys for global mapping, detailed study of high-priority targets and probing of the atmosphere and magnetosphere. It has an orbiter for detailed characterization of the surface, interior, atmosphere and magnetosphere. MESSENGER gets the velocity it needs from gravity assists provided by flying close to Earth, Venus and Mercury, and from chemical propulsion. It makes efficient use of its dry mass so that most of the mass launched into space is fuel. The mission is scheduled for launch in 2004 and is scheduled to run through 2009.



Drawing of the MESSENGER Mercury spacecraft with labeled components.

Muses-A (Hiten)



Artist's conceptoin of Muses-A in Earth's orbit in route to the Moon.

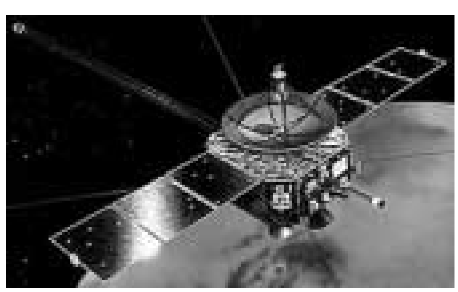
Muses-A, the Hiten, aimed to master on-orbit techniques, including swing-by by use of Lunar gravitation. When this Lunar flyby approached the Moon, the spacecraft injected a tiny Lunar orbiter, Hagoromo, into orbit around the Moon. Muses-A, an ISAS project, was launched in January 1980. The Hagoromo probe, after remaining in Lunar orbit for nearly 13 years, made impact with the Moon in April 1993.

Muses–D (Planet-C)

Muses-D (Planet C) is a Mercury orbiter in development by ISAS to investigate the iron core, surface composition and magnetic field and to produce high-resolution maps of Mercury's surface. The main propulsion will be an ion engine. It is schedule for launch in 2005.

Nozomi (Planet-B)

Nozomi, developed by ISAS, is a Mars orbiting mission designed to study the Martian upper atmosphere and its interaction with the solar wind and to develop technologies for use in future planetary missions. The spacecraft is equipped with several passive science instruments, including imagers and a mass spectrometer. This mission for imaging and atmospheric studies was launched in March 1998 and will continue in a heliocentric orbit until it encounters Mars in December 2003.



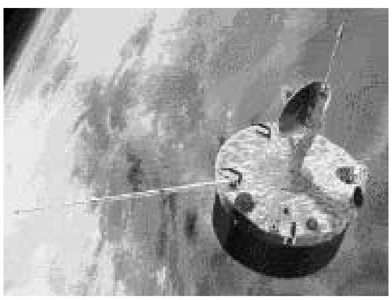
Artist's conceptoin of Nozomi in orbit around Mars.

Photograph of the Nozomi spacecraft in preparation for launch.

Phobos

Phobos 1 and Phobos 2 were IKI's next-generation in the Venera-type planetary missions, succeeding those last used during the Vega 1 and 2 missions to Halley's comet. They were to conduct studies of the interplanetary environment; perform observations of the Sun; characterize the plasma environment near Mars; conduct surface and atmospheric studies of Mars; and study the surface composition of the Martian satellite Phobos. Passive sensing techniques collected data to be relayed back to Earth. They were launched in July 1988. Phobos 1's batteries depleted before the mission was completed, and contact with Phobos 2 was lost before the end of the mission.

Pioneer Venus

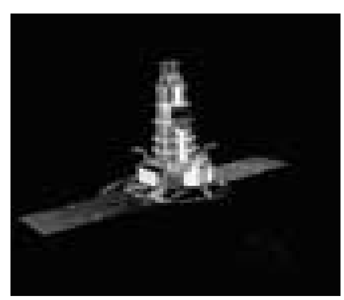


Artist's conceptoin of Nozomi in orbit around Mars.

NASA ARC's Pioneer Venus project consisted of two components, an orbiter and a multiprobe, that were launched separately. Scientific components were provided by Los Alamos National Laboratory and Sandia National Laboratory. The orbiter was launched in May 1978 and was injected into a highly elliptical orbit around Venus. The orbiter permitted global mapping of the clouds, atmosphere and ionosphere; measurement of upper atmosphere, ionosphere, and solar wind-ionosphere interaction; and mapping of the planet's surface by radar. The multiprobe, designed to conduct atmospheric experiments, was launched in November and conducted experiments during descent until impact.

Ranger

NASA's Ranger series (2-9) was the United States' first attempt to obtain close-up images of the Lunar surface and the first robotic spacecraft sent toward the Earth's Moon. The spacecraft were designed to fly straight down toward the Moon and send images back until the moment of impact. The missions ran from 1961 though 1965 and provided detailed images that were used by planners for the Apollo mission.



The Ranger spacecraft for imaging the Moon.

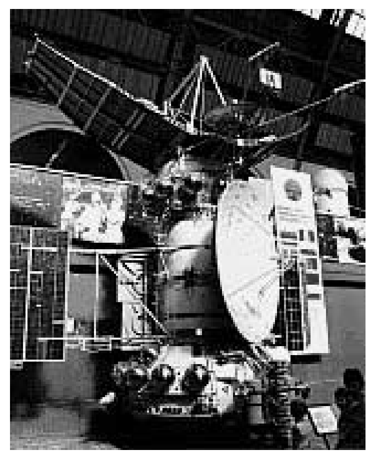
Small Missions for Advanced Research in Technology (SMART) 1

Artist's conception of the SMART-1 Lunar orbiter.

SMART-1 is a Lunar orbiter designed as part of the European Space Academy's (ESA) Horizons 2000 Science plan to test spacecraft technologies for future missions. It will demonstrate innovative and key technologies for scientific deep-space missions by first going to the Moon. The primary technology being tested is a solar-powered ion drive. SMART-1 will carry an associate technology that will monitor all aspects of the electric propulsion. The orbiter will carry an experimental deep-space telecommunications system and an instrument payload to monitor the ion drive and study the Moon. It is scheduled for launch in December 2002.

Venera

Venera was a series of Soviet designed Venus orbiters, landers, and probes for scientific studies of the atmosphere and surface. The series also included two Venus flybys, both of which failed. The Venera program ran from 1961 through 1983.



The Soviet Venus lander, Venera.

The Soviet Venus orbiter, Venera.

Zond

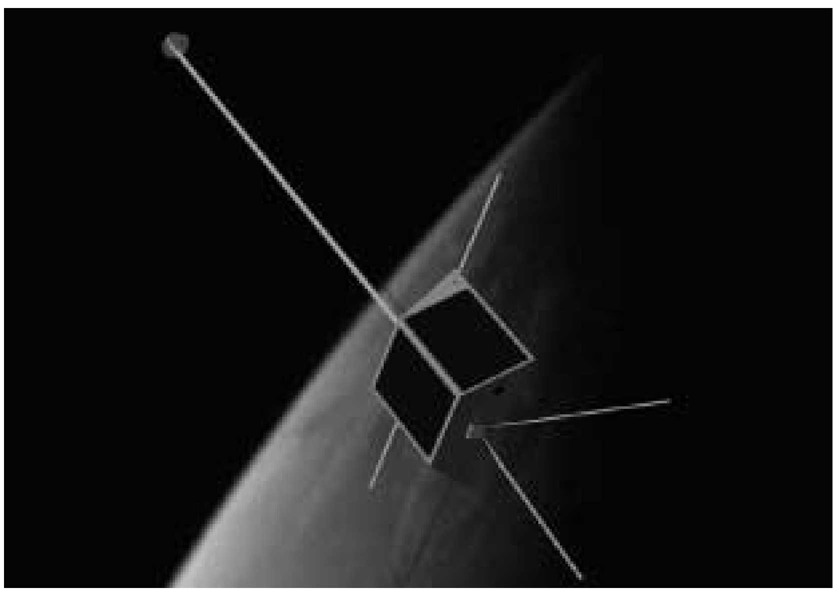
The Zond series of Lunar missions were part of the Soviet Lunar program. Together with the Luna series, the Soviet Lunar program had 20 successful missions to the Moon. Zond consisted of five Lunar missions from 1965 through 1970. This series included Lunar flybys and orbiters for imaging and passive science data collection.

4.3 Orbital

4.3.1 Semi-Autonomous

Citizen Explorer (CX-1)

CX-1 is a satellite project at the University of Colorado for JPL, supported by the Colorado Space Grant Consortium. The satellite was designed with assistance from the ASPEN planning and scheduling system to evaluate power and other engineering requirements. The ASPEN planner also is used in its ground operations to automate generation of validated command sequences. Automated planning technology aids in balancing competing mission needs by enabling mission designers to layout mission operations' plans and analyze science return in the context of different hardware configurations and mission operations' policies. The same planning system can be used during operations. It is hoped that automating the sequence generation process and encapsulating the operation specific knowledge will allow spacecraft commanding by non-operations personnel so that eventually scientists will be able to command the spacecraft directly.



Artist's conception of the Citizen Explorer 1 operating in Earth's orbit.

Operating in Earth's orbit, the Citizen Explorer project will provide: extensive geographical coverage of ozone, aerosol and ultraviolet radiation measurements; extensive coverage to study large scale, global phenomena and localized trends; and a unique opportunity for scientists and students to explore localized atmospheric trends vs. urban atmospheric variations. Many of the observations will be taken from ground stations at primary (K-12) schools. In addition, CX-1 will observe sudden stratospheric warming events and episodes of Solar activity and study the effects of these phenomena on the atmosphere and biosphere on a geographically broad scale. The CX-1 satellite was scheduled for launch on a Delta-II launch vehicle in December 1999.

Earth Observing–1 (EO-1)

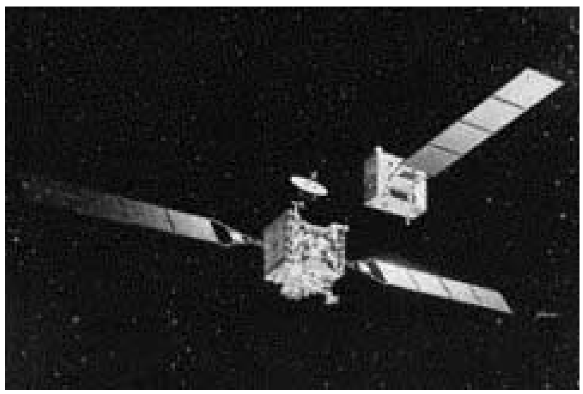
Earth Observing–1 is the first satellite in NASA's New Millenium Earth Observing series and is being developed by Goddard Space Flight Center. The EO-1 satellite for Earth science uses enhanced formation technology (EFF) for autonomous planning, execution and calibration of satellite constellation formations to fly in formation with another satellite, Landsat 7. The approach to formation flying uses fuzzy logic. Using EFF technology for onboard constellation and formation control will enable a large number of spacecraft to be managed with a minimum of ground support. The result will be a group of spacecraft with the ability to detect errors and cooperatively agree on the appropriate maneuver to maintain their desired positions and orientation. The EFF technology features flight software that is capable of autonomously planning, executing and calibrating routine spacecraft maneuvers to maintain satellites in their respective constellations and formations. It is applicable to any mission class that desires to fly multiple satellites autonomously. EO-1 will use the ASPEN planner for daily task scheduling.

The EO-1 mission will fly seven new crosscutting spacecraft technologies that will reduce the cost, mass, and complexity of future Earth observing spacecraft and allow more scientific payload to fly on future missions. The missions will develop and validate instruments and technologies for space-based Earth observations with unique spatial, spectral and temporal characteristics not previously available. Earth Observing-1 will be inserted into an orbit flying in formation with the Landsat 7 satellite, which will be taking a series of the same images so they can be compared to evaluate EO-1. Earth Observing–1 is scheduled to launch in the summer of 2000.

Illustration of the Earth Observing-1 craft, a technology demonstrator for formation flight.

Engineering Test Satellite (ETS) 7

ETS 7 is a NASDA developed satellite program that is intended to test the basic technologies of autonomous rendezvous docking and space robotics. The project consists of two satellites; one, the docking satellite, is autonomously controlled and the other, the docking station, is remotely piloted. The satellites were launched together and then separated after launching. Rendezvous docking experiments have been conducted twice; Japan's first successful fully autonomous docking occurred in 1998 when the two satellites first held a fixed-distance formation and then later docked. Small parts manipulations and propellant replenishment experiments have been conducted using remotely piloted robot arms on one of the satellites. The program began in 1997 and finished its experiments in 1999. It will continue to be operated so it can acquire long-term trend data of satellite equipment.



Artist's conception of the Engineering Test Satellites performing and autonomous docking maneuver.

Experimental Servicing Satellite (ESS)

The goal of DLR's ESS space project is to build a satellite that is equipped with a robot manipulator in order to repair failed satellites by human teleoperation from the ground. One of the key features in development is the autonomous capturing of the target satellite by machine vision. A shared autonomy will distribute intelligence between man and machine. The feasibility of graphically simulating the robot within its environment is extended by emulating different sensor functions like distance, force-torque and vision sensors to achieve a correct copy of the real system behavior as far as possible. These simulation features are embedded in a task-driven, high-level robot programming approach. To provide robots capable of acting and reacting autonomously, ESS will use TeleSensor Programming concepts. The keys to achieving local autonomy are the extensive use of sensor data processing and the ability to instruct the robot on an intuitive semantic level.

Photograph of the Experimental Servicing Satellite testbed performing a manipulation test task.

In the lab, the visual serving phase is simulated using a two robot system. One of the robots carries a satellite mockup with an original sized apogee jet and performs a typical tumbling motion of a rigid body under zero gravity. The second robot tracks the satellite by inserting a specialized capture tool into the apogee motor in order to capture it. The approach of the target satellite is controlled by real-time, model-based machine vision. Once contact between capture tool and apogee motor has been established force-torque sensing takes over. In both phases all six degrees of freedom are being controlled. The teleoperation system is used to control the whole capturing sequence. ESS was launched by ROTEX in 1994.

Extreme Ultraviolet Explorer (EUVE)

The Extreme Ultraviolet Explorer is an Earth orbiting ultraviolet spectrometer developed by ARC and the Center for Extreme Ultraviolet Astrophysics (CEA) at the University of California, Berkley. It has autonomous generation of configuration/orientation sequences to satisfy specified goals. Scientists developed special grazing incidence mirrors in which the light collecting surface does not directly face the source, but is instead positioned almost parallel to the incoming radiation. This also works with diffraction gratings that can be used at grazing-incidence angles to separates the incoming extreme ultraviolet radiation into its individual wavelengths. The grazing-incidence mirror and grating surfaces must be made exceptionally smooth because any surface irregularity will change the direction of the reflecting light rays. A detector was developed that does not require a protective cover so that it can work in this spectral region. The detectors sense the position of each incoming photon and record the exact moment it is received so that photographic quality images can be created.

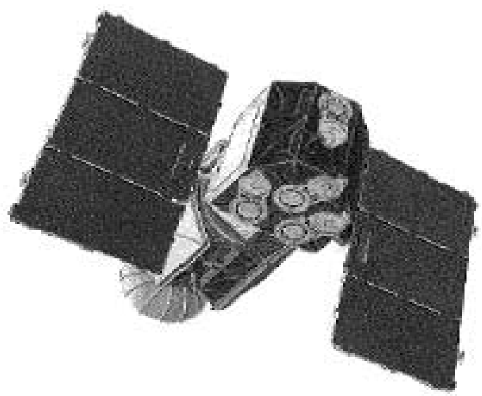


Illustration of the concept for the Extreme UV Explorer spectrometer satellite.

The technology in EUVE is necessary because components used in standard telescopes and spectrometers cannot be used for extreme ultraviolet studies. Also, ordinary visible light and ultraviolet light detectors cannot be used at these wave lengths because their protective covers absorb the extreme ultraviolet light preventing it from reaching the actual detection devices. Finally, because extreme ultraviolet studies have departed from typical astronomical investigations, standards for calibrating laboratory measurements had not been established. EUVE launched in 1992 and was placed in circular Earth orbit at an altitude of 528 km. It completes an orbit every 96 minutes.

Hubble Space Telescope (HST)

The Hubble Space Telescope, developed by NASA Headquarters, GSFC and MSFC, is a large deep-space telescope operating in Earth's orbit to avoid the atmospheric problems of Earth-based telescopes. The scheduling of the telescope is a complex problem that has been automated using a short-term and a long-term planner. Requests for observations, specified as coordinates and including time constraints, are entered by scientists and translated to the planner. Each request is given a priority by the observing committee based on time constraints and level of scientific importance. The long-term scheduler, SPIKE, assigns observations to week-long blocks of time, first assigning high priority observations and then filling in with lower priority ones. These blocks are generally overbooked, and are scheduled up to one year in advance. A complete one-year schedule, including 5000 observation requests, takes less than one hour to complete. The short term scheduler, Science Planning and Scheduling System (SPSS), assigns specific times to the observations assigned to each week and translates the schedule into telescope pointing command sequences. Unscheduled observations are fit in as time allows.

Photograph of the Hubble Space Telescope in the Space Shuttle bay for repairs.

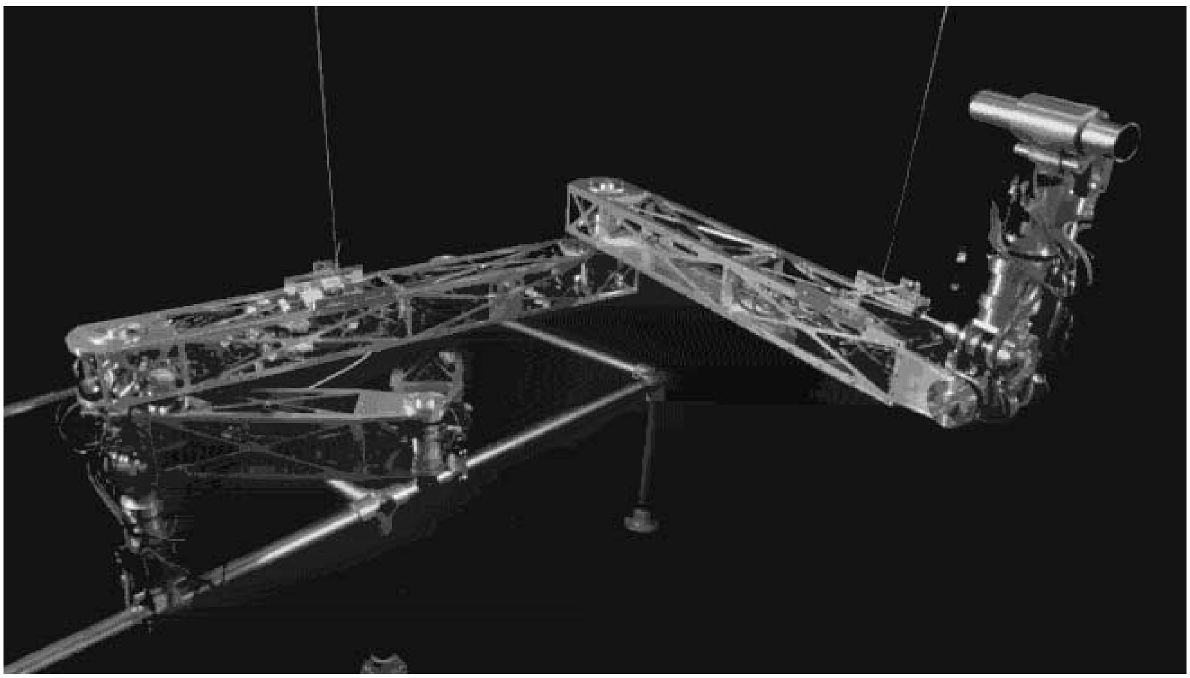
Multimode Proximity Operations Device (MPOD)

The MPOD is a free-flying mobile robot designed at the Space Systems Laboratory of the University of Maryland (SSL) to test new ways of piloting spacecraft. In particular, current research has focused on autonomous approach and docking. The system is controlled by an adaptive neural network control system, but can be controlled directly through the onboard cockpit. Acoustic sensors provide the robot with proximity information. The current testbed has six degree-of-freedom motion. It has demonstrated successfully the concept of a free-flying astronaut assistant, as well as the neural network control system's ability to autonomously dock the MPOD with a docking station. The testbed operates underwater to simulate zero-gravity with neutral buoyancy. Research into neural network control for free-flyers at SSL was first published in 1995.

Underwater photograph of MPOD testing autonomous docking capability.

Skyworker

Skyworker is an assembly, inspection and maintenance robot that autonomously transports and manipulates payloads of kilograms to tons over kilometer distances by walking. The robot is the product of NASA and CMU. Skyworker research began evolving a class of attached mobile robots as a workforce for orbital assembly, inspection and maintenance in 1999. High-level commands, such as goal locations or manipulation specification for objects, first are decomposed into a sequence of steps and then implemented as motion control commands. Skyworker's onboard networked motion controllers achieve high frequency internal control leaving the computer to perform higher level functions.



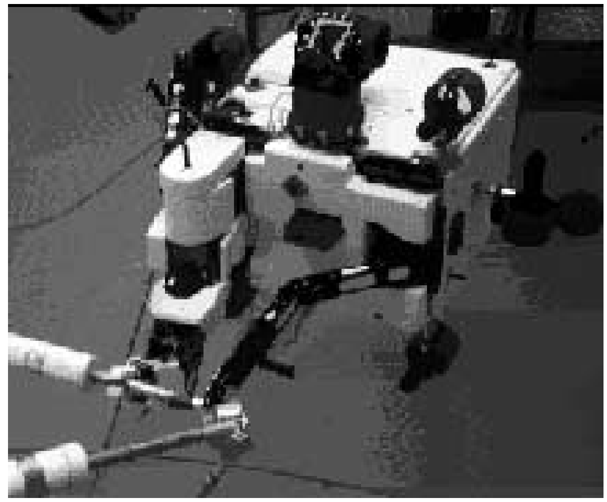
Photograph of the autonomous Skyworker prototype, supported by a gravity compensation system.

Walking robots expend less fuel than free flying robots, have a longer reach than fixed manipulators, and do not require strong attachment points like fixed manipulators so are the better alternative for large scale construction. Skyworker walks and works on the structure it is building and can move a payload at a constant velocity while walking, which reduces the forces exerted on the structure as well as increases the robot's power efficiency. The Skyworker prototype is operated under a gravity compensation system. It features rechargeable power and wireless communication.

4.3.2 Teleoperated

Beam Assembly Teleoperator (BAT)

The Beam Assembly Teleoperator is one of the first attempts at a robot for space construction applications. The BAT project was designed to be capable of a very specific assembly task in EVA: assembly of the same structure used by SSL for the Experimental Assembly of Structures in EVA (EASE) program with no modifications for robotic manipulation. Developed by the SSL, the mobile robot and manipulator was tested for space construction in an underwater neutral buoyancy environment. BAT was later implemented for testing on a 1985 US Space Shuttle mission.



Photograph of the Beam Assembly Teleoperator performing a test assembly task underwater.

Roboter Technology Experiment (ROTEX)

The ROTEX is a telerobotic arm for Shuttle EVA operation developed by DLR. The multi-sensory robot worked successfully in several control modes: teleoperated on-board by astronauts, ground-based teleoperation using predictive graphics, and sensor-based off-line programming. The variety of operational modes demonstrates the flexibility of the overall system. This space robot technology experiment used multi-sensory gripper technology, local (shared autonomy) sensory feedback control concepts, and delay-compensating graphics simulation. In the various operational modes, the robot successfully closed and opened connector plugs, assembled structures from single parts, and captured a free-floating object. In April 1993, ROTEX was flown on Space Shuttle Columbia (STS 55).

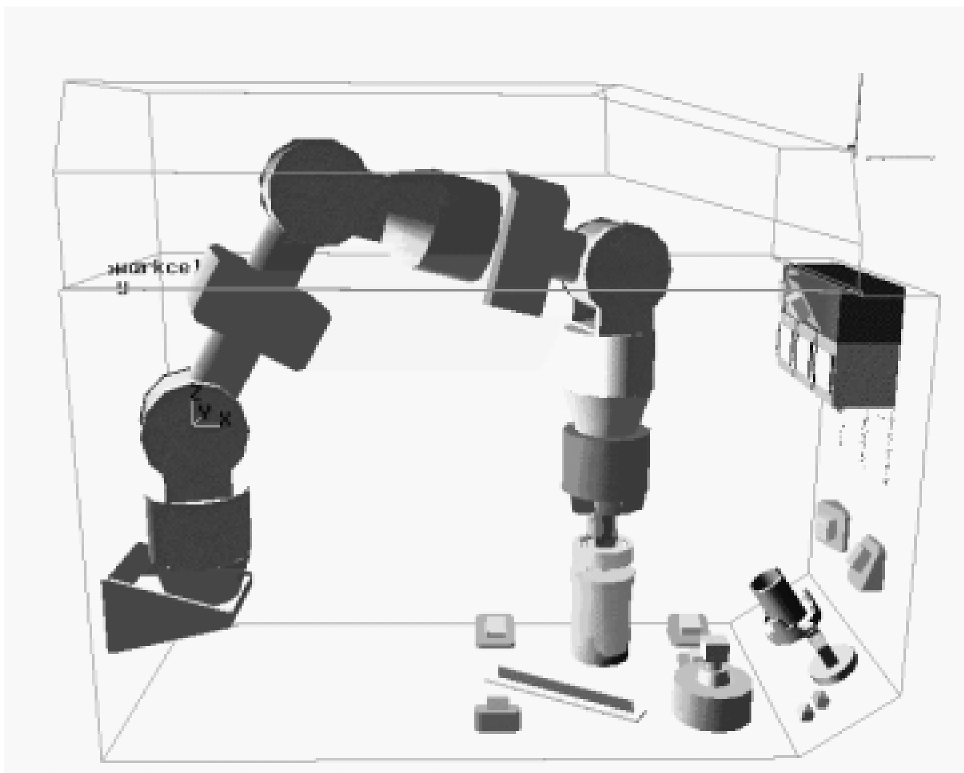


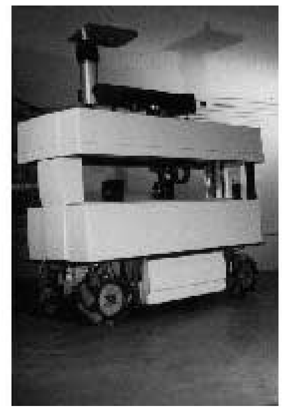
Illustration showing the graphical itnerface used to control the ROTEX arm on the Space Shuttle.

4.4 Human Assistance

4.4.1 Autonomous

Tessellator

Tessellator is an autonomous robot to inspect and waterproof the Space Shuttle's heat shield tiles. The NASA funded CMU robot autonomously characterizes tile anomalies such as cracks, scratches, gouges, discoloration and erosion by comparison with a database collected in previous tile inspections, and can ask the operator to investigate suspicious tiles which it cannot characterize. By inspecting tiles more accurately than the human eye, Tessellator reduces the need for multiple reinspections. As an additional task, Tessellator injects a toxic waterproofing chemical into each tile that prevents the light weight, silica tiles from absorbing water. Tessellator autonomously assures coverage of the Shuttle's tiles by tessellating the space into regular subregions and inspecting each subregion. The only human assistance generally required is the location of the Shuttle to be inspected.



Photograph of Tessellator.

Artist's conception of Tessellator investigating the tiles underneath the Space Shuttle.

In spring 1994, integration of mechanics and electronics of the vision system and waterproofing system occurred at CMU. Tessellator was delivered to Kennedy Space Center in June 1994. Final integration occurred in Fall 1994.

4.4.2 Semi-Autonomous

Autonomous EVA Robotic Camera (AERCam)

Image of AERCam assisting a Shuttle astronaut.

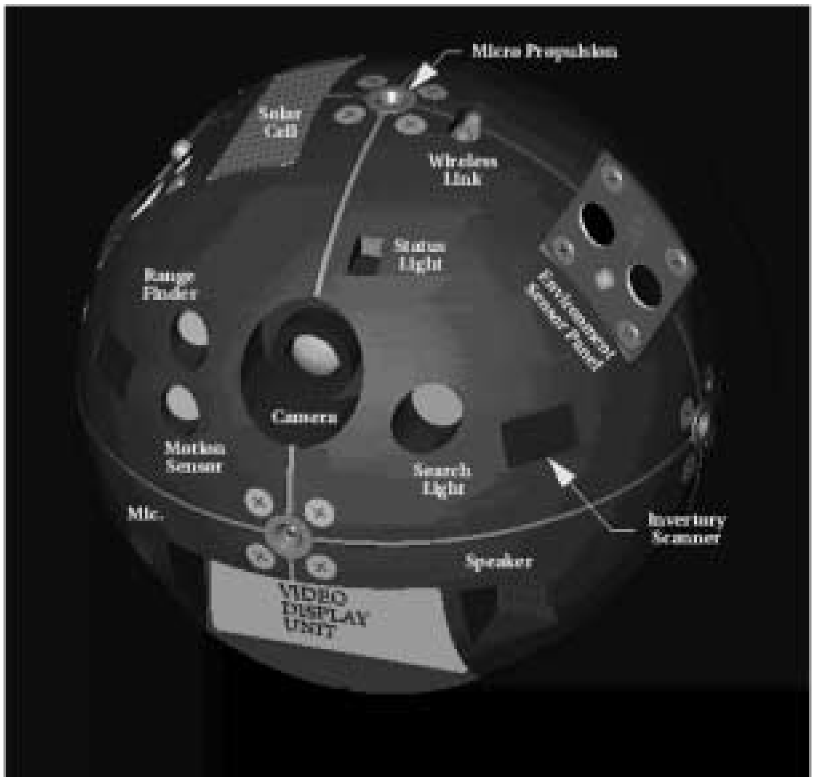
JSC's AERCam project developed Sprint as the first in a proposed series of autonomous EVA (extravehicular activity) robotic cameras. The AERCam Sprint free-flyer is a small, unobtrusive, free-flying camera platform for use outside the International Space Station (ISS) or the Space Shuttle. Current flight versions are teleoperated, but have autonomous altitude and position control; versions with autonomous navigation are in development. The cameras are being designed so they can be positioned without major impact to the actual design of ISS. Unlike the fixed cameras typically used, the mobility of AERCam provides a greater number of locations with camera coverage and a less restricted camera view. This flexibility allows IVA crews to observe extravehicular activities from various positions, and enables visual inspection of any location without an EVA crew member or remote manipulator system camera.

AERCam was developed and flight tested on a 1997 Shuttle mission to demonstrate the feasibility and capability of such a system.

Close-up image of AERCam.

Personal Satellite Assistant (PSA)

The PSA is a small, free-flying robotic assistant with autonomous navigation capability developed by NASA ARC. It is designed as support for astronauts living and working in space aboard the Space Shuttle, Space Station and during future space exploration missions to the Moon and Mars. It is an astronaut support device designed to move and operate independently in the microgravity environment of space-based vehicles. The PSA has sensors for measuring gases, temperature and air pressure. The PSA can perform video conferencing and can communicate with electronic support devices such as computer servers, avionics systems and wireless LAN bridges. A terrestrial testbed has been in operation since 1998, and has demonstrated basic worksite support capability including visual monitoring and communications support.



Computer schematic of the Personal Satellite Assistant free-flyer.

4.4.3 Teleoperated

Charlotte

Charlotte is a small spider-like robot designed by McDonnell Douglas Aerospace for Shuttle bay operations. The robot has six degrees of freedom and has a dexterous arm with an end effector. Charlotte is supported in the Shuttle bay by eight suspension cables. Wheels in contact with the suspension cables allow it to move to specified locations throughout the Shuttle bay, or to maintain position. It provides automated support for experiments and extra-vehicular activities and provides video feedback. Charlotte successfully demonstrated the ability to assist astronauts by changing experiment samples in the Shuttle bay in a Shuttle mission in 1995.

Image of Charlotte in use by a Space Shuttle astronaut.

Multisensory Articulated Hand and Robotnaut

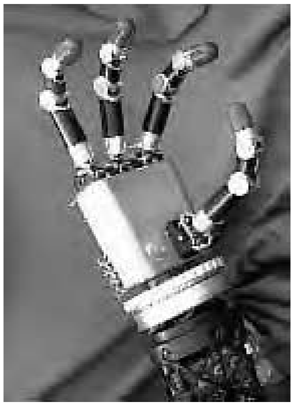


Image the DLR hand.

The DLR Articulated Hand is an articulated hand prototype with force and torque sensing for humanoid space robots. The hand is controlled by a human teleoperator; the operator's motions are transformed into motor control commands to mimic gestures. The DLR Articulated Hand is a multi-sensory four-finger hand with a total of twelve degrees of freedom in which all actuators are integrated in the hand's palm or directly in the fingers. The anthropomorphic fingertips are crucial for grasping and manipulation so they are modular and easily exchangeable with specially adapted versions. Each finger has fine-resolution sensing, actuation, and electronic preprocessing technology. The first version was completed in April 1997 and the project is ongoing. It is intended to be integrated with DLR's anthropomorphic telerobot platform, Robotnaut, to increase the dexterity and capability of such systems.

The DLR Robotnaut is designed to be a light-weight, humanoid telerobotic platform for space and EVA applications. The current robot prototype is a fixed-base model, with multiple end effectors for different purposes. Future models will be of lighter materials and incorporate the articulated hand. No flight missions for the hand or Robotnaut have been scheduled.

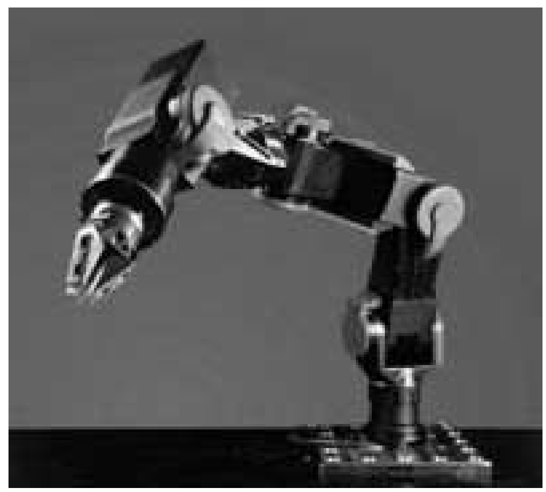


Image of the DLR Robotnaut, on which will be integrated the hand..

Ranger

Photograph of the Ranger craft.

To meet the increased demand for space operations, SSL at the University of Maryland is developing robotic assistance systems with capabilities ranging from simple teleoperation to complete autonomy. Both the Ranger Neutral Buoyancy Vehicle (NBV) and the Ranger Telerobotic Shuttle Experiment (RTSX) employ multiple manipulators to perform telerobotic servicing. The Ranger NBV is designed to demonstrate the ability of a free flying telerobotic system to perform many required operational tasks including EVA worksite preparation, on-orbit refueling, instrumentation package replacement, and deployment of failed mechanisms. Ranger combines current robotic technology with a free-flying spacecraft bus. RTSX is a four manipulator telerobot with one manipulator permanently attached to a SpaceLab pallet. The manipulators perform dexterous manipulation, body repositioning and stereo video viewing.

The Ranger project began in 1990. Ranger will provide telerobotic servicing on International Space Station Orbit Replaceable Units (ORUs) and EVA equipment in the Space Shuttle payload bay in late 2001.

Photograph of the Ranger craft testing in a neutral buoyancy underwater environment.

Robonaut

The objective of NASA's Robonaut Robotic Surrogate is to develop an anthropomorphic robotic astronaut surrogate to perform high-payoff and high-risk EVA tasks and provide quick responsiveness for EVA contingencies. To maximize overall EVA productivity and to increase crew safety, a robotic astronaut surrogate could perform tasks, such as worksite setup and close-out, in place of an astronaut. Also, a robotic surrogate could be used as an assistant astronaut working alongside a suited EVA astronaut, freeing up the second EVA astronaut to perform a parallel task, reducing the total EVA time required for the job.

Photograph of Robonaut, with dexterous arms and hands to assist astronauts.

Research on Robonaut focuses on dexterous manipulation with high degrees of freedom, integrated mechanical design, and ease of control. The Robonaut program began in 1997 and is ongoing.

Robonaut is applicable to many future EVA missions. The most immediate application will target the International Space Station, as directed by NASA's External Work System Program.

The graphical visualization and interface for Robonaut, demonstrating Robonaut in a space setting.

SPIDER

SPIDER is a teleoperated robotic arm designed by the Agenzia Spaziale Italiana (ASI). The arm is designed for payload servicing on the International Space Station. The arm has seven degrees of freedom and includes a simple two-fingered gripper end effector with force, torque, and tactile sensing. The next generation, currently in development, will be lighter weight, based on multi-finger end effector, and equipped with active matrix-like tactile sensor. Vision and proximity sensing also will be integrated. No date has been set for launch.

Photograph of SPIDER, the teleoperated robotic arm designed by the Agenzia Spaziale Italiana (ASI).

The Supplemental Camera and Maneuvering Platform (SCAMP)

The objective behind the Space Systems Laboratory's (University of Maryland) SCAMP is to build a vehicle that can give ground controllers and astronauts better views of EVA excursions and to inspect worksites in orbit. The initial SCAMP was designed to simply oversee the activities in the water, a free-floating camera platform. As the SCAMP design progressed, greater capabilities were added. SCAMP was first declared operational late in the summer of 1992 and currently is developing technologies to extend the abilities, roles and operations of free flying camera platforms.

Photograph of the first generation SCAMP, testing in an underwater environment.

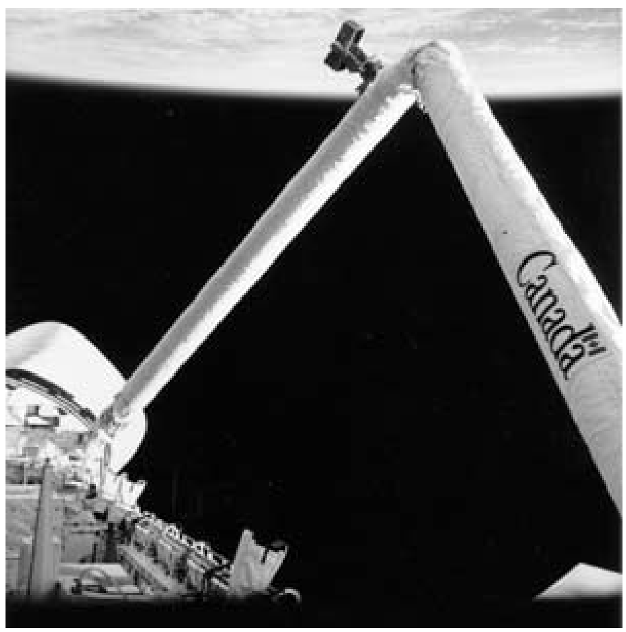
The SCAMP SSV (Space Simulation Vehicle) became operational in 1997. Design efforts have begun on the next generation for neutral buoyancy operations. Projects for a planetary imaging probe, a Europa under-ice explorer, and a personal imaging system for intra-vehicle operations are under conceptual study.

4.4.4 Directly Controlled

Shuttle Remote Manipulator System (SRMS/Canadarm)

The Shuttle Remote Manipulator System, nicknamed Canadarm, is a directly controlled robotic arm designed for operations on the Space Shuttle. It was developed by Spar Robotics, now MacDonald Dettwiler Space and Advanced Robotics, Ltd. The arm provides precise and delicate handling of Shuttle payloads by astronaut operators without necessity for extra-vehicular activity. The arm was first deployed in 1981, was used for the repair operations on the Hubble Space Telescope to help retrieve and redeploy the telescope, still is used on missions today, and is expected to play a role on the International Space Station.

The arm has a two degree-of-freedom shoulder, an elbow, and a three degree-of-freedom wrist. The end-effectors are specifically designed for manipulation of specific payloads. It is capable of manipulating payloads up to 266,000 kilograms in space. Direct control of joint angles, joint velocities, and end effector position and function is required.



Photograph of Canadarm (Shuttle Remote Manipulator system) in operation on the US Space Shuttle.

4.5 Tools

4.5.1 Autonomous

Adaptive Problem Solving (APS)

The Adaptive Problem Solving (APS) project is a collaborative effort between JPL's Planning and Scheduling Artificial Intelligence Group and JPL's Operations Mission Planner. APS is intended to solve large-scale scheduling and resource allocation problems through the development of self-customizing schedulers. The scheduling system automatically determines heuristic strategies customized to domain-specific problem distributions and constraints. One goal of the project is to automate the process of discovering the domain specific search strategies needed for scheduler customization. The result will be increased performance of the automated scheduling systems so these systems can be used interactively, and higher schedule quality to better utilize scheduled resources.

The APS has been in development for scheduling the Deep Space Network (DSN) to provide faster and better scheduling for dish pointing. In 1997, statistical machine learning techniques applied to the APS simulation led to strategies that improved on human expert derived strategies by decreasing time to create a schedule by 50% based upon DSN project requirements and actual orbit configurations. APS was also able to solve 15% more problems than the human experts. Currently, the project is extending techniques to allow for specialization of control strategies with empirical learning methods and to allow control of constraint relaxation to improve schedule quality.

Associate Principal Astronomer (APA)

The APA, which performs functions in support of a human principal astronomer, is a joint project with NASA's ARC, Tennessee State University, and Fairborn Observatory. The primary goal of the APA project is to design, build, test, and widely distribute a system for the low-cost, efficient management and closed-loop operation of multi-user, remotely located, automatic telescopes. Primary technical innovations in the APA are in the area of scheduling. The APA provides new and extremely powerful techniques for scheduling observations over an observing season and for sequencing observations within a night. The scheduling techniques are used to automatically derive a good observing sequence and to automatically use this sequence on a remotely located telescope. The APA accepts observation requests from a telescope's user and returns the results to the user through existing Internet e-mail and World Wide Web infrastructure. The requests and the results are specified using the Automatic Telescope Instruction Set (ATIS).

The current version of the APA is in operation on one telescope at Fairborn Observatory, where it is undergoing evaluation. A WWW based variant of the APA, the Service Observing Associate (SOA), has been developed to support operation on larger, non-automatic telescopes. On such telescopes, blocks of time (typically one or more nights) are allocated either to the scientists to carry out their own observations or to an observatory staff astronomer who executes the observations of multiple scientists (called service observing). In service observing, the schedules generated by the SOA are executed by the service observer, who initiates dynamic rescheduling. A prototype SOA system is currently undergoing testing.

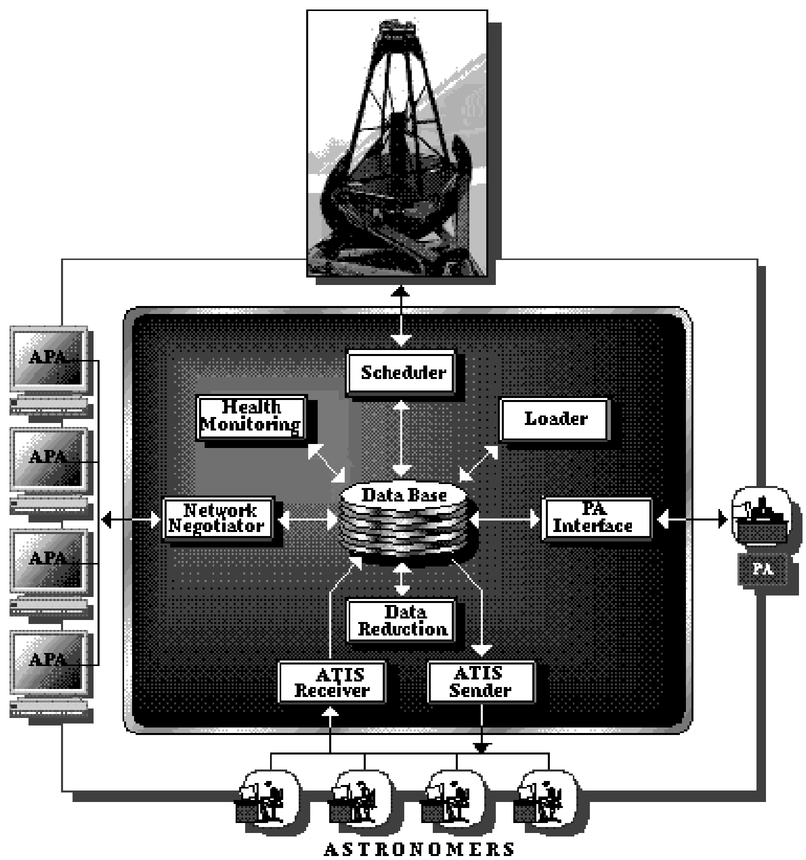


Illustration of the Associate Principal Astronomer architecture.

AutoNav

AutoNav, developed by JPL and Ames Research Center, is an autonomous localization system for spacecraft which uses celestial references to determine the position of the spacecraft within the solar system. The navigator images the sky with a predefined sequence of orientations in order to observe celestial features. The positions of nearby bodies, such as asteroids, relative to the "fixed" stars can be compared to celestial maps in order to determine position. In addition to localization, AutoNav has the capability to determine necessary engine commands for correcting the spacecraft's attitude based on a model of power availability and consumption, gravitational forces, and position.

AutoNav was specifically designed for the Deep Space 1 mission to comets and asteroids. It has successfully localized Deep Space 1 since its launch in 1998 and has controlled the new ion propulsion system.

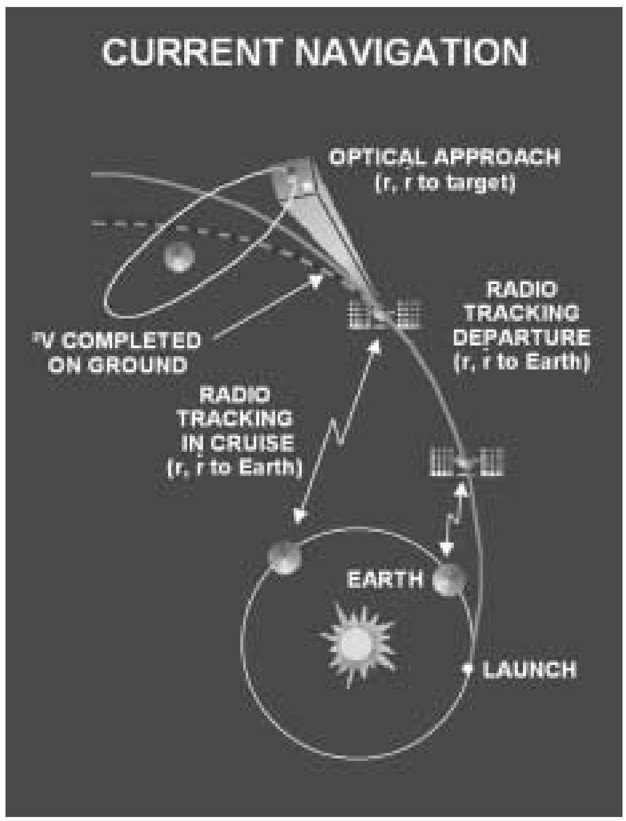
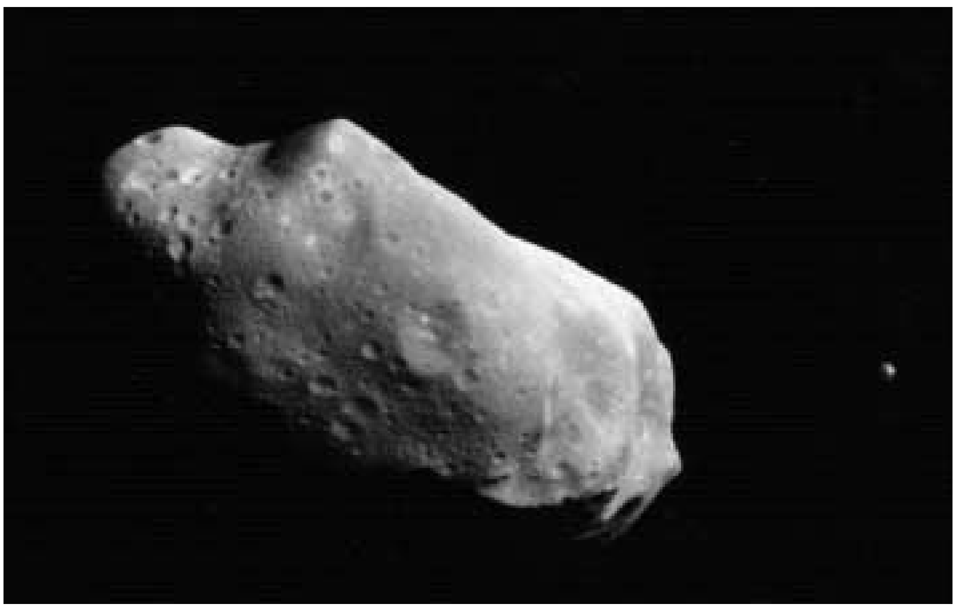


Illustration of the types of paths generated by AutoNav for Deep Space 1.

Autonomous Satellite Detection

An automated, on-board natural satellite detector such as Autonomous Satellite Detection offers the potential to detect and flag interesting and unexpected satellites for inspection during the course of a mission. Autonomous Satellite Detection, a prototype system, has been developed by JPL's Machine Learning Systems Group to perform this task. Automated detection of satellites on-board autonomous spacecraft would allow tracking the satellite for a duration sufficient to determine its orbit, which in turn is used to infer its mass. It may also enable production high resolution images and collection of scientific data (such as spectral data for composition) of unexpected targets by telling the spacecraft to point instruments or change course appropriately. Studies of important issues such as cratering history, satellite shape and surface geology can be conducted on these new, unpredicted targets. The ability to perform these unexpected experiments autonomously becomes increasingly important for spacecraft at distances from Earth which make direct operator intervention too slow to react opportunistically.



A photograph of asteroid Ida and its moon, Dactyl, typical of those used to test the classification system of Autonomous Satellite Detection.

Implemented on a space mission, this tool will identify candidate satellites in situations in which they consist of a very small number of image pixels registering barely above ground. The prototype has considered the simplified situation in which both the spacecraft and the satellite are stationary. It was tested successfully on all the images taken by Galileo of the asteroid Ida and its moon Dactyl, such as that shown above. All the parameters were selected autonomously by built-in procedures. This project was begun in the mid 1990's.

Autonomous Serendipitous Science Acquisition for Planets (ASSAP)

The Autonomous Serendipitous Science Acquisition for Planets (ASSAP) project seeks to develop and demonstrate the capability for adaptive autonomous science processing, data recognition and acquisition, and the derivation of super-resolved imagery from low-resolution instruments. The project is being developed at JPL by the Machine Learning Systems Group's Onboard Sciences Processing team for NASA's New Millennium Program. The project goal is to integrate a science unit with an on-board spacecraft architecture and other software modules, creating a mission in which different autonomous technologies and their interactions are demonstrated and implemented The testing of this integrated approach is performed on the Flight Systems Testbed at JPL.

Autonomous science object recognition would allow a spacecraft to identify targets of interest and generate observation requests, allowing the system to bypass limitations placed on direct control by bandwidth and Deep Space Network usage restrictions. Such autonomy would allow a spacecraft to report high-level measurements, rather than raw data, which reduces the bandwith required for data transmission by two to three orders of magnitude. Principal investigators may prioritize the observation requests generated by ASSAP to maximize the science return of the mission. Also, autonomous and adaptive object recognizers will enable reduction in software design and development cost. Finally, the ability to derive super-resolved images from low-resolution instruments further reduces costs. This project was begun in the mid 1990's.

Autonomous Small Planet In Situ Reaction to Events (ASPIRE)

The Autonomous Small Planet In Situ Reaction to Events (ASPIRE) is a system for science planning that is being developed by JPL's Machine Learning Systems Group. Its fully autonomous processing will provide the quick reaction necessary for capturing short-term events. ASPIRE's objective is to develop and demonstrate an autonomous science processing technology which enables in-situ detection, capture, and analysis of scientifically interesting short-term comet events. To date, the fully autonomous Science module is integrated with the fully autonomous Planning and Scheduling, Navigation, Tracking, and Comet Modeling modules. The classification part of the Science module is not yet integrated; only the coordinates, direction and magnitude of change can be reported. On-board capabilities include close-proximity navigation planning and execution, on-board pointing and propulsive planning and execution, and onboard mission planning and sequencing. This allows replanning, repointing, and close observation of an event.

The first version of ASPIRE is implemented on the Flight System Testbed at JPL. It can run according to a predetermined script or in an interactive mode, where events are inserted through user interface. A mission to demonstrate ASPIRE on short-term comet events is under investigation.

Continuous Activity Scheduling Planning Execution and Replanning (CASPER)

CASPER is a planning tool that integrates replanning. CASPER supports the continuous modification and updating of a current working plan that is needed for an autonomous spacecraft to balance long-term and short-term considerations in light of changing operating context. CASPER uses the ASPEN planner's plan optimization module. The planning tool is being developed by NASA's Planning and Scheduling Artificial Intelligence Group at JPL.

Image of a simulated mission used to test the iterative replanning techniques of CASPER

Spacecraft plans often must be modified due to fortuitous events or unforeseen problems, and a planning process that is more responsive to changes in the operations context would increase the overall time for which the spacecraft has a consistent plan so that it can continue to work on the requested goals. *Iterative repair* is an approach to modify to an existing plan by incorporating new information (see also Component Technologies, Planning and Scheduling). As opposed to the traditional planning approach in which the planner uses goals and initial state to compute each plan from scratch, CASPER allows incremental changes in goals and state to invoke the planner and produce incremental changes in the existing plan. Conflicts arising in the newly updated plan are identified and resolved iteratively. The planner is then responsible for maintaining a consistent satisfying plan with the most current information and must be ready to continually modify the plan.

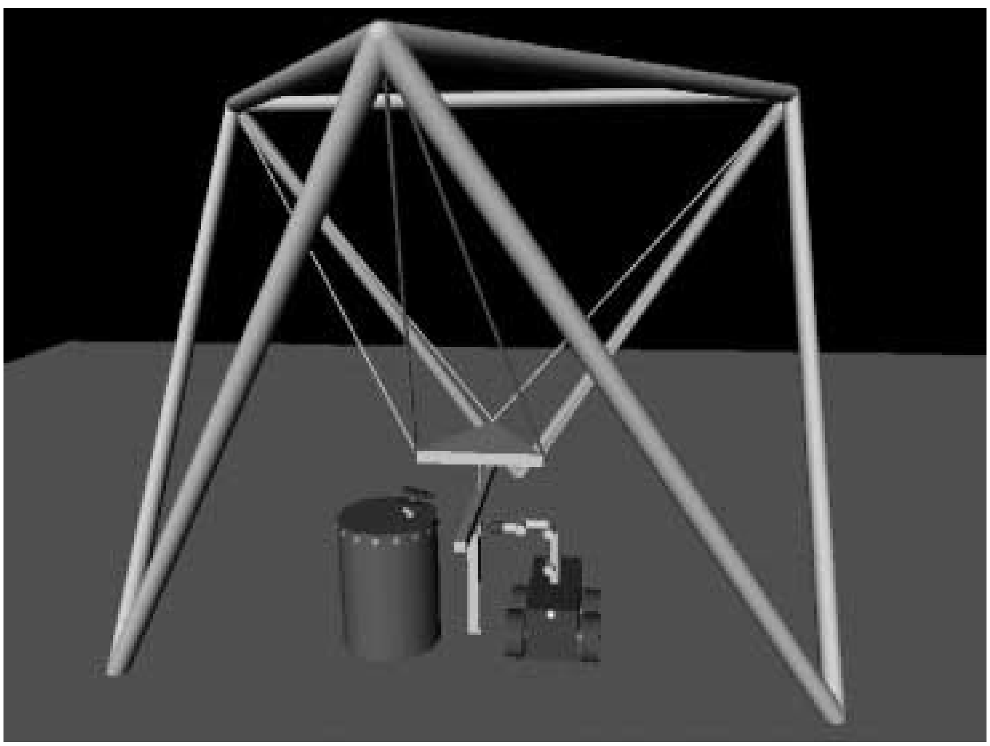
Starting from an existing plan reduces the amount of computation required to adapt the plan to new information and reduces the time required to do so. This allows the planner to be more responsive to unexpected to changes in the environment and to reduce reliance on predictive models. It also means that fault protection and execution layers need to control the spacecraft over a shorter time horizon.

Distributed Robotic Architectures (DIRA)

The Distributed Robotic Architectures project is a joint effort between Carnegie Mellon University, Johnson Space Center, and NIST. The objective of DIRA is to develop an architecture that supports the ability of multiple interacting robotic agents to react to changing environments and new information. This will enable the completion of complex coordinated tasks with reliability. The primary challenge is to maintain the individual autonomy of robotic agents while allowing them to act as a team.

DIRA is based on a three-tiered hierarchical architecture. Current research includes distributed asynchronous communications protocols and the development of task-specific algorithms to enable teams of robots to complete complex tasks.

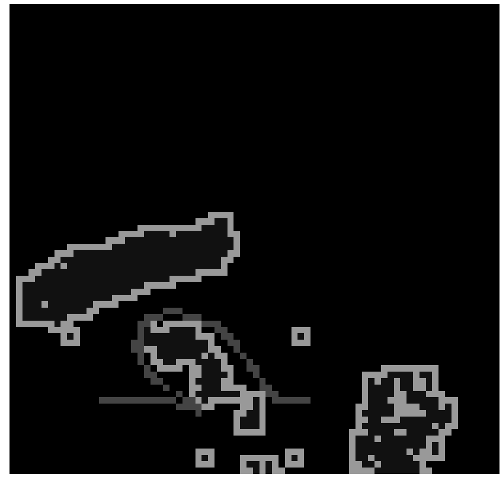
The intended platform for DIRA is a heterogeneous multiple robot system which includes Robocrane, a roving eye, and a mobile manipulator. The system will be applicable to other systems of multiple robots. Testing is ongoing using the Robocrane prototype and the Bullwinkle rover. A simulation system integrated with DIRA allows for simulated testing of these agents.



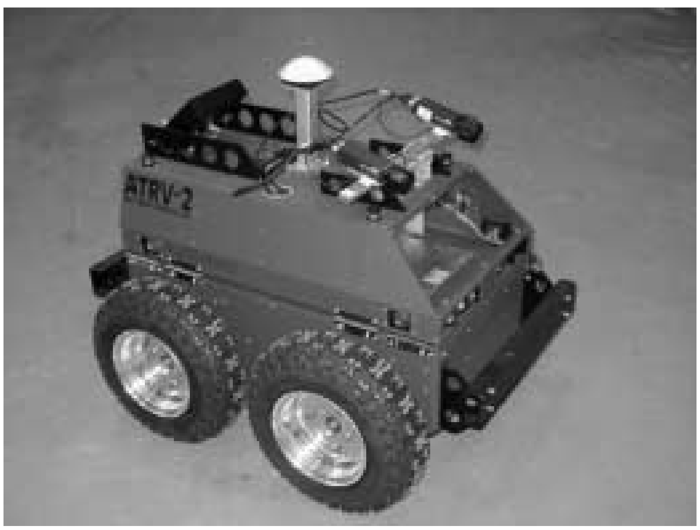
The DIRA simulator view of Robocrane and Bullwinkle working in cooperation.

Mars Autonomy Project

For greater functionality and efficiency, rovers for future Mars missions must be highly capable and autonomous, particularly in their ability to navigate safely between sites. The Mars Autonomy project is being led by Carnegie Mellon University's Robotics Institute as part of NASA's Intelligent Robotics Program. It focuses on the area of autonomous navigation by integrating previously developed local obstacle avoidance and global path planning algorithms. The resulting navigational tool is implemented in simulation and on a Mars-relevant terrestrial rover in order to demonstrate reliable long-distance navigation in Mars-like terrain. The project will demonstrate collision avoidance and route planning.



A sample path generated by the Mars Autonomy package's global (D) and local (Morphin) path planners . The path is shown in red and obstacles in blue and green.*



The Bullwinkle rover, used as a test platform for the Mars Autonomy project path planner.

The project's local obstacle avoidance package, Morphin is an autonomous system that considers the set of arcs along which the rover could move for the next few meters and, by integrating the terrain roughness along each arc, decides which paths are safe. Morphin is able to recommend good steering commands to the arbiter and vetoes dangerous commands. D* is another autonomy package being used for global planning in the Mars Autonomy Project. It is a heuristic path planning algorithm based on A* principles. It can reuse old plans when presented incrementally with new information, making it ideal for real-time, sensor-based planning on a mobile platform. The major challenge of the Mars Autonomy Project is integrating the various modules making up the autonomy system. In the future, the project will employ better positioning constraints and provide guarantees against getting lost.

Multi-Rover Integrated Science Understanding System (MISUS)

JPL's Machine Learning Systems Group is developing the MISUS, an advanced computing and robotic technology to allow a rover or a cluster of rovers to carry out autonomous scientific investigations. MISUS provides a framework for autonomously generating and achieving planetary science goals with multiple robotic agents. It integrates techniques from machine learning with planning and scheduling to enable autonomous multi-rover behavior for analyzing science data, evaluating what new science observations to perform, and deciding what steps should be taken to perform them. These techniques also are integrated with a simulation environment that can model different planetary terrain and science data. MISUS is the first attempt at coordinating multiple planetary rovers, and the project was first published in 1999.

Science data classification in MISUS is performed using machine learning clustering methods, which use image and spectral mineralogical features to help classify different planetary rock types. A planning and scheduling component is used to determine the necessary rover activities required to achieve the science goals generated and requested by the learning system. Based on an input set of goals and each rover's initial conditions, the planner generates a sequence of activities that satisfy the goals while obeying each rover's resource constraints and operation rules. The MISUS system is primarily intended for Martian biology and geology but would be applicable to other environments and investigations. Testing will be conducted on terrestrial locations such as Yellowstone (Wyoming, USA), Iceland, and New Zealand; these locations are selected because of their volcanic nature, analogous to the volcanic regions of Mars.

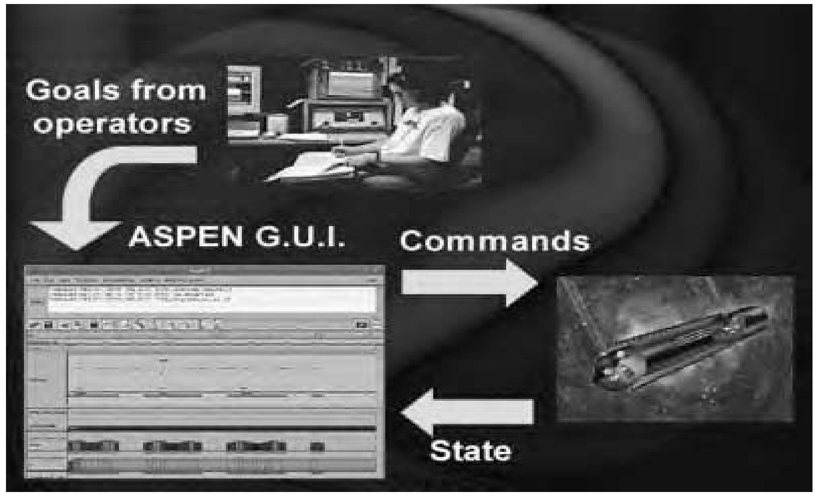
Remote Agent

The Remote Agent is an autonomous control system designed by JPL. The system uses a network model, including constraints, with both probabilistic and deterministic state transitions. High-level goals are given to the planner, such as a flight trajectory, scientific targets, and periodic tasks. The planner uses heuristical backtrack searching (that is, beginning at the goal) to produce both short-term and long-term plans. These plans are flexible, and formulated in the context of temporal dependencies and preconditions; exact values for the execution times of individual actions are determined during plan execution. Plans are periodically updated as conditions and goals change, tasks are achieved, and the spacecraft changes. Several modules divide up each task: the Planner (creates the temporally flexible plans), the Smart Executive (translates plans into alternative sequences of commands and determines the values of temporal variables) and the Mode ID & Reconfiguration (the model-based controller which executes the commands and reports results).

Remote Agent was first tested with New Maap (New Millenium Autonomous Architecture Prototype) which tested a simulation based on the Cassini Saturn orbit insertion. It was implemented on Deep Space 1 (launched 1998) for control of trajectory correction, using stellar localization, and control of science instruments. It is anticipated to be used on Space Technology 3, the Space Interferometry Mission, and possible Mars in-situ propellant and life support production missions.

4.5.2 Semi-Autonomous

Automated Scheduling and Planning Environment (ASPEN)



The ASPEN high-level architecture and graphical user interface.

Based on Artificial Intelligence techniques, Automated Scheduling and Planning Environment (ASPEN) is a modular reconfigurable application framework that is capable of supporting a wide variety of autonomous planning and scheduling applications. It is being developed at JPL by the Planning and Scheduling Artificial Intelligence Group. ASPEN provides a set of reusable software components that implements the elements commonly found in complex planning/scheduling systems including: an expressive modeling language, a resource management system, a temporal reasoning system, a set of algorithms for generating and repairing schedules, and a graphical interface. ASPEN encodes complex rover operability constraints, flight rules, spacecraft and rover hardware models, science experiment goals, and operations procedures to allow for automated generation of low-level rover sequences by use of advanced artificial intelligence planning and scheduling technology. ASPEN is unique among planning and scheduling systems because it has an easy to use modeling language, a reconfigurable framework, scalability autonomy, real-time replanning and response, and plan optimization.

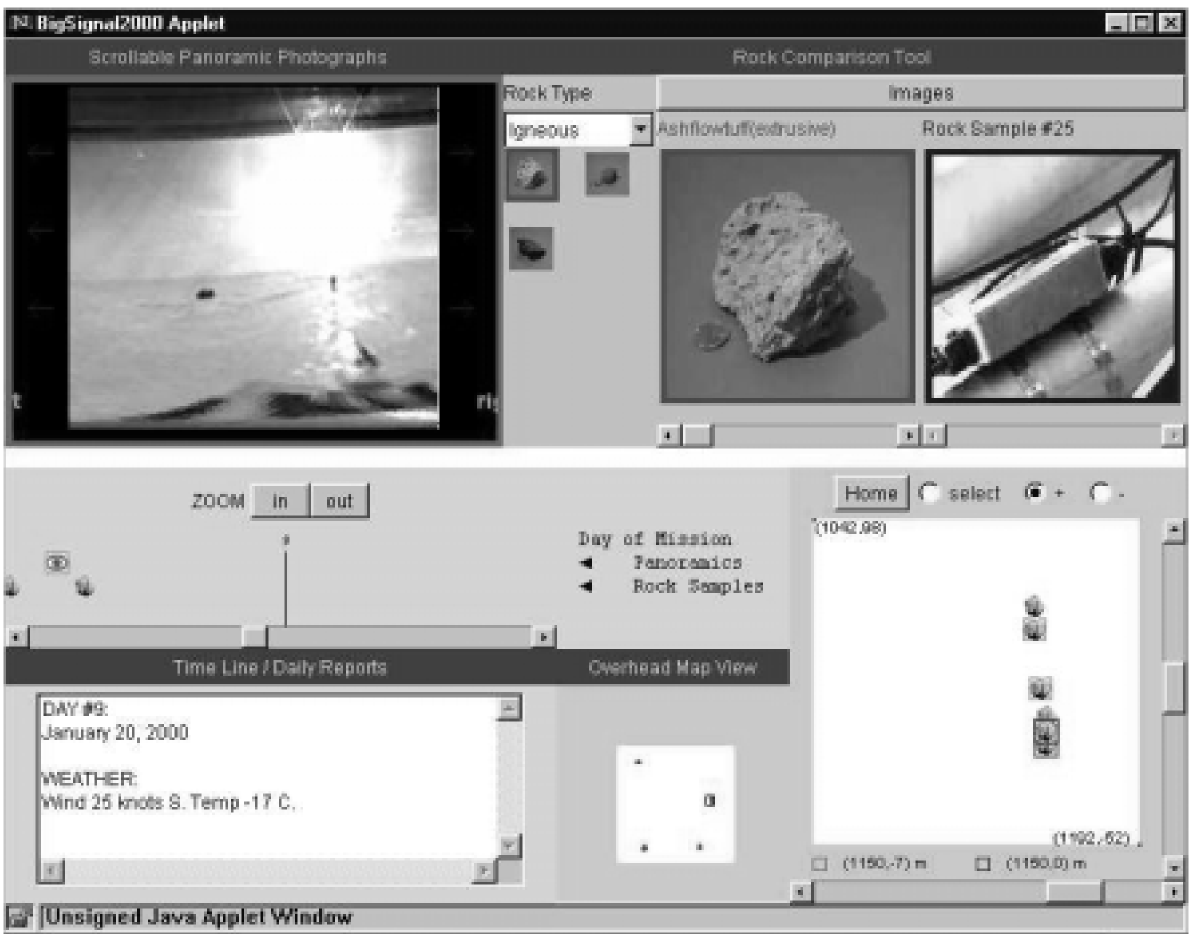
With the current use of WITS (a graphical user control interface), scientists can make requests without detailed knowledge of the operations constraints on the rover. ASPEN is being integrated with WITS to provide a more intelligent tool for rover ground operations. WITS visualizes the terrain around the rover, generates the initial sequence and sends the final sequence to the rover. ASPEN takes the initial sequence from WITS and generates a more complete and valid sequence to return to WITS. ASPEN has also been linked with CASPER for replanning. A preliminary version of the integrated system has been demonstrated on the Rocky 7 rover. ASPEN is available for external licensing.

ASPEN has been used for several missions. It was used for the design of Citizen Explorer-1 and will be used on Earth to generate command sequences for that mission. ASPEN will also provide the daily scheduling for Earth Observing-1.

Big Signal Antarctica 2000

The Big Signal Antarctica 2000 web site allowed access to data gathered by the CMU Nomad robot during its meteorite search expedition to Elephant Moraine, Antarctica in January 2000. Primarily designed as an educational tool, it allowed its users to learn about robotics as well as study images of the rocks and meteorites that Nomad found. A Java applet or HTML web page (as shown below) displays Nomad's finds on a map and a timeline simultaneously. This presents the information both chronologically and spatially to give context. Users can then compare Nomad's finds with a pre-compiled "rock library" – images of 25 representative rocks and meteorites. These rock library images can even be rotated for further inspection. Furthermore, the user can view reports from the team and panoramic images taken by the robot. This type of information interaction does not allow the user to command the robot, but does allow the user to explore in the robot's remote world, thus providing a kind of telepresence. [Coppin et al., 1999].

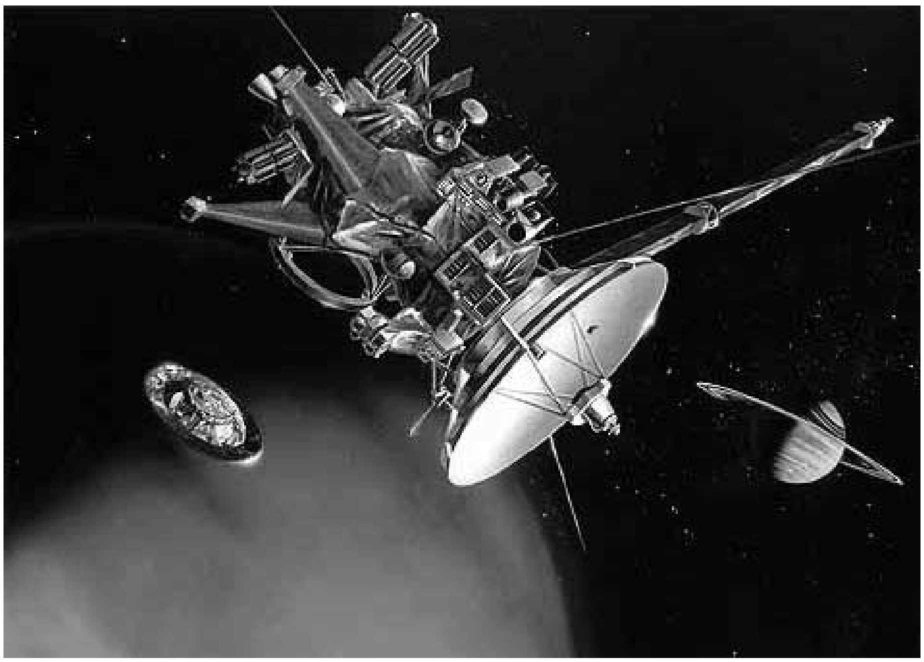
More about the Nomad robot can be found in the Systems, Planetary section of this document, as part of the Robotic Antarctic Meteorite Search project.



The Big Signal Antarctica Interface, showing camera images from the robot, the command window, and various status windows.

Cassini Operations and Science Mission Optimizer (COSMO)

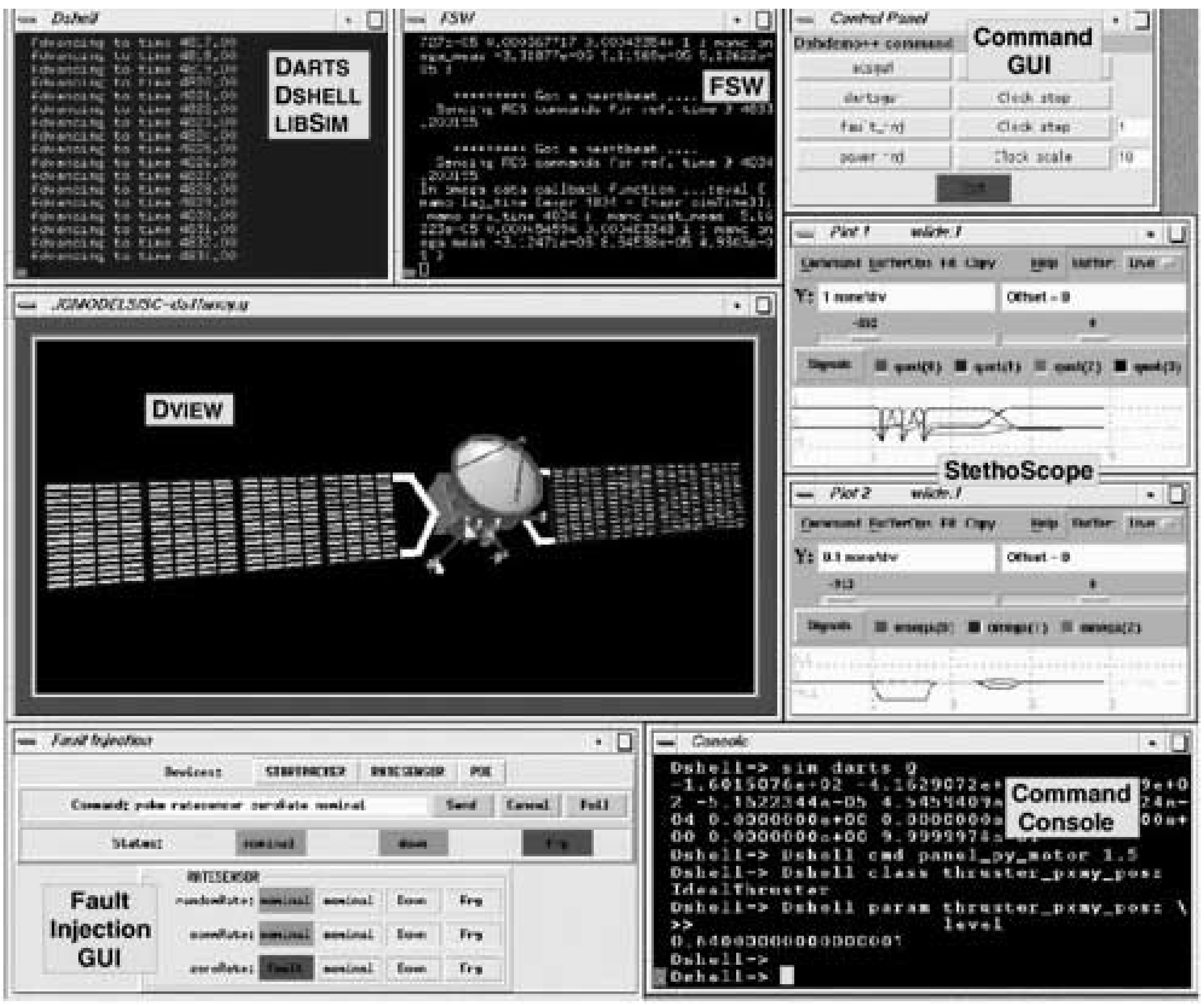
Cassini Operations and Science Mission Optimizer (COSMO) is an interactive graphical planner designed by NASA's Ames Research Center. COSMO helps scientists to schedule scientific observations on spacecraft without requiring them take spacecraft engineering constraints and other mission constraints into account; this frees the scientists from the requirement of having this knowledge in order to schedule observations with the craft. The planner is a heuristical planner that operates off-line, on Earth. Currently COSMO is being used to aid in the planning of observations using the Cassini spacecraft.



Artist's rendition of Cassini and its probe Huygens making observations of Saturn and its moons, guided by the COSMO planner.

DARTS Shell (DShell)

DShell is a multi-mission spacecraft simulation environment for real-time, hardware-in-the-loop simulations for testing and verification of flight software and hardware. Developed by JPL, it is in use on the Cassini Flight System Testbed, Multi-mission Ground Systems Office, Mars Pathfinder and Galileo. It provides the user with visual methods of interpreting the status of the spacecraft, including a simulated view of the craft itself and graphical representations of the craft's health. The command interface is also simultaneously provided to the user. The importance of such a tool is its ability to aid in the planning and execution of missions, determining control inputs in order to achieve specific goals, and free operators from performing tasks by hand. Such an accurate tool for modeling spacecraft dynamics could be of great importance for autonomous planning systems with its ability to accurately determine the results of actions.



The DARTSShell graphical user interface, showing the various fields available to the user, including the command console, a simulated view of the spacecraft and health monitoring results.

Enigma

The Enigma system was developed at NASA Johnson Space Center (JSC), NASA's lead center for human exploration of space. Enigma was developed with the primary goal of demonstrating Shuttle tasks (for example, satellite deployment) to astronauts. For this reason, it has a graphical interface which allows easy creation of key-frame animations. To create a key-frame animation, the user specifies the pose of cameras and objects in the scene at a few key frames, and the system interpolates to generate the pose at intermediate frames.

More recently, Enigma has been used by designers of the International Space Station's robotic arm. Enigma has a generic communications interface that allows other programs to control the visualization. The interface was developed so that users of the TRICK dynamics package (also developed at JSC) could visualize the results of their simulations.

Enigma has been in development for more than ten years and is a mature package with an established base of users; however, it also contains some legacy design decisions. Rather than using a standard language, such as VRML97, for its scene graph structure, it has its own native format. This makes the use of third-party tools to generate 3-D models more difficult.

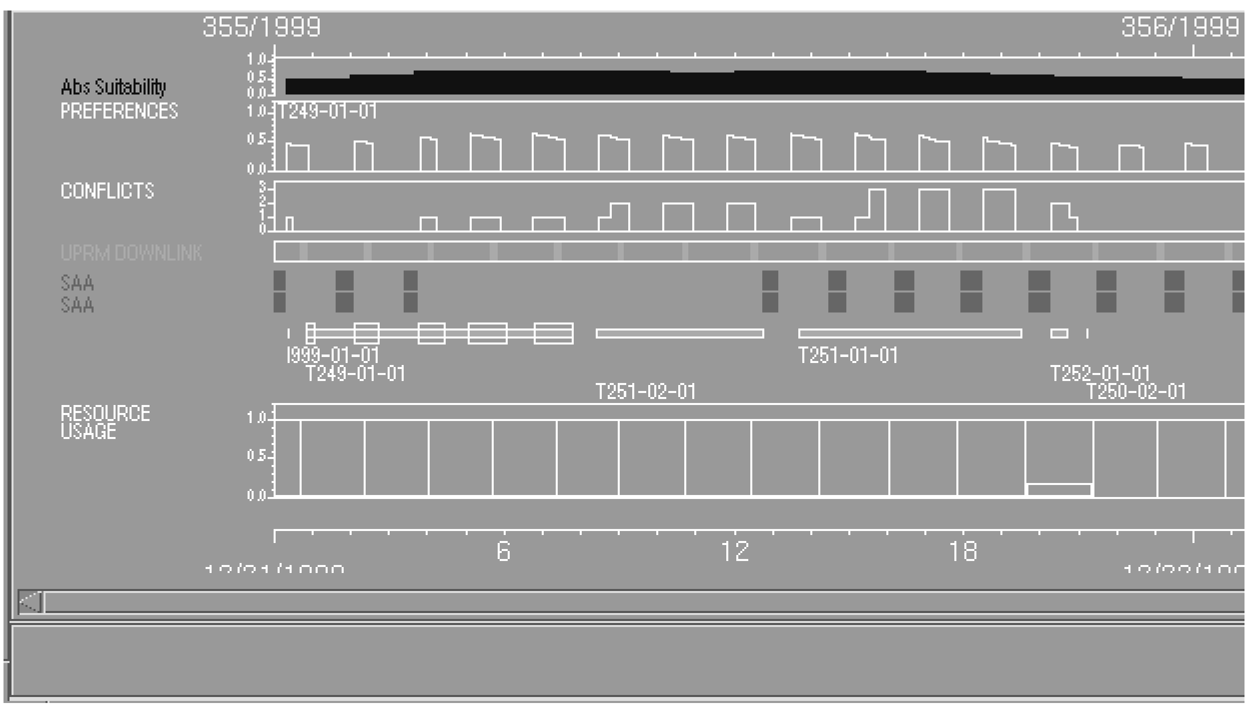
Science Planning and Scheduling System (SPSS)

The SPSS planner was developed for the Hubble Space Telescope to schedule low-level activities in order to collect specified science data. The SPSS planner is tightly integrated with the Spike system, which provides a list of tasks that are to be accomplished within specific time blocks. SPSS uses these long-term block plans to schedule specific observations at specific times within time blocks and to subsequently produce the proper sequence of low-level activities to achieve the observations. These low-level activities include vehicle maneuvering, instrument pointing for target acquisition, star tracking updates, instrument calibration, and data collection. SPSS determines the schedule taking into account all constraints, both of the spacecraft (such as minimizing maneuvering) and of the observations (such as lighting conditions, timing, sequence, etc).

SPIKE

SPIKE is a semi-autonomous ground-based planner for scheduling scientific observations on spacecraft. It is designed for use on the Hubble Space Telescope. SPIKE is capable of long-, medium-, and short-term scheduling of a list of prioritized observations which includes position and hard and soft constraints on time, observation duration, sequence, orbital parameters, preferences (a fuzzy logic approach) and resources. SPIKE is capable of replanning in the event of rare opportunities. The scheduler uses multi-start stochastic repair and approaches the problem as a constraint satisfaction problem by weighting different constraints. Sample schedules are produced using heuristics, and then constraint violations are removed. Finally the schedule is optimized.

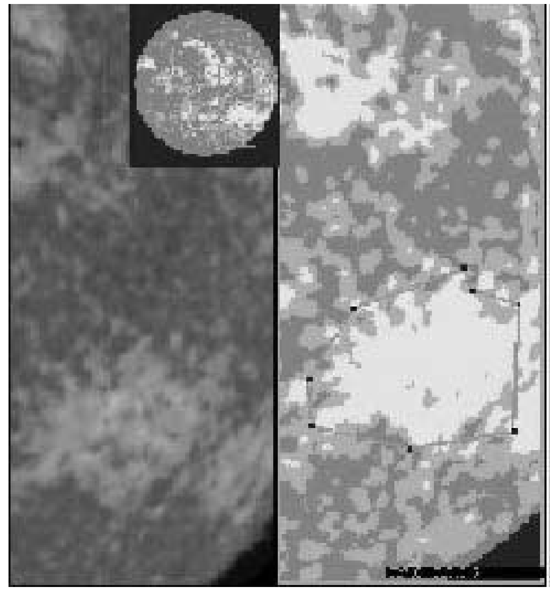
SPIKE is used primarily for the Hubble Space Telescope to schedule sequences of observations for the long-term. Observations are assigned to specific blocks of time based on a committee-assigned priority as well as constraints. These blocks are typically overscheduled, and remaining observations after a time-block is completed must be reassigned. SPIKE is also in use on several other space systems: APL's Far Ultraviolet Spectroscopic Explorer (FUSE), ESO's Very Large Telescope (VLT), NAOJ's 8 meter telescope Subaru, the JPL led Space Infrared Telescope Facility (SIRTF), Harvard's Chandra X-ray Observatory (formerly Advanced X-ray Astrophysics Facility, AXAF), two Earth-observing missions by GSOC and DLR (MOMS-2P and MIR97), UC Berkeley's Extreme Ultraviolet Explorer (EUVE), GSFC's Advanced Satellite for Cosmology and Astrophysics (ASCA), and GSFC's X-ray Timing Explorer (XTE).



A portion of the graphical interface for Spike used to schedule FUSE. The interface indicates preferences, conflicts, and percentage of resources in use.

StarTool

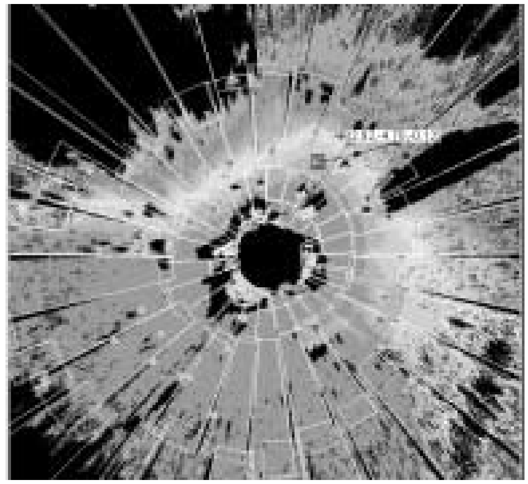
StarTool is a graphical tool, designed by JPL, for automated image segmentation that has been built on top of an image browser. The tool is designed to use the segmented images it generates to do autonomous feature classification for scientific investigations; in particular, Solar features are to be identified. Many Solar features can be readily identified by visual inspection, making image segmentation a natural approach. Autonomous identification of these features, including changes through time, will allow scientists to refine models of solar radiation, provide insight into Earth-Sun interaction, and to understand temporal patterns of occurrence of Solar features. The method for image segmentation employed uses a Markov random field model to label pixels in the presence of distortion and noise. Once segmented, the image features are classified as particular Solar feature types using a "degree of membership" in the feature classes. This degree of membership is determined by level of similarity to prototypes of the feature class. Currently, the system is operating on images of the sun acquired from independent sources. StarTool was first published in 1996.



Portion of the graphical interface of Startool, showing images at different zoom levels, and with highlighting (green) on areas of interest.

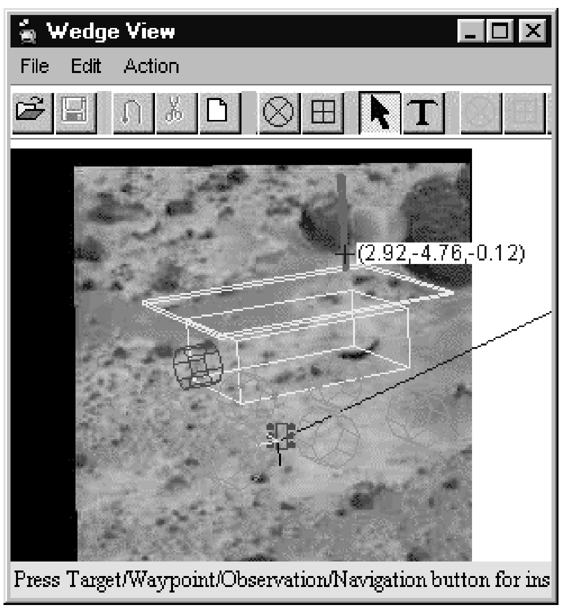
Web Interface for Telescience (WITS)

WITS is a graphical interface for spacecraft model-based control. Developed by JPL from 1996-1998, it also is used to simulate spacecraft behavior and can be used to directly control the spacecraft in a visually intuitive way. WITS is used to simulate spacecraft and environment and the interactions between them.



The WITS grpahical interface, a sample of the panoramic view of terrain generated by the Sojourner rover.

Three-dimensional terrain data gathered by the robot's stereo camera system allow the user to take length measurements of features in the robot's local area and annotate interesting or dangerous locations. The user can also access models of the robot to describe sensor deployment or robot movement commands to the system. With these tools, science goals are defined as waypoints (intermediate goals) and sensors to be deployed. Engineers then verify these waypoints. Finally, with an ordered list of activities, command sequences are created using the ASPEN automated scheduling system [Backes et al 1998]. This system was successful in reducing the time required to generate a Mars Pathfinder command sequence to about 12 hours from many days. Simulation results can be directly applied to controlling the spacecraft by testing sequences of commands and are used as the interface by which commands input by a user are directly transmitted to and executed by the spacecraft.



The WITS graphical interface, a sample of the system model window, showing Sojourner on Martian terrain.

Data provided by Sojourner allowed testing of the WITS system and use of the WITS system for educational outreach. The first flight system using WITS for real control was the Mars Polar Lander, which used WITS as the control interface for the robot arm. Mars Polar Lander was lost before landing on Mars in 1999, so the system has not yet been tested in space.

A sample WITS command sequence generated for a demo:

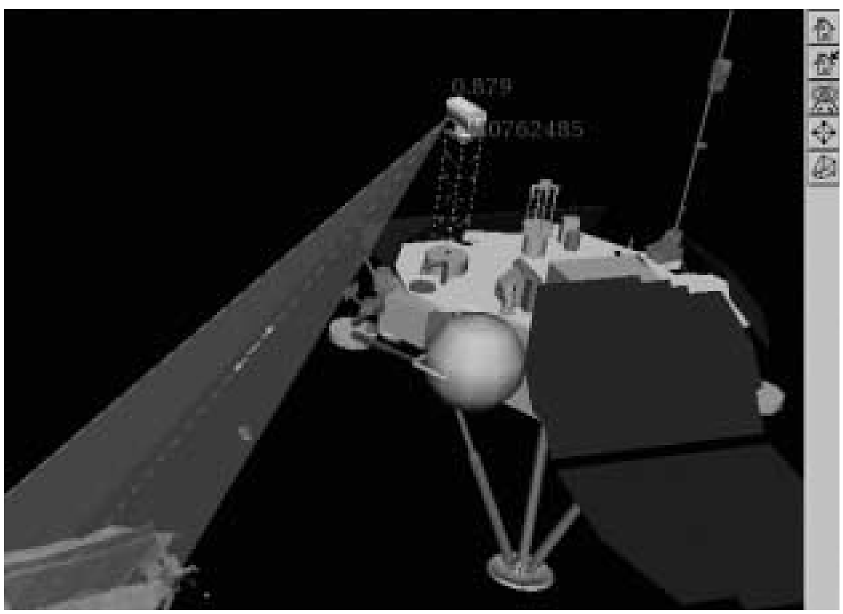
go_to_location(rover, anytime, 1)
dig(L2, anytime, 13)
spectrometer_read(L3, anytime, 14)
dig(L4, anytime, 15)
image_nav_time(L5, 1998-092/10:15:00.000, 16,full,now,60)

VIZ

Viz is the latest system in a series of tools for dealing with planetary rovers and landers. Viz is optimized for high-end graphical workstations with multiple processors. On these platforms, it continues to allow smooth user interaction even when the graphics engine is in communication with several other pieces of software.

A focus of Viz development is the rapid prototyping of graphical interfaces that supplement 3-D visualization. The communication interface is implemented in several different programming languages, including Java and the Python scripting language. Viz is distributed with an example science interface that was used on the Mars Pathfinder mission, and allows the user to easily pick objects in a 3-D model and measure their distance, area, and volume. Like Enigma, Viz also has been used to create key-frame animations. However, Viz cannot export these animations to widely used movie formats.

As a recently developed package, Viz is less mature, but draws from its use of the OpenInventor graphics library (distributed by Template Graphics Systems). This library enables Viz to read the widely used VRML 1.0 and VRML97 scene formats, gives it a powerful user interface, and is continuing to expand in capability.



A simulated view of a lander in VIZ, indicating in highlight the region that would be visible to the camera.

5 The Future of Space Robotics

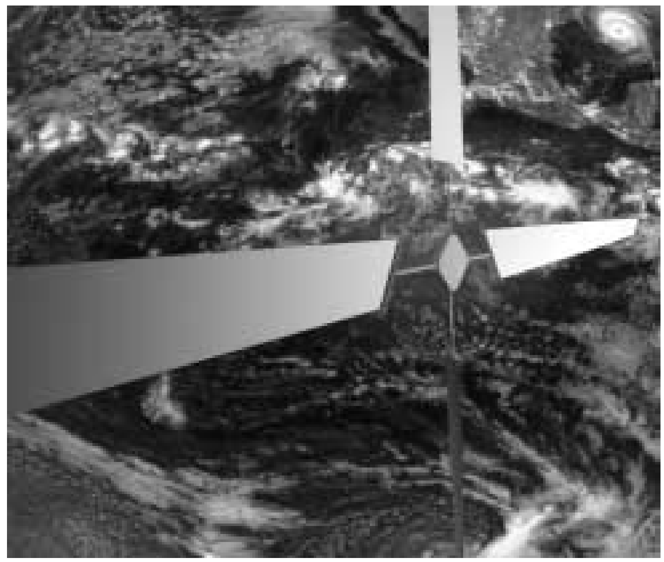
The future of space robotics is discussed in this chapter. Several space program themes are explored, including twenty-year projections for development and the particular challenges in achieving these long-term goals. Additionally, the areas of research key in making progress in these areas are discussed.

5.1 Space Program Themes

The space program themes discussed here each represent a type of space mission. The role of robotics is discussed for each of these themes and a twenty-year development schedule is suggested. In addition, the particular challenges facing roboticists working toward these missions are outlined.

5.1.1 Deep Space Exploration

Robotics has a role in deep space exploration in several avenues, and is the most demanding on robotics technology. Autonomous spacecraft can be positioned at locations in the solar system where long-term operations are required, such as observation or communication. Other craft can explore the solar system over distances and timeframes too long for supporting astronauts, and return large amounts of data to Earth. Lastly, such craft can pass out of the solar system and intelligently explore and report back, far beyond the capabilities of previous explorers. Such spacecraft operating at large distances from Earth cannot depend on reliable human intervention.



Artist's conception of a Solar Blade, an autonomous, light-weight, exploration robot.

<u>Twenty Year Projection: Deep Space Exploration</u>

- Current: Some intelligent craft, like Deep Space 1, can do autonomous planning
  Conventional data-relay craft are the common mode
- 5-10 Years Solar powered spacecraft exploring the solar system
  Autonomy for satisfying mission goals
- 20 years Light-weight spacecraft with new types of propulsion exploring beyond the solar system
  Autonomy for adjusting mission goals

Research Challenges

The primary research challenges for deep space exploration include mission weight and autonomy.

Autonomy

Deep space missions cannot rely on human intervention at any level. In this respect, they stretch the limits of the requirements for autonomy. Not only must they perform self-monitoring and maintenance, they must be capable of adapting to any new situation, opportunistically taking advantage of circumstances, and making high-level mission decisions.

Light-Weight Craft

Large amounts of conventional fuels are required to propel craft to the outer edges of our solar system and beyond. Larger weight craft (requiring more fuel) are also more expensive to launch. Craft like the Solar Sail must be developed to reduce weight and cost. Current images of the Solar Sail, like the Solar Blade in development at CMU, include four blades of 8-micron Kapton (20 by 1 meter each) supported by a bow-and-arrow-like structure. A highly intelligent computer system must accurately (and intelligently) control the blade pitch motors; station-keeping is performed using only solar pressure and must keep an appropriate orientation with respect to the sun. Solar Blade is expected to prove maneuvering capabilities within Earth's orbit, and then fly past the Moon. It will deploy an ultralight heliogyro spacecraft and spiral out into the solar system.

5.1.2 Planetary Exploration

Planetary exploration missions require unmanned vehicles to navigate local features, find long-range goals, and perform missions of science or construction (such as in preparation for a colony). Near-bodies, such as the Moon and Mars, can accommodate some human assistance, though not always in a timely fashion. Further bodies, such as asteroids or outer Moons, must possess much the same level of autonomy as deep space craft.

Sojourner rover on Mars.

Twenty Year Projection: Planetary Exploration

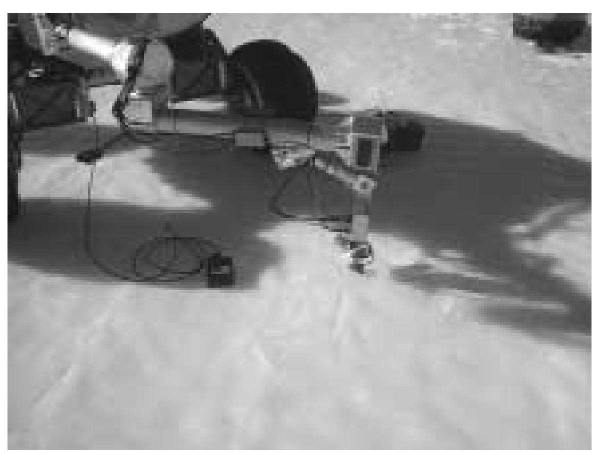
- Current: Wheeled rovers capable of traversing 100's of meters over conventional terrain
  Supervised autonomy requiring occasional intervention and rescue
- 5-10 Years Rovers with multiple mobility modes
  All-terrain traversability at ranges of 100's of kilometers
  Extended periods of autonomy
- 20 years All-terrain traversability at ranges of 1000's of kilometers
  Unattended autonomy

Research Challenges

The primary research challenge for planetary exploration is long-range navigation. A robotic planetary explorer must solve path planning, path generation, obstacle avoidance, and motor control simultaneously. The current approach to solving this problem at CMU is demonstrated by systems such as Mars Autonomy. In theory, the problem is separated into local and global navigation for path planning and execution. Use of grid-based, rather than exact, representations eliminates the need for an exact map and reduces computational requirements. The space is represented at multiple levels, depending on need. Focus is also placed on development of an efficient, incremental path planner. Simulations are being conducted on computer and on real robots in simulated environments. Sensing and robot dynamics models are at resolutions lower than required to adequately navigate over long-range at high speeds. Additionally, the processing required cannot keep up at high speeds on current computing systems.

5.1.3 Science

Most planetary and deep space missions in development and consideration by NASA are scientific missions. Development of autonomous science in both venues will make such missions more effective, more efficient, and in some cases makes possible what is impossible with more conventional craft.



The Nomad rover performing autonomous analysis and classification of rocks, searching for Antarctic meteorites.

Twenty Year Projection: Science

- Current: Execution of sample acquisition and target data collection with single command
- 5-10 Years Autonomous sample acquisition and partial data analysis
- 20 years Autonomous sample acquisition and data analysis
  Autonomous target selection for achieving mission goals
  Autonomous selection of analytical instruments

Research Challenges

The largest challenge in autonomous science is the development of planning algorithms that can determine targets of interest based on multiple mission goals, plan and adapt long-term paths to accommodate the mission goals, and the complex analysis and processing of data from multiple sensors. Currently, this has only been possible in simplified environments (such as rocks on ice, rocks are terrestrial or meteorite) using few sensors and a well-trained learning program.

5.1.4 Outposts and Colonies

Robots offer a remote presence prior to human arrival. These robots can survive in extreme environments (temperature, radiation, etc) without life support, hibernating when necessary. Thus, site preparation by robots can reduce risk to humans by exploring the area and setting infrastructure in place ahead of time. Once humans arrive, robots can still assist the colony by performing mundane and dangerous tasks, and by extending the senses and capabilities of the humans.

Artist's conception of a Lunar outpost for humans and robots.

Each of these tasks can only be accomplished efficiently with multiple robots. This provides flexibility, redundancy and fault tolerance, high efficiency, and more inexpensive systems (rather than a single system that is very large and capable of all tasks).

The motivations for colonies are several. In addition to the basic desire to establish a foothold in space, permanent colonies on extra-terrestrial bodies can reduce the costs of deep space missions by reducing the material that must be lifted from Earth's gravity. The Moon is highly accessible from

Earth using well-developed technology, minimum fuel, and short transfer times. Communications can be established using high bandwidth and low latency. The Moon also offers a resource that provides for a robot-friendly environment: abundant solar power.

<u>Twenty Year Projection: Outposts and Colonies</u>

- Current: Supervised coordination of single rover with single lander
  Required daily communication with Earth
  Small-team autonomous coordination in terrestrial, simple applications
- 5-10 Years Small teams of simple robots conducting wide-area search and data collection
  Rare (weekly or monthly) high-level instructions from Earth
  Highly self-sustaining systems with some robotic repair and maintenance
- 20 years Permanent presence in space of large robot teams for construction and science
  Rare required (monthly or yearly) intervention from humans

<u>Research Challenges</u>

In addition to the obvious autonomy issues discussed for other types of missions, two main challenges in robotics technology exist for development of robotic colonies in space: reliability and coordination.

Reliability

Current robots may last for several years, but require frequent repair. Long-term robotic presence (particularly in the absence of humans) must develop a higher level of reliability. Self-repair or specialized repair robots are one means of increasing reliability, and projects are ongoing in this vein. Additionally, greater reliability in components (motors, power sources, computers, sensors, etc) will increase the overall reliability of robots and multi-robot systems.

Coordination

Coordination for colonies must occur at the mission, control, and task levels. Coordination must be possible without accurate inter-robot calibration. Coordination applications include docking, moving large objects, mapping, and construction. Currently, research has been most successful in coordinating robots at the control and task levels.

5.1.5 Orbital Construction and Service

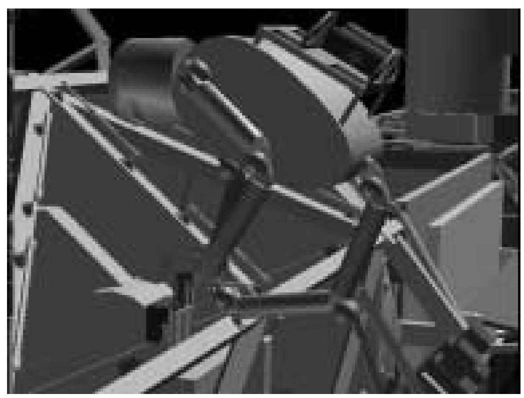
Orbital facilities, such as the International Space Station and the proposed orbital power stations are on a very large scale. Power facilities are anticipated to be kilometers long and massing millions of kilograms. These structures, consisting of millions of parts, will take many years to build and must be operational and maintained for decades. The orbital environment is quite hazardous to astronauts due to the radiation, wide temperature ranges, and lack of atmosphere. As a result, this type of mission is another ideal application for robots.

Twenty Year Projection: Orbital Construction and Service

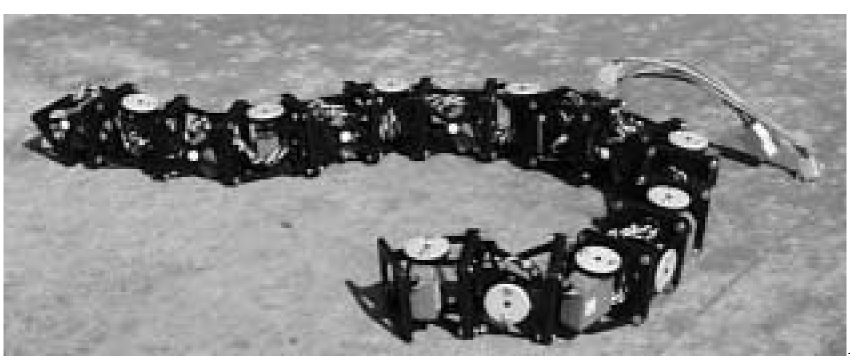
- Current: Extra-vehicular activities conducted by astronauts with teleoperated arm
  Semi-autonomous free-flyers for observation and inspection
- 5-10 Years Semi-autonomous EVA robots for specific space-structure repair
- 20 years Autonomous robots for general space-structure repair and construction

Research Challenges

Several research challenges must be met for the orbital construction and service robots to be developed. Such robots may be required to operate in proximity to astronauts and/or sensitive equipment. The complexity and small scale of tasks requires highly maneuverable systems with a high-degree-of-freedom control. Many of these systems may need to function within specific structures, such as the trusses proposed for orbital power stations. Most robots will also be required to locomote as well as manipulate, perhaps with the same limbs. Operations in orbit also involve the complication of working in micro-gravity environments. All of these things must be done with a high level of autonomy; humans cannot directly or frequently supervise the numbers of robots required to do such large-scale construction.



An anthropomorphic service robot at work, artist's conception.

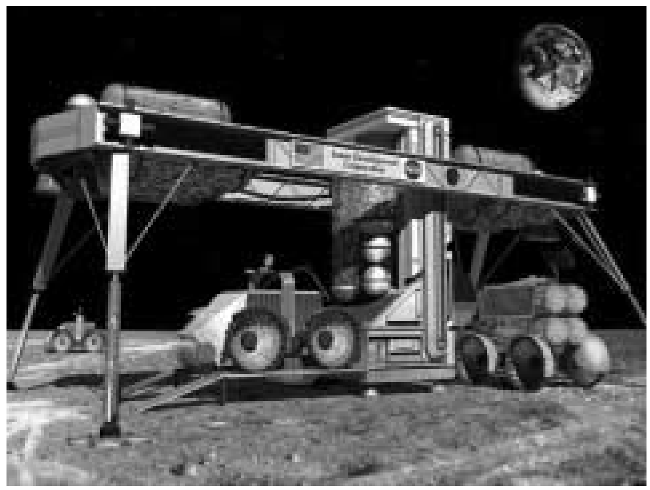


A prototype serpentine robot.

Attempts to solve these problems have resulted in diverse designs. Humanoid robots have been proposed to mimic the dexterity, range of motion, and sensing capabilities of humans as well as to present a familiar face to the humans with which it must interact. Such humanoid robots, such as JSC's Robonaut, may be most capable of efficiently using equipment and structures initially designed for humans. The complexity issues are most directly related to this type of design. Multipods, like UMD's Ranger, with interchangeable tools and grasping hands/feet are proposed for autonomous assembly, inspection, and maintenance. The challenges of multi-purpose limbs and truss negotiation are specifically attacked with this design. Current projects, such as CMU's Skyworker, address the issues of lightly walking on a truss structure, low-energy locomotion, and sensing while working. Serpentine manipulators, capable of threading through tight spaces such as trusses, grasping, and limbless motion, have also been proposed and investigated. These suffer from more complex path and motion planning issues.

5.1.6 In-Situ Resource Extraction and Utilization

Development of systems that process and/or use in-situ resources can support the development of colonies, provide fuels for spacecraft return voyagers, provide power, and provide raw materials for construction. Due to the hazardous environment of space, the need for complex mechanisms, and the tedium involved in such work, robots are ideal for this work.



Artist's conception of a robotic worksite for in-situ resource processing.

Twenty Year Projection: In-Situ Resource Extraction and Utilization

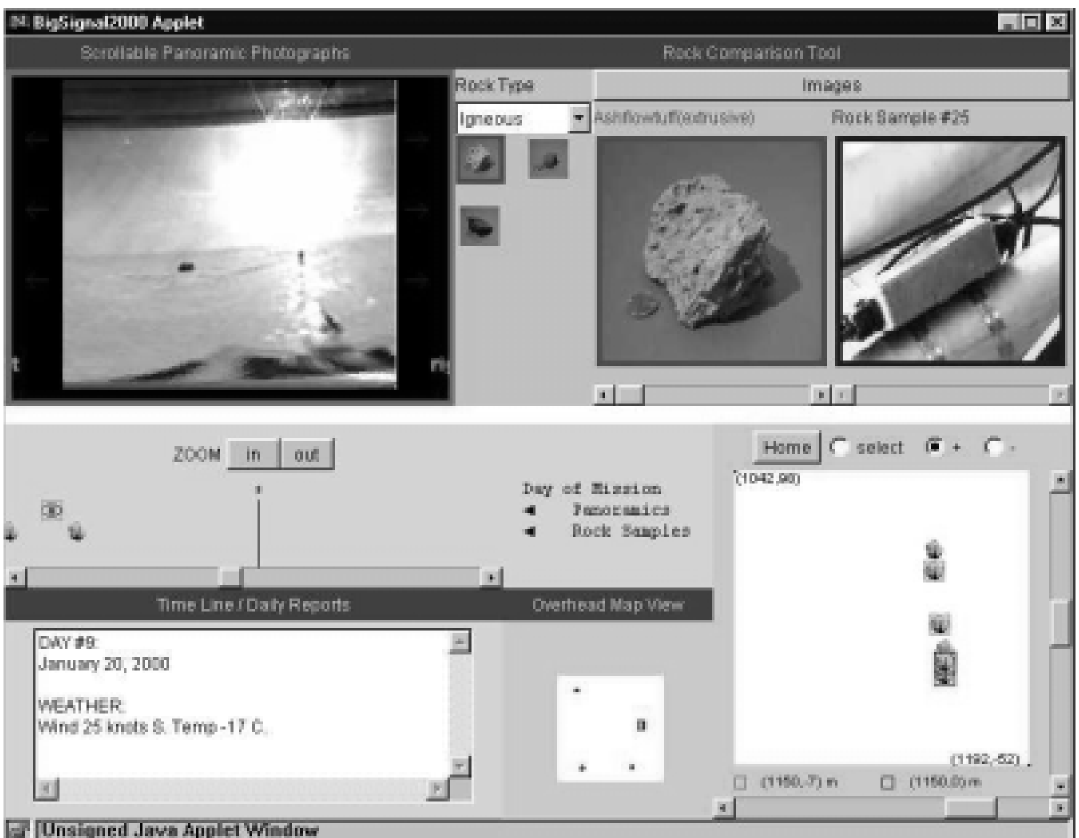
- Current: Mars precursor experiments scheduled test propellant production technology
  Subsurface capability to 10 centimeters.
- 5-10 Years Base technology for Mars in-situ propellant production through micro-gravity soil and subsurface resource processing
  Subsurface capability 10-100 meters
  Self-refuelling production robots
- 20 years Autonomous interface with non-production robots for refuelling
  Power supplied to robotic and human facilities
  Subsurface capability greater than 100 meters

Research Challenges

Research challenges to in-situ production are several. The power generated by such efforts must be greater than the power consumed by the vehicle in order to be efficient and vehicles must be small enough to be launched cost-effectively. Currently, resource-processing machinery is very power-intensive and heavy. Additionally, technologies from various applications must be successfully merged: waste cleanup (such as CMU's Pioneer for nuclear cleanup), autonomous resource gathering (The CMU Demeter for agriculture and the autonomous coal miner), and autonomous excavation and loading (also at CMU).

5.1.7 Human-Machine Systems

Robotic assistance in surface EVA will allow tasks that are more easily done by robots (due to physical demands or environmental concerns) to be performed by robots under supervisory control by humans. Additionally, it will allow for humans and robots to work together, making use of the strengths of each, to achieve complex tasks. Robotic technology can also allow data to be available and visualized in multiple modes by scientists and students very soon after collection via the World Wide Web. More efficient control of robots may also be accomplished through three-dimensional graphical Web interfaces.



Example of current scientific human-machine itnerface.

Twenty Year Projection: Human-Machine Systems

- Current: Supervised robotic sample acquisition and data collection for Earth-based analysis
  Three-dimensional visualization tools for system control and data
- 5-10 Years Collective autonomy of small teams commanded at high-level from Earth
  Robotic remote assistance to Earth-based scientific analysis
- 20 years Large robot crews aiding humans in surface science operations
  Immersive environments for interaction

Research Challenges

The research challenges in human-machine systems include the determination (perhaps autonomously) of which tasks are best accomplished by robots and which are best accomplished by humans and more effective means of visualizing information.

5.2 Recommended Research Areas

The questions that scientists seek to answer by looking into space include the search for life on other planets and the search for the origins of our universe. Exploration of space directed toward answering these questions is taking our spacecraft further and further from Earth and our guidance and protection. In order to send our robotic agents into the universe to become successful field researchers, robotics research must learn how to make them independent agents, able to think for themselves, defend themselves, and to work together to achieve goals beyond the reach of individuals.

Several areas of potential robotics research stand out as essential to the further development of autonomous space machines. These areas are presented here as recommended areas for space robotics research focus for the next two decades.

5.2.1 Reliability

Current visions of space robots invoke missions lasting months, maximum ranges of one kilometer and a reliance on the survival of individual components. Future robots must function for years, operating after millions of cycles and thousands of kilometers of travel, despite harsh environments. Hardware, software, and mechanical systems critical to mission success must be repaired or replaced in the event of failure.

Robot mechanical and computational reconfiguration and redundancy can achieve the reliability necessary for long-duration missions, and current methods for achieving reliability focus on redundancy of components and agents. Robot mechanical elements must employ cleverly implemented redundant components while minimizing mass.

In addition to the research currently underway on agent redundancy and the use of identical backup systems, reliability in individual components must be increased. Computer hardware and software architectures must be robust to radiation-induced upsets, and must adapt to changes in system behavior resulting from electrical or mechanical damage or environmental shifts. Mechanical systems must be built which minimize changes due to wear and the chance of breakage. Ultimately, the probability of malfunction can be reduced but not eliminated; research into the ability of robotic systems to repair themselves must make significant progress. More research like DLR's promising Experimental Servicing Satellite and autonomous versions of the dexterous astronaut assistants is required.

Lastly, if repairs or other means of compensation for failure are to be appropriately used, the robots must also be able to detect system faults, properly diagnose the symptoms, and determine the appropriate corrective actions. The present state of the art in fault diagnosis is not yet up to the challenge.

5.2.2 Autonomy

Often operating at the limits of our ability to directly control, both in distance to our robotic agents and in the numbers of robotic agents required for complex tasks, robots for space must be imbued with independent reasoning to eliminate the need for persistent oversight by humans. Current robot operations devote engineers, scientists and deep space antennas to monitor every move of a space robot during active periods. Robot autonomy and human interface research should allow one operator to control many robots, with focused interaction necessary only in emergencies. Future robot operators will be able to direct complex tasks with a specification of the goal and constraints. Using a technology called projected existence, which is being derived from the present means of visualization, operators will look through the eyes of robots under their control, passively monitoring the progress of a task when desirable. Should a robot require assistance when presented with a particularly difficult task or an emergency, the robot operator will supersede the automatic functionality, controlling the robot at the level of manipulation or locomotion. In combined human/robot extravehicular activity, interaction between astronauts and robots will require new interfaces emphasizing speech and gesture recognition which are natural for humans and effective for scientific field use. Continued research into intelligent autonomous control software, such as Remote Agent, and projected existence must be pursued in order to achieve the required level of autonomy.

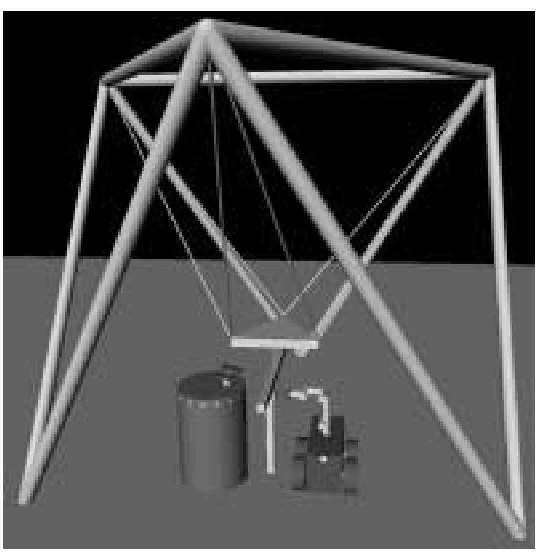
Artist's conception of a projected existence interface, to aid in the control of many, highly-autonomous robotic agents.

5.2.3 Robot Team Coordination

Planetary surveys and large-scale facility assembly are campaigns beyond the capability of a single robot. Bold agendas such as these require teams of autonomous agents working in concert. Robot teams must organize themselves to perform successfully and efficiently despite team member heterogeneity, equipment malfunction and constantly evolving goals. Research must address the design of architectures that enable decentralized coordination of multiple agents to minimize the reliance of team performance on a single lead robot. Robot teams must decompose complex tasks, delegate subtasks to individuals and reallocate jobs as conditions and goals change. Promising projects such as Skyworker, Distributed Robotic Architectures, and MISUS have only just begun to scratch the surface in developing means for many robotics to efficiently work together to complete complex tasks.

Artist's conception of multiple Skyworker robots working together to perform a complex orbital construction task.

5.2.4 Robotic Worksystems



Simulation of a multi-robot worksystem: crane, sensor, and manipulator.

The future will require robots to construct large-scale orbiting facilities that may be kilometers in extent and composed of millions of elements. Any development toward robotic or human colonies will also require large-scale construction, likely prior to human arrival. Space solar power facilities are envisioned in geosynchronous whose harsh radiation environment may eliminate the possibility of employing human construction crews. Timely construction of mega-facilities, as well as their subsequent inspection and maintenance, may require hundreds of robots working concurrently. Software architectures and communications networks must support the coordination of robots that will work together to build and maintain orbiting facilities, ensuring despite contingencies and failures. Surface robots may excavate material for radiation protection, clear surface debris for landing site preparation and transport heavy equipment from site to site. Worksystems must be light enough for transportation to a planetary surface but massive enough for moving regolith. Robots must reason through complex scenarios of earthmoving and site work to support surface facilities and in situ resource extraction.

5.2.5 Robotic Exploration and Discovery

Artist's conception of a remote planetary exploration robot.

Robots will be our agents of planetary surface exploration both independently and alongside astronauts. Future robots will handle the repetitive and time-consuming tasks of data collection, leaving humans to handle the high-level interpretation of information. Robots have established a rudimentary capability to perform science autonomously using multiple sensors and probabilistic classification methods. Research must drive autonomous science and discovery far beyond the current level, enabling efficient geological and biological surveys of vast regions. Robots must distinguish and classify a wide range of rock and mineral species with high reliability. Future robots capable of gross terrain feature identification will catalog large-scale geological formations, and will tailor their scientific examinations to the type of feature being studied. The search for lifeforms requires a fundamental development of instrumentation and classification strategies that will force us to define the essential components differentiating organisms from nonliving matter. Robots must determine paths across planetary landscapes that will lead to the greatest scientific information gain and simultaneously optimize the collection and use of solar power and other resources.

References

[Albus 1991] J. Albus. "Outline for a Theory of Intelligence." *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21(3), pages 473-509, May/June 1991.
[Albus et al 1988] J. Albus, R. Lumia, and H. McCain. "Hierarchical Control of Intelligent Machines Applied to Space Station Telrobots." *Transactions on Aerospace and Electronic Systems*, Vol. 24, pages 535-541, September 1988.
[Anderson 1972] J. A. Anderson. "A Simple Neural Network Generating an Interactive Memory." *Mathematical Biosciences*, Vol. 14, pages 197-220. 1972.
[ARC a] MarsMap - VR for Mars Pathfinder. NASA Ames Research Center. World Wide Web address: http://img.arc.nasa.gov/~blackmon/MarsMap/VR_4_MarsPathfinder.html.
[ARC b] Viz – 3D Visualization. NASA Ames Research Center. World Wide Web address: http://img.arc.nasa.gov/~nguyen/viz/doc/html/
[Arkin 1987] R. C. Arkin. "Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behavior." **Proceedings of the International Conference on Robotics and Automation, Raleigh, NC**. March 1987.
[Arkin 1989] R. C. Arkin. "Motor Schema-Based Mobile Robot Navigation." *International Journal of Robotics Research*, Vol. 8(4), pages 92-112, August 1989.
[Arkin and Bekey 1997] R. C. Arkin and G. A. Bekey. **Robot Colonies**. Kluwer Academic Publishers, Dordrecht, The Netherlands. 1997.
[Asada and Slotine 1986] H. Asada and J.-J. E. Slotine. **Robot Analysis and Control**. John Wiley & Sons, New York, New York, USA. 1986.
[Backes et al 1998] P. G. Backes, K. S. Tso, and G.K. Tharp. "Mars Pathfinder Mission Internet-Based Operations Using WITS." **Proceedings of the 1998 International Conference on Robotics and Automation, Leuven, Belgium**. IEEE Publishing. May 1998.
[Backes et al 1999] P. G. Backes, G. Rabideau, K. S. Tso, and S. Chien. "Automated Planning and Scheduling for Planetary Rover Distributed Operations." **Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, Michigan**. IEEE Press, Piscataway, New Jersey, USA. May 1999.
[Barnes and Gray 1991] D. Barnes and J. Gray. "Behavioral Synthesis for Co-Operant Mobile Robot Control." **International Conference on Control.** Pages 1135-1140. 1991.
[Beetz and McDermott 1994] M. Beetz and D. McDermott. "Improving Robot Plans During Their Execution." **Proceedings International Conference on Artificial Intelligence Planning Systems**. Pages 3-12. 1994.
[Bellingham and Consi 1990] J. G. Bellingham and T. R. Consi. "State Configured Layered Control." **First Workshop on Mobile Robots for Subsea Environments, Monterey, California**. Pages 75-80. October 1990.
[Ben-Bassat 1988] M. Ben-Bassat. "AITest – A Real Life Expert System for Electronic Troubleshooting." **Proceedings of the 4th Conference on AI Applications**. IEEE Computer Society Press. 1988.
[Blum and Furst 1997] A. Blum and M. Furst. "Fast Planning Through Planning Graph Analysis." *Artificial Intelligence*, Vol. 90, pages 281-300. 1997.
[Blum and Langford 1999] A. Blum and J. Langford. "Probabilistic Planning in the Graphplan Framework." **Proceedings European Conference on Planning**. 1999.

[Bonasso et al 1997] R. P. Bonasso, R. J. Firby, E. Gat, D. Kortenkamp, D. Miller, and M. Slack. "A Proven Three-Tiered Architecture for Programming Autonomous Robots." *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 9(2), 1997.
[Borelly et al 1998] J. J. Borrelly, E. Coste-Maniere, B. Espiau, K. Kapellos, R. Pissard-Gibollet, D. Simon, and N. Turro. "An Integrated and Modular Approach for the Specification, the Validation and the Implementation of Complex Robotics Missions." *International Journal of Robotics Research*, Vol. 17(4), pages 338-359, April 1998.
[Brogan 1991] W. L. Brogan. **Modern Control Theory**, third edition. Prentice Hall, Upper Saddle River, New Jersey, USA. 1991.
[Brooks 1986] R. A. Brooks. "A Robust Layered Control System for Mobile Robot." *IEEE Journal of Robotics and Automation*, Vol. 2(1), pages 14-23, March 1986.
[Brunner et al 1993] B. Brunner, G. Hirzinger, K. Landzettel, and J. Heindl. "Multisensory Shared Autonomy and Tele-Sensor-Programming - Key Issues in the Space Robot Technology Experiment ROTEX." **IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Yokohama, Japan**. IEEE Press, Piscataway, New Jersey, USA. July 1993.
[Busuioc et al 1996] M. Busuioc et. al. "Distributed Intelligent Agents – A Solution for the Managment of Complex Services." **IATA- ECAI Workshop Proceedings**. 1996.
[Castelfranchi 1998] C. Castelfranchi and R. Conte. "Limits of Economic and Strategic Rationality for Agents and MA Systems." *Robotics and Autonomous Systems*, Vol. 24, pages 127-139. 1998.
[CMU a] Distributed Robotic Architectures. Carnegie Mellon University. World Wide Web address: http://www.frc.ri.cmu.edu/projects/dira.
[Chien et al 1997] S. Chien, F. Fisher, E. Lo, H. Mortensen, R. Greeley. "Using Artificial Intelligence Planning to Automate Science Data Analysis for Large Image Databases." **Proceedings Conference on Knowledge Discovery and Data Mining, Newport Beach, CA**. August 1997.
[Chien et al 1998] S. Chien, N. Muscettola, K. Rajan, B. Smith, and G. Rabideau. "Automated Planning and Scheduling for Goal-Based Autonomous Spacecraft." *IEEE Intelligent System*s, Vol. 13(5), pages 50-55. 1998.
[Chien et al 1999] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau. "Integrated Planning and Execution for Autonomous Spacecraft." **Proceedings of the IEEE Aerospace Conference, Aspen, CO**. IEEE Press, Piscataway, New Jersey, USA. March 1999.
[Chien et al 2000] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau. "Using Iterative Repair to Improve the Responsiveness of Planning and Scheduling." **Proceedings International Conference on Artificial Intelligence Planning and Scheduling, Breckenridge, CO**. April 2000.
[Coppin et al 1999] P. Coppin, A. Morrissey, M. Wagner, M. Vincent, and G. Thomas. "Information Interaction for Public Telerobotic Exploration." Presented at the Workshop on Current Challenges in Internet Robotics, International Conference on Robotics and Automation. May 15, 1999.
[Coste-Maniere and Turro 1997] E. Coste-Maniere and N. Turro. "The MAESTRO Language and its Environment: Specification, Validation and Control of Robotic Missions." **International Conference on Intelligent Robots and Systems, (IROS), Grenoble, France**. Pages 836-841. September 1997.
[Cuo et al 1997] Y. U. Cuo, A. S. Fukanaga, and A. B. Kahng. "Cooperative Mobile Robotics: Antecedents and Directions." **Autonomous Robots**, R. C Arkin and G. A. Bekely (editors), Vol. 4(1). March 1997.
[Currie and Tate 1991] K. Currie and A. Tate. "O-Plan: The Open Planning Architecture." *Artificial Intelligence*, Vol. 52, pages 49-86. 1991.

[DeCoste 2000] D. DeCoste. "Learning Envelopes for Fault Detection and State Summarization." Presented at the IEEE Aerospace Conference. March 2000.
[DeCoste 1997] D. DeCoste. "Automated Learning and Monitoring of Limit Functions." Fourth International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS-97), Japan. August 1997.
[Desrochers 1992] A. A. Desrochers (editor). **Intelligent Robotic Systems for Space Exploration**. Kluwer Academic Publishers, Norwell, Massachusetts, USA. 1992.
[Dias and Stentz 1999] M. B. Dias and A. X Stentz. "A Free-Market Architecture for Coordinating Multiple Robots." Carnegie Mellon University Technical Report CMU-RI-TR-99-42, Pittsburgh, Pennsylvania, USA. December 1999.
[Dickmanns et al 1990] E. D. Dickmanns, B. Mysliwetz, and T. Christians. "An Integrated Spaito-Temporal Approach to Automatic Visual Guidance of Autonomous Vehicles." *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20(6). November/December 1990.
[Draper et al 1994] D. Draper, S. Hanks, and D. Weld. "Probabilistic Planning with Information Gathering and Contingent Execution." **Proceedings International Conference on AI Planning Systems**. June 1994.
[Drummond and Bresina 1990] M. Drummond and J. L. Bresina. "Anytime Synthetic Projection: Maximizing the Probability of Goal Satisfaction." **Proceedings National Conference on Artificial Intelligence**. Pages 138-144. 1990.
[Duda et al 1979] R. Duda, H. Gaschnig, and P. Hart. "Model Design in hte PROSPECTOR Consultant System for Mineral Exploration." **Expert Systems in the Micro-Electronic Age**, D. Michie (editor). Edinburgh University Press, Edinburgh, Scotland. 1979.
[Erman and Lesser 1983] L. E. Erman and V. R. Lesser. "The Hearsay II Speech Understanding System: A Turotial." **Proceedings of IFAC Symposium**, V. M. Ponomaryov (editor). Leningrad, USSR. 1983.
[Erol et al 1995] K. Erol, D. Nau, and V. S. Subrahmanian. "Complexity, Decidability and Undecidability Results for Domain-Independent Planning." *Artificial Intelligence*, Vol. 76(1-2), pages 75-88. 1995.
[Faratin et al 1998] P. Faratin, C. Sierra, and N. J. Jennings. "Negotiation Decision Functions for Autonomous Agents." *Robotics and Autonomous Systems*, Vol. 24, pages 159-182. 1998.
[Feigenbaum et al 1971] E. Feigenbaum, G. Buchnan, and J. Lederberg. "Generality and Problem Solving: A Case Study Using the DENDRAL Program." *Machine Intelligence*, Vol. 6, pages 165-190. 1971.
[Fennema et al 1990] C. Fennema, A. Hanson, E. Riseman, J. R. Beveridge, and R. Kumar. "Model-Directed Mobile Robot Navigation." *IEEE Transactions on Systems, Man and Cybernetics,* Vol. 20(6). November/December 1990.
[Fesq and Stephan 1989] L. M. Fesq and A. Stephan. "On-Board Fault Management Using Modeling Techniques." **Proceedings of IEEE 24th Intersociety Energy Conservation Engineering Conference (IECEC), Washington DC**. IEEE Press, Piscataway, New Jersey, USA. August 1989.
[Fikes and Nilsson 1972] R. Fikes and N. Nilsson. "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving." *Artificial Intelligence*, Vol. 2(3-4), pages 189-208. 1972.
[Firby 1987] R. J. Firby. "An Investigation Into Reactive Planning in Complex Domains." **Proceedings National Conference on Artificial Intelligence, Seattle, Washington**. Pages 202-206. July 1987.
[Fuchs et al 1990] J. J. Fuchs, A. Gasquet, B. Olalainty, and K. Currie. "PlanERS-1: An Expert Planning System for Generating Spacecraft Missions." **International Conference on Expert Planning Systems, Brighton, UK**. IEE, London, UK. Pages 70-75. 1990.

[Fukuda and Nakagawa 1990] T. Fukuda and S. Nakagawa. "Analysis and Evaluation of Cellular Robotics (CEBOT) as a Distributed Intelligent System by Communication Amount." **IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. IEEE Press, Piscataway, New Jersey, USA. Pages 827-834. 1990.
[Gat 1992] E. Gat. "Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots." **Tenth National Conference on Artificial Intelligence, San Jose, California**. July 1992.
[Gat 1996] E. Gat. "ESL: A Language for Supporting Robust Plan Execution in Embedded Autonomous Agents." **AAAI Fall Symposium on Plan Execution**. AAAI Press, Menlo Park, California, USA. 1996.
[Gennery 1999] D. Gennery. "Traversability Analysis and Path Planning for a Planetary Rover." *Autonomous Robots*. 1999.
[Georgeff 1987] M. Georgeff and A. Lansky. "Reactive Reasoning and Planning." **National Conference on Artificial Intelligence, Seattle, Washington**. Pages 972-978. July 1987.
[Glombek et al 1997] M. P. Golombek, R. A. Cook, T. Economou, W. M. Folkner, A. F. C. Haldemann, P. H. Kallemeyn, J. M. Knudsen, R. M. Manning, H. J. Moore, T. J. Parker, R. Rieder, J. T. Schofield, P. H. Smith, and R. M. Vaughan. "Overview of the Mars Pathfinder Mission and Assessment of Landing Site Predictions." *Science*, Vol. 278, pages 1743-1748. December 1997.
[Grossberg 1976] S. Grossberg. "Adaptive Pattern Classification and Universal Recoding: I. Parallel Development and Coding of Neural Feature Detectors." *Biological Cybernetics*, Vol. 23, pages 121-134. 1976.
[Hagan et al 1996] M. T. Hagan, H. B. Demuth, and M. Beale. **Neural Network Deisgn**. PWS Publishing Company, Boston, Massachusetts, USA. 1996.
[Hammond 1989] K. Hammond. **Case-Based Planning: Viewing Planning as a Memory Task**. Academic Press, New York, New York, USA. 1988.
[Hanks and Weld 1992] S. Hanks and D. Weld. "Systematic Adaptation for Case-Based Planning." **Proceedings International Conference on Artificial Intelligence Planning Systems, College Park, MD**. Pages 96-105. June 1992.
[Hebert 1997] M. H. Hebert. "SMARTY: Point-Based Range Processing for Autonomous Driving." **Intelligent Unmanned Ground Vehicle**, M. H. Hebert, C. Thorpe, and A. Stentz (editors), Kluwer Academic Publishers, Norwell, Massachusetts, USA. 1997.
[Hine et al 1995] B. Hine, P. Hontalas, T. Fong, L. Piguet, E. Nygren, and A. Kline. "VEVI: A Virtual Environment Teleroperations Interface for Planetary Exploration." **SAE Twenty-Fifth International Conference on Environmental Systems, San Diego, California**. July 1995.
[Holldobler and Wilson 1990] B. Holldobler and E. Wilson. **The Ants**. Belkanp Press of Hardvard Press, Cambridge, Massachusetts, USA. 1990.
[Hopfield 1982] J. J. Hopfield. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of the National Academy of Sciences*, Vol. 79, pages 2554-2558. 1982.
[Isermann and Balle 1997] R. Isermann and P. Balle. "Trends in the Application of Model-Based Fault Detection and Diagnosis of Technical Processes." **Control Engineering Practice**. Vol. 5, pages 709-719. 1997.
[Jin et al 1994] K. Jin, P. Liang, and G. Beni. "Stability of Synchronized Distributed Control of Discrete Swarm Structures." **IEEE ICRA**. IEEE Press, Piscataway, New Jersey, USA. Pages 1033-1038. 1994.
[Johnson and Adorf 1992] M. D. Johnson and H. M. Adorf. "Scheduling with Neural Networks: The Case of the Hubble Space Telescope." *Computers & Operations Research*, Vol. 19(3-4), pages 209-240. 1992.

[Kaelbling et al 1998] L. Pack Kaelbling, M. L. Littman, and A. R. Cassandra. "Planning and Acting in Partially Observable Stochastic Domains." *Artificial Intelligence*, Vol. 101. 1998.
[Kautz and Selman 1996] H. Kautz and B. Selman. "Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search." **Proceedings National Conference on Artificial Intelligence**. 1996.
[Kelly 1995] A. Kelly. "An Intelligent Predictive Control Approach to the High Speed Cross Country Autonomous Navigation Problem." Ph.D Thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. 1995.
[Kerpedjiev and Roth 2000] S. Kerpedjiev, and S. F. Roth. "Mapping Communicative Goals into Conceptual Tasks to Generate Graphics in Discourse." **Proceedings of Intelligent User Interfaces (IUI '00)**. January 2000.
[Kohonen 1982] T. Kohonen. "Correlation Matrix Memories." *IEEE Transactions on Computers*, Vol. 21, pages 353-359. 1982.
[Korf 1987] R. E. Korf. "Real-Time Heuristic Search: First Results." **Proceedings of the Sixth National Conference on Artificial Intelligence**. July 1987.
[Kosko 1992] B. Kosko. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice Hall, Englewood Cliffs, New Jersey, USA. 1992.
[Kuester and Corde Lane 1995] S. P. Kuester and J. Corde Lane. "Interface Design Issues of the Ranger Telerobotic Flight Experiment." **ICES '95 Proceedings**. 1995.
[Kushmerick et al 1994] N. Kushmerick, S. Hanks, and D. Weld. "An Algorithm for Probabilistic Least-Commitment Planning." **Proceedings National Conference on Artificial Intelligence, Seattle, WA**. July 1994.
[Laubach et al 1998] S. Laubach, J. Burdick, and L. Matthies. "Autonomous Path-Planning for the Rocky 7 Prototype Microrover." **Proceedings of the IEEE International Conference on Robotics and Automation**. IEEE Press, Piscataway, New Jersey, USA. 1998.
[Lee and Xu 1996] C. Lee and Y. Xu. "Online, Interactive Learning of Gestures for Human/Robot Interfaces." **IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota**, Vol. 4, pages 2982-2987. IEEE Press, Piscataway, New Jersey, USA. 1996.
[Lowry et al 1997] M. Lowry, K. Havelund, and J. Penix. "Verification and validation of AI Systems That Control Deep-Space Spacecraft." **Tenth International Symposium on Methodologies for Intelligent Systems in Springer-Verlag Lecture Notes in Artificial Intelligence**, Vol. 1325. October 1997.
[Lumelsky and Stepanov 1986] V. J. Lumelsky and A. A. Stepanov. "Dynamic Path Planning for a Mobile Automaton with Limited Information on the Environment." *IEEE Transactions on Automatic Control*, Vol. AC-31(11). November 1986.
[Marciniak 1994] J. J. Marciniak. **Encyclopedia of Software Engineering**. John Wiley & Sons, Chichester, England. 1994.
[Matin and Oxman 1988] J. Matin and S. W. Oxman. **Building Expert Systems**. Prentice Hall, Upper Saddle River, New Jersey, USA. 1988.
[Matthies et al 1995] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe, and T. Litwin. "Mars Microrover Navigation: Performance Evaluation and Enhancement." *Autonomous Robots*, Vol. 2, pages 291-311. 1995.
[McAllister and Rosenblitt 1991] D. McAllester and D. Rosenblitt. "Systematic Nonlinear Planning." **Proceedings National Conference on Artificial Intelligence**. 1991.
[McCulloch and Pitts 1943] W. McCulloch and W. Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics*, Vol. 5, pages 115-113. 1943.
[McDermott 1982] J. McDermott. "R1: A Rule-Based Configurator of Computer Systems." *Artificial Intelligence*, Vol. 19, pages 39-88. 1982.

[Muscettola et al 1992] N. Muscettola, S. F. Smith, A. Cesta, and D. D'Aloisi. "Coordinating Space Telescope Operations in an Integrated Planning and Scheduling Framework." *IEEE Control Systems*, Vol. 12(1). February 1992.
[Muscettola et al 1997] N. Muscettola, P. Nayak, B. Pell, and B. Williams. "The New Millennium Remote Agent: To Boldly Go Where No AI System Has Gone Before." **Fifteenth International Joint Conference on Artificial Intelligence**. 1997.
[Musliner et al 1993] D. Musliner, E. Durfee, and K. Shin. "CIRCA: A Cooperative Intelligent Real-Time Control Architecture." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23(6), 1993.
[Narendra and Parthasarathy 1990] K. S. Narendra and K. Parthasarathy. "Identification and Control of Dynamical Systems Using Neural Networks." *IEEE Transactions on Neural Networks*, Vol. 1(1). March 1990.
[Neuman 1995] P. G. Neuman. **Computer Related Risks**. Addison-Wesely, Reading, Massachusetts, USA. 1995
[Nii and Aiello 1979] P. Nii and N. Aiello. "AGE (Attempt to Generalize): A Knowledge-Based Program for Building Knowledge-Based Programs." **Proceedings of the IJCAI**. 1979.
[Parr and Russell 1995] R. Parr and S. Russell. "Approximating Optimal Policies for Partially Observable Stochastic Domains." **Proceedings of International Joint Conference on Artificial Intelligence**. 1995.
[Pecheur and Simmons 2000] C. Pecheur and R. Simmons. "From Livingstone to SMV: Formal Verification for Autonomous Spacecrafts." Presented at the First Goddard Workshop on Formal Approaches to Agent-Based Systems, April 5-7, 2000. To appear in **Lecture Notes of Computer Science**, Springer Verlag, New York, New York. 2000.
[Penberthy and Weld 1992] J. S. Penberthy and D. Weld. "UCPOP: A Sound, Complete, Partial-Order Planner for ADL." **Proceedings of KR-92, Cambridge, MA,** pages 103-114. October 1992.
[Peot and Smith 1992] M. Peot and D. Smith. "Conditional Nonlinear Planning." **Proceedings International Conference on AI Planning Systems, College Park, Maryland**. 1992.
[Pirzadeh and Snyder 1990] A. Pirzadeh and W. Snyder. "A Unified Solution to Coverage and Search in Explored and Unexplored Terrains Using Indirect Control." **Proceedings of the IEEE International Conference on Robotics and Automation**. IEEE Press, Piscataway, New Jersey, USA. 1990.
[Pomerleau 1993] D. A. Pomerleau. "Knowledge-Based Training of Artificial Neural Networks for Autonomous Robot Driving." **Robot Learning**, J. H. Connel and S. Mahadevan (editors). Kluwer Academic Publishers, Norwell, Massachusetts, USA. 1993.
[Pruitt 1981] D. G. Pruitt. **Negotiation Behavior**. Academic Press, New York, New York, USA. 1981.
[Rabideau et al 1999] G. Rabideau, R. Knight, S. Chien, A. Fukunaga, and A. Govindjee. "Iterative Repair Planning for Spacecraft Operations in the ASPEN System." **Proceedings International Symposium on Artificial Intelligence Robotics and Automation in Space (ISAIRAS), Noordwijk, The Netherlands**. June 1999.
[Rolston 1988] D. W. Rolston. **Principles of Artificial Intelligence and Expert Systems Development**. McGraw-Hill, New York, New York, USA. 1988.
[Rosenblat 1958] F. Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." *Psychological Review*, Vol. 65, pages 386-408. 1958.
[Rosenblatt and Thorpe 1995] J. Rosenblatt and C. Thorpe. "Combining Multiple Goals in a Behavior-Based Architecture." **International Conference on Intelligent Robots and Systems (IROS), Pittsburgh, Pennsylvania**. August 1995.

[Roumeliotis et al 1998a] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey. "Sensor Fault Detection and Identification in a Mobile Robot." **IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, Canada**. IEEE Press, Piscataway, New Jersey, USA. 1998.
[Roumeliotis et al 1998b] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey. "Fault Detection and Identification in a Mobile Robot Using Multiple-Model Estimation." **Proceedings of the IEEE International Conference on Robotics and Automation**. Pages 2223-2228. May 1998.
[Roy et al 1999] N. Roy, W. Burgard, D. Fox, and S. Thrun. "Coastal Navigation -- Mobile Robot Navigation with Uncertainty in Dynamic Environments." **Proceedings International Conference on Robotics and Automation, Detroit**. May 1999.
[Rumelhart and McClelland 1986] D. E. Rumelhart and J. L. McClelland (editors). **Parallel Distributed Processing Explorations in the Microstructure of Cognition**, Volume 1. MIT Press, Cambridge, Massachusetts, USA. 1986.
[Russell and Norvig 1995] S. Russell and P. Norvig. **Artificial Intelligence: A Modern Approach**. Prentice Hall, Upper Saddle River, New Jersey, USA. 1995
[Schneider et al 1996] M. Schneider, A. Kandel, G. Lnagholz, and G. Chew. **Fuzzy Expert Systems**. John Wiley & Sons, Chichester, England. 1996.
[Schneider et al 1998] S. Schneider, V. Chen, G. Pardo-Castellote, and H. Wang. "ControlShell: A Software Architecture for Complex Electro-Mechanical Systems." *International Journal of Robotics Research*. Spring 1998.
[Schoppers 1987] M. Schoppers. "Universal Plans for Reactive Robots in Unpredictable Environments." **Proceedings 10th IJCAI**. 1987.
[Seraji 1999] H. Seraji, "Traversability Index: A new concept for planetary rovers." **Proceedings of the IEEE International Conference on Robotics and Automation**. IEEE Press, Piscataway, New Jersey, USA. 1999.
[Shah 1988] R. P. Shah. "JET-X: Jet Engine Troubleshooting Expert System." **Proceedings of the International Workshop on Artificial Intelligence for Industrial Applications**. IEEE Press, Piscataway, New Jersey, USA. 1988.
[Sheridan 1992] T. B. Sheridan. **Telerobotics, Automation and Human Supervisory Control**. MIT Press, Boston, MA, USA. Page 107. 1992.
[Shortliffe 1976] E. H. Shortliffe. **Computer-Based Medical Consultation: MYCIN**. Elsevier North, Holland. 1976.
[Simmons 1994] R. Simmons. "Structured Control for Autonomous Robots." *IEEE Transactions on Robotics and Automation*, Vol. 10(1), pages 34-43. February 1994.
[Simmons 1988] R. Simmons. "A Theory of Debugging Plans and Interpretations." **Proceedings National Conference on Artificial Intelligence, St. Paul, MN**. August 1988.
[Simmons and Apfelbaum 1998] R. Simmons and D. Apfelbaum. "A Task Description Language for Robot Control." **International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada**. October 1998.
[Singh et al 2000] S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja, and K. Schwehr. "Recent Progress in Local and Global Traversability for Planetary Rovers." **Proceedings of ICRA 2000, San Francisco**. April 2000.
[Tanaka and Sueda 1988] T. Tanaka and N. Sueda. "Knowledge Acquisition in Image Processing – Expert System EXPLAIN." **Proceedings of the International Workshop on AI for Industrial Applications**. IEEE Press, Piscataway, New Jersey, USA. 1988
[Thrun 1995] S. Thrun. "An Approach to Learning Mobile Robot Navigation." *Robotics and Autonomous Systems*, Vol. 15, pages 301-319. 1995.

[Veloso and Carbonell 1993] M. Veloso and J. Carbonell. "Derivational Analogy in PRODIGY: Automating Case Acquisition." *Machine Learning*, Vol. 10, pages 249-278. 1993.
[Veloso 1995] M. Veloso, J. Carbonell, M. Perez, D. Borrajo, E. Fink, and J. Blythe. "Integrating Planning and Learning: The Prodigy Architecture." *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 7(1), pages 81-120. 1995.
[von Firsch 1967] K. Von Frisch. **The Dance Language and Orientation of Bees**. Belknap Press of Harvard University Press, Cambridge, Massachusetts, USA. Pages 57-63, 236-256. 1967.
[Voyles and Khosla 1999] R. M. Voyles, and P. K. Khosla. "Gesture-Based Programming: A Preliminary Demonstration." **Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, Michigan**. May 1999.
[Weld 1994] D. Weld. "An Introduction to Least-Commitment Planning." *Artificial Intelligence Magazine*, pages 27-61. Winter 1994.
[Weld 1999] D. Weld. "Recent Advances in AI Planning." *Artificial Intelligence Magazine*, 1999.
[Wellman 1995] M. P. Wellman. "Market-Oriented Computational Programming: Some Early Lessons," Market-Based Control: **A Paradigm for Distributed Resource Allocation**, S. H. Clearwater (editor). World Scientific, River Edge, New Jersey, USA. 1995.
[Widrow and Hoff 1960] B. Widrow and M. E. Hoff. "Adaptive Switching Circuits." **1960 IRE WESCON Convention Record, New York IRE**, Part 4. Pages 96-104. 1960.
[Wilkins 1988] D. E. Wilkins. **Practical Planning: Extending the AI Planning Paradigm**. Morgan Kaufmann, San Mateo CA, USA. 1988.
[Williams and Nayak 1996] B. C. Williams and P. P. Nayak. "A Model-Based Approach to Reactive Self-Configuring Systems." **Proceedings of AAAI**. 1996.
[Williams and Nayak 1997] B. C. Williams and P. P. Nayak. "A Reactive Planner for a Model-Based Executive." **Proceedings of IJCAI**. 1997.

Appendix A: Table of Acronyms

AI	Artificial Intelligence
APL	Applied Physics Laboratory (Johns Hopkins University)
ARC	Ames Research Center (NASA)
ASI	Agenzia Spaziale Italiana (Italian Space Agency)
BNSC	British National Space Centre
CMU	Carnegie Mellon University
CSA	Canadian Space Agency
DLR	Deutsches Zentrum für Luft-und Raumfahrt (German Aerospace Center)
DSN	Deep Space Network
ESA	European Space Agency
ESO	European Southern Observatory
EVA	Extra-vehicular activity (activities conducted outside a spacecraft, in space)
GSFC	Goddard Space Flight Center
GSOC	German Space Operations Center
GUI	Graphical user interface
HMM	Hidden Markov Model
HQ	(NASA HQ) NASA Headquarters
IVA	Intra-vehicular activity (activities conducted inside a spacecraft)
IKI	Space Research Institute, Russia
ISAS	Institute of Space and Aeronautical Science (Japan)
ISSI	International Space Science Institute
JHU	Johns Hopkins University
JPL	Jet Propulsion Lab (NASA and California Institute of Technology)
JSC	Johnson Space Center

KSC	Kennedy Space Center
LMCO	Lockheed Martin Corporation
LRC	Langley Research Center
MCD	McDonnell Douglas
MDP	Markov Decision Process
MIT	Massachusetts Institute of Technology
MSFS	Marshall Space Flight Center
NAOJ	National Astronomical Observatory of Japan
NASA	National Aeronautics and Space Administration
NASDA	Japanese Space Agency
NIST	National Institute of Standards and Technology
POMDP	Partially Observable Markov Decision Process
SAIC	Science Applications International Corporation
SSL	Space Systems Laboratory (University of Maryland)
SWRI	Southwest Research Institute
UC	University of California (UC Berkeley)
UCO	University of Colorado (Boulder, Colorado Springs)
UMD	University of Maryland
UWA	University of Washington

Appendix B: Table of International Space Systems

The table included here is a list of all space missions, mission concepts, technologies, and tools discussed in the Systems section of this document. Each system is listed under its formal name and cross-listed with common names and/or acronyms. Information provided on each system includes the agencies responsible for the project, the dates of the project, a brief description of the project, and selected World Wide Web references for the project.

The meaning of the acronyms listed under the “Agency” heading can be found in Appendix A. Each system is listed with the section (Deep Space/Heliocentric, Planetary, Orbital, Human Assistance, Tools) and the subsection (Autonomous, Semi-Autonomous, Teleoperated, Directly Controlled) in which it appears in the Systems chapter of this document.

Name	Agencies	Section	Page	Dates	Description and References
ACE	NASA GSFC Cal Tech JHU APL	Deep Space, Directly Controlled		1997	See Advanced Composition Explorer
Adaptive Problem Solving (APS)	JPL	Tools, Autonomous		1993-	A scheduling system that is heurstic based and self-customizing for DSN pointing http://www-aig.jpl.nasa.gov/public/planning/aps/
Advanced Composition Explorer (ACE)	NASA GSFC Cal Tech JHU APL	Deep Space, Directly Controlled		1991	A mission to determine composition of intersteller and interplanetary dust and the solar corona; L1 orbit http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?97-045A http://helios.gsfc.nasa.gov/ace/ace.html http://F5www.srl.caltech.edu/ACE
AERCam	NASA JSC	Human Assistance, Semi-Autonomous		1997	See Autonomous EVA Robotic Camera
Aerobot	JPL	Planetary, Semi-Autonomous		2003	Teleoperated balloons with autonomous elevation control using path prediction and planning; project since 1993 http://robotics.jpl.nasa.gov/tasks/aerobot/homepage.html http://telerobotics.jpl.nasa.gov/aerobot/
Ambler	CMU NASA HQ	Planetary, Autonomous		1987-1990	Teleoperated large rover prototype for planetary rovers; autonomous gait planning http://ranier.oact.hq.nasa.gov/telerobotics_page/Technologies/0710.html http://ranier.oact.hq.nasa.gov/telerobotics_page/Technologies/0302.html
APA	NASA ARC	Tools, Semi-Autonomous		1993-	See Associate Principal Astronomer
APEX	NASA JPL	Planetary, Teleoperated		2001	See Mars Surveyor 2001
Apollo	NASA KSC	Planetary, Directly Controlled		1968-1972	Manned and unmanned lunar orbiting and landing missions, including a manned rover http://nssdc.gsfc.nasa.gov/planetary/lunar/apollo.html http://cass.jsc.nasa.gov/expmoon/apollo_landings.html
APS	JPL	Tools, Autonomous		1993-	See Adaptive Problem Solving
ASPEN	JPL	Tools, Semi-Autonomous		1998-	See Automated Scheduling and Planning Environment
ASPERA-3	ESA	Planetary, Teleoperated		2003	See Mars Express
ASPIRE	JPL	Tools, Autonomous		2003	See Autonomous Small Planet In Situ Reaction to Events
ASSAP	JPL	Tools, Autonomous		1996-	See Autonomous Serendipitous Science Acquisition for Planets
Associate Principal Astronomer (APA)	NASA ARC	Tools, Semi-Autonomous		1993-	A system for the control and scheduling of telescopes to accommodate multiple requests http://ic-www.arc.nasa.gov/ic/projects/xfr/index.html
Athena	NASA JPL Cornell	Planetary, Semi-Autonomous		2003, 2005	See Mars Surveyor 2003/2005
Athena Precursor Experiment (APEX)	NASA JPL	Planetary, Teleoperated		2001	See Mars Surveyor 2001
Automated Scheduling and Planning Environment (ASPEN)	JPL	Tools, Semi-Autonomous		1998-	A system for autonomous scheduling and planning that uses the WITS interface; generates and optimizes command sequences http://www-aig.jpl.nasa.gov/public/planning/rover/ http://www-aig.jpl.nasa.gov/public/planning/aspen/aspen_index.html
AutoNav	NASA ARC JPL	Tools, Autonomous		1998	An autonomous celestial localization program for Deep Space 1 http://nmp.jpl.nasa.gov/ds1/tech/autonav.html
Autonomous EVA Robotic Camera (AerCam, Sprint)	NASA JSC	Human Assistance, Semi-Autonomous		1997	A free-flying robotic camera for shuttle/space station EVA; teleoperated versions have flown, autonomy is in development http://spaceflight.nasa.gov/station/assembly/sprint/ http://tommy.jsc.nasa.gov/projects/Sprint/
Autonomous Satellite Detection	JPL SWRI	Tools, Autonomous		1996-	A tool for autonomous recognition of natural satellites and planetary bodies; tested on Galileo images http://www-aig.jpl.nasa.gov/mls/onboard/sathome.html
Autonomous Serendipitous Science Acquisition for Planets (ASSAP)	JPL	Tools, Autonomous		1996-	A tool for adaptive autonomous science data processing with data recognition and acquisition for the New Millenium Program http://www-aig.jpl.nasa.gov/public/mls/onboard/onboard_home.html
Autonomous Small Planet In Situ Reaction to Events (ASPIRE)	JPL	Tools, Autonomous		2003	A system for in-situ detection, capture and analysis of short term comet events, which includes adaptive planning to respond to rapid occurrences; currently in use on a testbed, mission under consideration http://www-aig.jpl.nasa.gov/public/mls/aspire/aspire.html
BAT	UMD SSL	Orbital, Teleoperated		1985	See Beam Assembly Teleoperator
Beam Assembly Teleoperator (BAT)	UMD SSL	Orbital, Teleoperated		1985	A mobile robot for space construction, neutral buoyancy testbed and shuttle mission http://www.ssl.umd.edu/homepage/Projects/BAT/bat.html

Name	Agencies	Section	Page	Dates	Description and References
BepiColombo	ESA	Planetary, Directly Controlled		2009	Mercury orbital exploration for scientific data collection http://www.estec.esa.nl/spdwww/future/html/meo2.htm http://nssdc.gsfc.nasa.gov/planetary/prop_missions.html#isas_mercury_orb
Big Signal Antarctica 2000	CMU	Tools, Semi-Autonomous		2000	A visual user interface to study data collected by the Robotic Antarctic Meteorite Search. Http://www.bigsignal.net
Canadarm	Spar	Human Assistance, Directly Controlled		1975-	See Shuttle Remote Manipulator System
CASPER	JPL	Tools, Autonomous		1998-	See Continuous Activity Scheculing Planning Execution and Replanning
Cassini	NASA KSC JPL ESA ASI	Planetary, Semi-Autonomous		1997	Saturn and Titan probe; uses the COSMO heuristic planner to aid in planning for visiting science targets http://www.jpl.nasa.gov/cassini/ http://webserver.gsfc.nasa.gov/java/cassini.html http://ic-www.arc.nasa.gov/ic/projects/ops-sked/Cassini/www-cassini.html
Cassini Operations and Science Mission Optimizer (COSMO)	NASA ARC	Tools, Semi-Autonomous		1997	Heuristic planner with a graphical interface that assists in planning without the user's knowledge of mission and engineering constraints; used on Cassini http://ic-www.arc.nasa.gov/ic/projects/ops-sked/Cassini/www-cassini.html#COSMO
Charlotte	NASA MCD	Human Assistance, Teleoperated		1995-1996	robotic spider suspended on cables for shuttle bay operations http://www.ksc.nasa.gov/shuttle/missions/sts-63/sts-63-press-kit.txt http://www.ksc.nasa.gov/shuttle/missions/sts-63/mission-sts-63.html http://www.boeing.com/assocproducts/mdip/charlot.htm
Citizen Explorer (CX-1)	JPL UCO	Orbital, Semi-Autonomous		2000	Earth orbiter for studying weather patterns which uses ASPEN generated command sequences which consider power and engineering requirements http://citizen-explorer.colorado.edu/ http://www-aig.jpl.nasa.gov/public/planning/cx1
Clementine (Deep Space Probe Scienc Experiment/ DSPSE)	NASA GSFC JPL NRL	Planetary, Directly Controlled		1994	Lunar orbiter for radar mapping most of the lunar surface http://nssdc.gsfc.nasa.gov/planetary/clementine.html http://cass.jsc.nasa.gov:80/expmoon/clementine/clementine.html http://setas-www.larc.nasa.gov/CLEM/dspse.html
Comet Nucleus Tour (CONTOUR)	NASA GSFC JHU APL Cornell	Deep Space, Directly Controlled		2002	A series of flybys of three comets to investigate comet composition http://www.contour2002.org http://webserver.gsfc.nasa.gov/java/contour.html
Continuous Activity Scheduling Planning Execution and Replanning (CASPER)	JPL	Tools, Autonomous		1998-	A tool for planning with incremental replanning, "iterative repair," which uses the ASPEN planner http://www-aig.jpl.nasa.gov/planning/casper
CONTOUR	NASA GSFC JHU APL Cornell	Deep Space, Directly Controlled		2002	See Comet Nucleus Tour
CX-1	JPL UCO	Orbital, Semi-Autonomous		2000	See Citizen Explorer 1
Dante	CMU NASA HQ	Planetary, Semi-Autonomous		1992-1994	A series of walking robots as planetary prototypes for volcano exploration http://img.arc.nasa.gov/dante/dante.html
DARTS Shell	JPL	Tools, Semi-Autonomous		1997-	A dynamic spacecraft simulator for mission modelling http://dshell.jpl.nasa.gov/dshell.html
Deep Impact	UMD JPL	Deep Space Semi-Autonomous		2004	Craft to impact with comet Tempel to investigate the compisition and character of a comet's core; autonomous E71selection of impact site on comet's sunward side http://www.ss.astro.umd.edu/deepimpact
Deep Space 1 (DS-1)	NASA ARC JPL	Deep Space, Autonomous		1998	A mission to flyby asteroids and comets to determine composition; uses a remote agent for localization and course correction http://www.jpl.nasa.gov/ds1news/
Deep Space 2 (DS-2)	JPL LMCO NASA ARC	Planetary, Autonomous		1999	Mars penetrators launched from the Mars Polar Lander during descent for water detection: LOST http://nmp.jpl.nasa.gov/ds2/
Deep Space 3 (DS-3)	NASA	Deep Space Semi-Autonomous		2003	See Space Technology 3
Deep Space 5 (DS-5)	NASA GSFC	Deep Space Semi-Autonomous		2003	See Space Technology 5
Deep Space Probe Science Exerpiment (DSPSE)	NASA GSFC JPL NRL	Planetary, Directly Controlled		1994	See Clementine
DIRA	CMU NASA JSC NIST	Tools, Autonomous		1999-	See Distributed Robotic Agents
Distributed Robotic Agents (DIRA)	CMU NASA JSC NIST	Tools, Autonomous		1999-	A tool for the visualization and design of mutliple robotic agent systems http://www.frc.ri.cmu.edu/projects/dira http://ww.wisd.cme.nist.gov/projects/robocrane
DS-1	NASA ARC JPL	Deep Space, Autonomous		1998	See Deep Space 1

Name	Agencies	Section	Page	Dates	Description and References
DS-2	JPL LMCO NASA ARC	Planetary, Autonomous		1999	See Deep Space 2
DS-3	NASA	Deep Space Semi-Autonomous		2003	See Space Technology 3
DS-5	NASA GSFC	Deep Space Semi-Autonomous		2003	See Space Technology 5
DSPSE	NASA GSFC JPL NRL	Planetary, Directly Controlled		1994	See Clementine
Earth Observing-1 (EO-1)	NASA GSFC	Orbital, Semi-Autonomous		2000	A constellation of satellites for Earth science with autonomous planning, execution, and calibration of satellite constellation formation using fuzzy logic http://eo1.gsfc.nasa.gov
Engineering Test Satellite 7 (ETS 7)	NASDA	Orbital, Semi-Autonomous		1997-	A satellite with autonomous docking capability http://yyy.tksc.nasda.go.jp/Home/Projects/ETS-VII/index_e.html http://www.spacer.com/spacenet/text/ets7-b.html
Enigma	NASA JSC	Tools, Semi-Autonomous		1988-	A visualization tool for task simulation and system design. http://tommy.jsc.nasa.gov
Envision	Deneb	Tools, Semi-Autonomous		1997-	A 3-D visualization tool for designing, verifying, and rapid prototyping of systems http://www.deneb.com
EO-1	NASA GSFC	Orbital, Semi-Autonomous		2000	See Earth Observing-1
ESS	DLR	Orbital, Semi-Autonomous		1997-	See Exerpimental Servicing Satellite
ETS 7	NASDA	Orbital, Semi-Autonomous		1999	See Engineering Test Satellite 7
Europa Orbiter	JPL	Planetary, Directly Controlled		2003	An orbital explorer to investigate the presence/absence of subsurface oceans on Europa http://www.jpl.nasa.gov/ice_fire//europao.htm
EUVE	NASA UC Berkeley	Orbital, Semi-Autonomous		1992-	See Extreme UltraViolet Explorer
Experimental Servicing Satellite (ESS)	DLR	Orbital, Semi-Autonomous		1994-	A robotic manipulator for satellite repair with autonomous satellite capture; a testbed is in operation http://www.robotic.dlr.de/VISION/Projects/Ess/ess.html http://www.robotic.dlr.de/TELEROBOTICS/ess.html
Extreme UltraViolet Explorer (EUVE)	NASA UC Berkeley	Orbital, Semi-Autonomous		1992-	An Earth-orbiting UV spectrometer with autonomous generation of configuration/orientation sequences to satisfy specified goals using Spike http://ic-www.arc.nasa.gov/ic/projects/ops-sked/Euve/www-euve.html http://www.cea.berkeley.edu/~pubinfo/html/EUVE.html
FIDO	JPL	Planetary, Autonomous		1998-	See Field Integrated Design & Operations Rover
Field Integrated Design & Operations Rover (FIDO)	JPL	Planetary, Autonomous		1998-	Rover testbed for planetary rovers using autonomous local navigation See also: Long Range Science Rover http://robotics.jpl.nasa.gov/tasks/etrover/homepage.html
Galileo	NASA HQ JPL	Deep Space, Directly Controlled		1989-2000	A flyby mission to Jupiter, Europa and Io for imaging and science data http://www.jpl.nasa.gov/galileo/ http://webserver.gsfc.nasa.gov/java/galileo.html
Giotto	ESA	Deep Space, Directly Controlled		1985-1999	A flyby mission of Halley's comet for scientific observation and imaging http://sci.esa.int/giotto
Hagoromo	ISAS	Planetary, Directly Controlled		1980	See Muses-A
Hiten	ISAS	Planetary, Directly Controlled		1980-1993	See Muses-A
HST	NASA	Orbital, Semi-Autonomous		1990-	See Hubble Space Telescope
Hubble Space Telescope (HST)	NASA HQ NASA GSFC NASA MSFC	Orbital, Semi-Autonomous		1990-	The Hubble Space Telescope is an Earth-orbiting platform for deep-space astronomy. It is scheduled by long-term (SPIKE) and short-term (SPSS) autonomous planners http://www.stsci.edu/hst
Huygens	ESA	Planetary, Directly Controlled		1997-2004	An orbiter for Titan exploration, to be launched from Cassini in 2004 http://sci.esa.int/huygens/ http://nssdc.gsfc.nasa.gov/planetary/huygens.html
ICE	NASA GSFC	Deep Space, Directly Controlled		1978-1997	See International Sun-Earth Explorer
Inflatable Rover	JPL	Planetary, Teleoperated		1996-	A series of teleoperated inflatable rovers for planetary surface exploration http://robotics.jpl.nasa.gov/tasks/inflrovers/homepage.html
International Solar Polar Mission	JPL ESA	Deep Space, Directly Controlled		1990	See Ulysses

Name	Agencies	Section	Page	Dates	Description and References
International Sun-Earth Explorer (ISEE, ICE)	NASA GSFC	Deep Space, Directly Controlled		1978-1997	A group of three spacecraft, two in Earth-like heliocentric orbits and a third that moved between the Lagrange point and the earth's orbit for large-baseline measurements of solar wind and the solar-terrestrial relationship http://nssdc.gsfc.nasa.gov/space/isee.html http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?78-079A
Laser Interferometer Space Antenna (LISA)	JPL ESA	Deep Space, Directly Controlled		2010	A three-craft mission operating in heliocentric orbits at 1 AU and separated by five million km to detect gravity waves http://lisa.jpl.nasa.gov/mission/mission.html
LISA	JPL ESA	Deep Space, Directly Controlled		2008	See Laser Interferometer Space Antenna
Long Range Science Rover (Rocky)	JPL	Planetary, Autonomous		1987-	A long-range planetary science rover testbed for autonomous planning and plan optimization using ASPEN http://robotics.jpl.nasa.gov/tasks/rovertech/homepage.html
Luna	IKI	Planetary, Directly Controlled		1959-1976	A series of llunar flylbys, impacts, orbiters, landings and rovers (See Lunokhod) http://nssdc.gsfc.nasa.gov/planetary/lunar/lunarussr.html
Lunar Orbiter	NASA LRC	Planetary, Directly Controlled		1966-1967	A series of lunar orbiters for imaging as a precursor to the Apollo missions http://nssdc.gsfc.nasa.gov/planetary/lunar/lunarorb.html
Lunar Prospector	NASA ARC NASA HQ LMCO	Planetary, Directly Controlled		1998-1999	A lunar orbiter for spectroscopy, particularly in search of water http://lunar.arc.nasa.gov/ http://cass.jsc.nasa.gov/expmoon/prospector/prospector.html http://nssdc.gsfc.nasa.gov/planetary/lunarprosp.html
Lunar-A	ISAS	Planetary, Directly Controlled		2003	A craft with a lunar orbiter and penetrators for gelogical studies of the Moon http://www.isas.ac.jp/e/enterp/missions/lunar-a/cont.html http://www.isas.ac.jp/info/future/lunarA-e.html http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?LUNAR-A
Lunik	IKI	Planetary, Directly Controlled		1959-1976	See Luna
Lunokhod	IKI	Planetary, Teleoperated		1970-1973	A rover for imaging and science on the Moon, part of the Luna series 17 in 1971 and 21 in 1973 http://www.nasm.edu/ceps/etp/tools/tools_rover.html#lunk http://www.friends-partners.org/~mwade/project/luna.htm http://planetscapes.com/solar/eng/craft1.htm http://antwrp.gsfc.nasa.gov/apod/ap990109.html
Magellan	JPL SAIC	Planetary, Directly Controlled		1989-1994	A Venus orbiter for radar surface mapping and atmosphereic composition analysis http://www.jpl.nasa.gov/magellan/fact.html http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?89-033B
Mariner	JPL	Planetary, Directly Controlled		1962-1971	Mariner 2 and 5 conducted Venus flybys for imaging. 2 was the first interplanetary mission Mariner 4, 7, 7, and 8 conducted Mars flybys for imaging http://www.jpl.nasa.gov/missions/past/ http://nssdc.gsfc.nasa.gov/planetary/projects.html
Mars 2	IKI	Planetary, Teleoperated		1971	A Mars orbiter, lander, and rover to image and conduct scientific atmosphere and soil analyses. The landing module malfunctioned and failed to land safely on Mars. http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?71-045A http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?71-045D
Mars 3	IKI	Planetary, Teleoperated		1971	A Mars orbiter, lander, and rover to image and conduct scientific atmosphere and soil analyses. Mars 3 was the first successful soft landing on Mars but communications were lost shortly after landing. http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?71-049A http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?71-049F
Mars 96	IKI	Planetary, Semi-Autonomous		1996	A group of Martian surface penetrators to conduct autonomous science and an orbiter: FAILED http://arc.iki.rssi.ru/mars96/mars96hp.html http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?96-064A
Mars Autonomy	CMU	Tools, Autonomous		1998-	A system for autonomous local and global navigation for planetary rovers; in simulation http://www.frc.ri.cmu.edu/projects/mars
Mars Climate Orbiter	NASA JPL	Planetary, Directly Controlled		1998-1999	Mars orbiter for atmosphere composition and imaging: FAILED http://mars.jpl.nasa.gov/msp98/orbiter
Mars Express (ASPERA-3)	ESA	Planetary, Teleoperated		2003	An orbiter, a Beagle 2 lander, and a rover for mapping water on Mars http://sci.esa.int/marsexpress http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?MARSEXP
Mars Geoscience/ Climatology Orbiter (MGCO)	JPL	Planetary, Directly Controlled		1992-1993	See Mars Observer
Mars Global Surveyor (MGS)	JPL Stanford	Planetary, Directly Controlled		1996	A Mars orbiter for imaging and spectroscopy http://mars.jpl.nasa.gov/mgs/

Name	Agencies	Section	Page	Dates	Description and References
Mars Network	JPL	Planetary, Directly Controlled		2003	A constellation of Mars orbiters for sience and communication relay http://marsnet.jpl.nasa.gov/
Mars Observer (Mars Geoscience/ Cilmatology Orbiter, MGCO)	JPL	Planetary, Directly Controlled		1992-1993	A Mars orbiter for imaging and atmospheric studies: FAILED http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?92-063A
Mars Pathfinder	JPL	Planetary, Semi-Autonomous		1996-1997	Mars lander and Sojourner rover with autonomous execution of scripted commands and autonomous obstacle avoidance http://mars.jpl.nasa.gov/MPF/index1.html http://mpfwww.jpl.nasa.gov/rover/sojourner.html
Mars Polar Lander	JPL	Planetary, Teleoperated		1998-1999	Mars lander for the investigation of water near the south pole, used WITS control interface: FAILED http://mars.jpl.nasa.gov/msp98/index.html
Mars Surveyor 2001	JPL	Planetary, Teleoperated		2001	Mars orbiter; Downgraded from Mars Athena Precursor Experiment (APEX) which included a rover http://mars.jpl.nasa.gov/2001/ http://athena.cornell.edu/
Mars Surveyor 2003/2005 (Athena)	JPL Cornell	Planetary, Semi-Autonomous		2003, 2005	Mars rover, teleoperated with autonomous obstacle avoidance; terrestrial testbed is FIDO http://athena.cornell.edu/ http://nssdc.F64gsfc.nasa.gov/planetary/mars_2003_05.html
Marsokhod	IKI	Planetary, Semi-Autonomous		1993-1999	A Mars rover prototype demonstrated in desert and volcano settings http://img.arc.nasa.gov/Marsokhod/marsokhod.html http://img.arc.nasa.gov/
Mercury Surface, Space Environment Geochemistry and Ranging (MESSENGER)	NASA GSFC JHU APL	Planetary, Directly Controlled		2004	A Mercury orbiter for investigation of the core and polar compositions, density, magnetics, and geological history http://sd-www.jhuapl.edu/MESSENGER/
MESSENGER	NASA GSFC JHU APL	Planetary, Directly Controlled		2004	See Mercury Surface, Space Environment and Geochemistry Ranging
MISUS	JPL	Tools, Autonomous		1999-	See Multi-Rover Integrated Science Understanding System
MPOD	UMD SSL	Orbital, Semi-Autonomous		1995-	See Multimode Proximity Operations Device
Multimode Proximity Operations Device (MPOD)	UMD SSL	Orbital, Semi-Autonomous		1995-	A tool and testbed for autonomous spacecraft approach and docking using neural network control http://www.ssl.umd.edu/homepage/Projects/MPOD/mpod.html
Multi-Rover Intgrated Science Understanding System (MISUS)	JPL	Tools, Autonomous		1999-	A tool for autonomously achieving science goals (planning and scheduling), using incremental planning "iterative repair" and learning with clustering methods http://www-aig.jpl.nasa.gov/public/mls/multirover
Multisensory Articulated Hand / Robotnaut	DLR	Human Assistance, Teleoperated		1997-	An articulated hand prototype with force and torque sensing for humanoid space robots such as DLR Robotnaut http://www.robotic.dlr.de/HAND http://www.robotic.dlr.de/LBR/
Muses-A (Hiten)	ISAS	Planetary, Directly Controlled		1980-1993	A lunar flyby, which injected a lunar orbiter (Hagoromo) into Lunar orbit http://www.isas.ac.jp/e/enterp/missions/complate/hiten.html
Muses-C	ISAS JPL	Deep Space Semi-Autonomous		2002	See MuSpace Engineering Spacecraft and Nanorover
Muses-CN	ISAS JPL	Deep Space Semi-Autonomous		2002	See MuSpace Engineering Spacecraft and Nanorover
Muses-D (Planet-C)	ISAS	Planetary, Directly Controlled		2005	A Mercury orbiter to investigate the iron core, surface composition, and magnetic field http://nssdc.gsfc.nasa.gov/planetary/prop_missions.html#isas_mercury_orb
MuSpace Engineering Spacecraft and Nanorover (Muses-C/Muses-CN)	ISAS JPL	Deep Space Semi-Autonomous		2002	A mission to land and return a sample from Asteroid (10302) 1989ML; the lander will conduct an autonomously navigated landing, and a nanorover will be deployed for spectroscopy http://www.isas.ac.jp/e/enterp/missions/muses-c/cont.html http://www.muses-c.isas.ac.jp/index.html http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?MUSES-C
Nanorover	JPL	Planetary, Semi-Autonomous		2002	A small rover for planetary surface operations and science; to be used on MUSES-C (MUSES-CN) See MUSES C http://robotics.jpl.nasa.gov/tasks/nrover/homepage.html
NEAP	SpaceDev	Deep Space, Directly Controlled		2001	See Near Earth Asteroid Prospector
NEAR	JHU APL NASA GSFC JPL	Deep Space, Directly Controlled		1996	See Near Earth Asteroid Rendezvoud

Name	Agencies	Section	Page	Dates	Description and References
Near Earth Asteroid Prospector (NEAP)	SpaceDev	Deep Space, Directly Controlled		2001	A craft to orbit and land on the Nereus asteroid, with intend to claim it for commercial prospecting http://www.spacedev.com/Missions/MicroNEAP.htm http://nssdc.gsfc.nasa.gov/planetary/prop_missions.html#neap
Near Earth Asteroid Rendezvous Shoemaker (NEAR)	JHU APL NASA GSFC JPL	Deep Space, Directly Controlled		1996	A flyby mission of the Eros asteroid for imagery and spectroscopy http://near.jhuapl.edu/
Nomad	CMU	Planetary, Autonomous		1995-2000	See Robotic Antarctic Meteorite Search
Nozomi (Planet-B)	ISAS	Planetary, Directly Controlled		1996 - 2003	A mars orbiter for imaging and atmospheric studies http://www.planet-b.isas.ac.jp/index-e.html http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?98-041A http://www.isas.ac.jp/e/enterp/missions/nozomi/cont.html
PDM	NASA JPL	Planetary, Teleoperated		1998-2005	See Planetary Dexterous Manipulators
Personal Satellite Assistant (PSA)	NASA ARC	Human Assistance, Semi-Autonomous		1998-	A freeflyer for monitoring, communication, and worksite support with semi-autonomous navigation and teleoperated modes; a testbed is in operation http://ic-www.arc.nasa.gov/ic/psa/
Phobos	IKI	Planetary, Directly Controlled		1998	Two orbiters for Mars and Sun observations: FAILED http://nssdc.gsfc.nasa.gov/planetary/phobos.html
Pioneer	NASA ARC	Deep Space, Directly Controlled		1972-1997	Pioneer 10: Flybys of the asteroid belt and Jupiter to obtain images and data on high-energy particles and magnetic fields, now in route to constellation Taurus Pioneer 11: Flybys of Jupiter and Saturn, where it investigated high-energy particles http://spaceprojects.arc.nasa.gov/Space_Projects/pioneer/PNhome.html
Pioneer Venus	NASA ARC	Planetary, Directly Controlled		1978-1992	A Venus orbiter and landing probe which conducted imaging, spectroscopy, field and atmospheric analyses http://spaceprojects.arc.nasa.gov/Space_Projects/pioneer/PNhist.html http://nssdc.gsfc.nasa.gov/planetary/pioneer_venus.html
Planet-A	ISAS	Deep Space, Directly Controlled		1985	See Suisei
Planetary Dexterous Manipulators (PDM)	JPL	Planetary, Teleoperated		1998-2005	A series of dexterous manipulators for planetary/space applications including instrument pointing and sample collection; some flown on Mars Pathfinder and other missions http://robotics.jpl.nasa.gov/tasks/pdm/homepage.html
Planet-B (Nozomi)	ISAS	Planetary, Directly Controlled		1996-2003	See Nozomi
Planet-C	ISAS	Planetary, Directly Controlled		2005	See Muses-D
Pluto-Kuiper Express	NASA GSFC JPL	Deep Space, Directly Controlled		2004	A mission to Pluto and moon Charon for imaging, mapping, and compositional characterization http://www.jpl.nasa.gov/ice_fire//pkexprss.htm
PROP-M	IKI	Planetary, Teleoperated		1971	See Mars 2 and Mars 3
PSA	NASA ARC	Human Assistance, Semi-Autonomous		1998-	See Personal Satellite Assistant
Ranger	UMD SSL	Human Assistance, Teleoperated		1990-	A freeflying teleoperated robot with multiple manipulators for shuttle and space station EVA operations http://www.ssl.umd.edu/homepage/Projects/RangerTSX/RangerTSX.html http://www.ssl.umd.edu/homepage/Projects/ranger.html http://img.arc.nasa.gov/Ranger/index.html
Ranger	NASA JPL	Planetary, Directly Controlled		1961-1965	A series of Lunar orbiters (2 and 3) for imaging as a precusor to the Apollo missions http://www.jpl.nasa.gov/missions/ranger/ http://nssdc.gsfc.nasa.gov/planetary/lunar/ranger.html
Remote Agent	NASA ARC JPL	Tools, Autonomous		1998	The autonomous control system for Deep Space 1. Tested with NewMAAP simulation tool 1995. Autonomous scheduling for achieving goals, primarily course corrections http://www-aig.jpl.nasa.gov/public/planning/nmds1/ http://www.rax.arc.nasa.gov/ http://ic-www.arc.nasa.gov/ic/projects/Executive/ http://nmp.jpl.nasa.gov/ds1/tech/autora.html http://rax.arc.nasa.gov/
Robonaut	NASA JSC	Human Assistance, Teleoperated		1997-	A teleoperated humanoid robot astronaut surrogate for EVA activities on the shuttle and space station http://ranier.oact.hq.nasa.gov/telerobotics_page/fy97plan/chap2g.html#Robonaut http://vesuvius.jsc.nasa.gov/er_er/html/robonaut/robonaut.html http://tommy.jsc.nasa.gov/robotnaut/Robonaut.html

Name	Agencies	Section	Page	Dates	Description and References
Roboter Technology Experiment (ROTEX)	DLR	Orbital, Teleoperated		1993	A teleoperated robotic arm for shuttle EVA operations, a first step in service robots http://www.robotic.dlr.de/TELEROBOTICS/rotex.html
Robotic Antarctic Meteorite Search/ Nomad	CMU	Planetary, Autonomous		1995-2000	Autonomous science and navigation experiment. Terrestrial prototype for large-scale planetary rovers, Nomad demonstrated in desert and Antacrtica; autonomously classifies rocks as terrestrial or meteorite http://www.frc.ri.cmu.edu/FRC/nomad.html http://img.arc.nasa.gov/Nomad/nomad.html
Robotnaut	DLR	Human Assistance, Teleoperated		1997-	A light-weight humanoid robot platform for space/EVA - See Multisensory Articulated Hand
Rocky	JPL	Planetary, Autonomous		1990-	See Long Range Science Rover
Rosetta	ESA	Deep Space, Teleoperated		2003	A rendezvous and landing on Comet Wirtanen http://sci.esa.int/rosetta http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?ROSETTA
ROTEX	DLR	Orbital, Teleoperated		1993	See Roboter Technology Experiment
Sakigake (MS-T5, Pioneer)	ISAS	Deep Space, Directly Controlled		1985	A heliocentric orbiter to collect data on plasma wave spectra, solar wind ions, and interplanetary magnetic fields; a flyby of Halley's comet was made http://www.isas.ac.jp/e/enterp/missions/complate/sakigake.html http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?85-001A
SCAMP	UMD SSL	Human Assistance, Teleoperated		1992-	See Supplemental Camera and Maneuvering Platform
Science Planning and Scheduling System (SPSS)	NASA STSCI	Tools, Semi-Autonomous		1990-	A tool for scheduling, including short-term observations for Hubble Space Telescope http://www.pst.stsci.edu/~samson/overview_doc.cgi http://www.pst.stsci.edu/spss/spss.html
Selene	ISAS NASDA	Planetary, Semi-Autonomous		2003	See Selenological and Engineering Explorer
Selenological and Engineering Explorer (Selene)	ISAS NASDA	Planetary, Semi-Autonomous		2003	A lunar orbiter and lander for spectroscopy and imaging to investigate Lunar evolution http://www.isas.ac.jp/e/enterp/missions/selene/cont.html http://spaceboy.nasda.go.jp/note/tansa/e/tan109_selene_e.html http://J96nssdc.gsfc.nasa.gov/planetary/prop_missions.html#selene1 http://yyy.tksc.nasda.go.jp/Home/Projects/SELENE/index_e.html
Shuttle Remote Manipulator System (SRMS/Canadarm)	SPAR	Human Assistance, Directly Controlled		1975-	A teleoperated arm for shuttle bay operations, included on a shuttle mission http://www.mdrobotics.ca/canaframe.htm
SIM	JPL	Deep Space, Directly Controlled		2006	See Space Interferometry Mission
Skyworker	CMU	Orbital, Semi-Autonomous		2002	A prototype for an autonomous walking rover for construction of orbital solar power stations http://www.frc.ri.cmu.edu/projects/skyworker
Small Missions for Advanced Research in Technology 1 (SMART-1)	ESA	Planetary, Directly Controlled		2002	A craft for demonstration of solar electric propulsion; it will conduct Lunar science from Lunar orbit http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?SMART_1 http://sci.esa.int/smart, http://www.ssc.se/ssd/
SMART-1	ESA	Planetary, Directly Controlled		2002	See Small Missions for Advanced Research in Technology 1
SOHO	ESA	Deep Space, Directly Controlled		1995-	See Solar and Heliospheric Observatory
Sojourner	JPL	Planetary, Semi-Autonomous		1996-1997	See Mars Pathfinder
Solar and Heliospheric Observatory (SOHO)	ESA	Deep Space, Directly Controlled		1995-	A Solar observatory for investigation of the sun's interior, corona, and solar wind; operating in orbit around the L1 Lagrangian point; data to be coordinated with TRACE http://sci.esa.int/soho http://sohowww.nascom.nasa.gov
Solar Probe	NASA JPL	Deep Space, Directly Controlled		2007	A Solar probe to near the Sun's surface to investigate solar wind, coronal energy and make density maps http://www.jpl.nasa.gov/ice_fire//sprobe.htm
Solar Terrestrial Relations Observatory (STEREO)	NASA GSFC JHU APL	Deep Space, Directly Controlled		2004	A pair of satellites to study the Earth-Sun relationships, plasma dynamics, and weather; operating in heliocentric orbit at 1 AU as part of the Solar Terrestrial Probes (STP) program http://sd-www.jhuapl.edu/STEREO/index.html http://STProbes.gsfc.nasa.gov/stereo.htm
Solar-A (Yohkoh)	ISAS NASA BNSC	Deep Space, Directly Controlled		1991	A heliocentric orbiter to investigate the Sun's corona and the Solar cycle with X-ray imaging http://www.isas.ac.jp/e/enterp/missions/yohkoh/cont.html

Name	Agencies	Section	Page	Dates	Description and References
Solar-B	ISAS NASA MSFC	Deep Space, Directly Controlled		2004	An orbiter in a sun-synchronous polar orbit to investigate the magnetic field, luminosity, radiation, and atmospheric disturbances http://www.isas.ac.jp/e/enterp/missions/solar-b/cont.html http://wwwssl.msfc.nasa.gov/ssl/pad/solar/solar-b.htm
Space Interferometry Mission (SIM)	JPL	Deep Space, Directly Controlled		2006	A satellite for large-baseline interferometry, operating in Earth's orbit with increasing lag behind Earth until about 5 AU http://sim.jpl.nasa.gov/
Space Technology 3 (Deep Space 3)	JPL	Deep Space Semi-Autonomous		2003	A constellation for formation flying satellites for long-baseline interferometry; autonomous maintenance of specified formations http://nmp.jpl.nasa.gov/st3/index.html
Space Technology 5 (Deep Space 5)	NASA GSFC	Deep Space Semi-Autonomous		2003	A constellation of nanosatellites flying in formations for magnetometry studies; autonomous ground scheduling and orbit determination and configuration will be employed http://nmp.jpl.nasa.gov/st5
SPIDER	ASI	Human Assistance, Teleoperated		1988-	A teleoperated robotic arm for space station operations http://www.asi.it/00HTL/eng/asicgs/robotics/programmaAR/ARhome.html
Spike	NASA STSCI	Tools, Semi-Autonomous		1990-	A tool for scheduling, including long-term observations for Hubble Space Telescope http://www.stsci.edu/spike/ http://www.pst.stsci.edu/spss/spss.html
Sprint	NASA JSC	Human Assistance, Semi-Autonomous		1997	See Autonomous EVA robotic Camera
SPSS	NASA STSCI	Tools, Semi-Autonomous		1990-	See Science Planning and Scheduling System
SRMS	SPAR	Human Assistance, Directly Controlled		1975-	See Shuttle Remote Manipulator System
Stardust	JPL UWA LMCO	Deep Space Semi-Autonomous		1999	A mission for cometary material and interstellar dust sample return with autonomous optical navigation and autonomous science instrument deployment http://stardust.jpl.nasa.gov/mission/msnover.html
Startool	JPL	Tools, Semi-Autonomous		1996-	A tool for autonomous solar feature identification using image segmentation; for future solar physics investigations http://www-aig.jpl.nasa.gov/mls/solar/solar.html
STEREO	NASA GSFC JHU APL	Deep Space, Directly Controlled		2004	See Solar Terrestrial Relations Observatory
Suisei (Planet-A)	ISAS	Deep Space, Directly Controlled		1985	A heliocentric satellite for a Halley's comet flyby to image the corona http://www.isas.ac.jp/e/enterp/missions/complate/suisei.html http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?85-073A
Supplemental Camera and Maneuvering Platform (SCAMP)	UMD SSL	Human Assistance, Teleoperated		1992-	A free-flying camera platform; currently teleoperated, autonomy is under investigation http://www.ssl.umd.edu/homepage/Projects/SCAMP/Project_overview.html
Surveyor	JPL	Planetary, Teleoperated		1966-1968	A series of Lunar landers for imaging and soil analysis. Successful missions 1,3,5,6,7 http://www.jpl.nasa.gov/missions/past/
Terrestrial Planet Finder (TPF)	JPL	Deep Space Semi-Autonomous		2001	A set of 5 satellites for locating Earth-like planets, including a formation-flying pair in heliocentric orbit at 1 AU http://origins.jpl.nasa.gov/missions/current_status.html http://tpf.jpl.nasa.gov/index.html
Tessellator	CMU	Human Assistance, Autonomous		1992-1994	An autonomous robot for investigating the shuttle for missing tiles http://www.ri.cmu.edu/projects/project_191.html
TPF	JPL	Deep Space Semi-Autonomous		2001	See Terrestrial Planet Finder
TRACE	NASA GSFC Lockheed	Deep Space, Directly Controlled		1998	See Transition Region and Coronal Explorer
Transition Region and Cornal Explorer (TRACE)	NASA GSFC Lockheed	Deep Space, Directly Controlled		1998	A heliocentric orbiter for investigation of magnetic fields and plasma; data combined with coordinated SOHO data http://vestige.lmsal.com/TRACE/
Ulysses (International Solar Polar Mission)	JPL ESA	Deep Space, Directly Controlled		1990	An orbiter investigating the space above the Solar poles, including solar wind and the heliosphere http://sci.esa.int/ulysses http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?90-090B http://ulysses.jpl.nasa.gov/mission/mission.html
Vega	IKI	Deep Space, Directly Controlled		1984	A pair of satellites which delivered balloons to Venus to scientific experiments on the atmosphere and then proceeded to do a Halley's comet flyby http://arc.iki.rssi.ru/ssp/vega.html http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmc?84-125A http://nssdc.gsfc.nasa.gov/cgi-bin/database/www-nmd?84-128A
Venera	IKI	Planetary, Directly Controlled		1961-1983	A series of 16 Venus orbiters and landers for scientific studies of the atmosphere and surface http://nssdc.gsfc.nasa.gov/planetary/venera.html

Name	Agencies	Section	Page	Dates	Description and References
Voyager	NASA GSFC MIT JHU APL U Iowa JPL	Deep Space, Directly Controlled		1977	A series of planetary flybys of Jupiter and Saturn (Voyager 1 and 2), and Uranus and Neptune (Voyager 2), followed by exploration outside the solar system to investigate plasma, cosmic rays, radiation, and magnetic fields http://vraptor.jpl.nasa.gov/voyager/voyager.html
Web Interface for Telescience (WITS)	JPL	Tools, Semi-Autonomous		1996	A graphical interface for spacecraft model-based control, used on the Mars Polar Lander http://www-aig.jpl.nasa.gov/public/planning/rover/ http://wits.jpl.nasa.gov/public/index.htm
WITS	JPL	Tools, Semi-Autonomous		1996	See Web Interface for Telescience
Yohkoh	ISAS NASA BNSC	Deep Space, Directly Controlled		1991	See Solar-A
Zond	IKI	Planetary, Directly Controlled		1965-1970	A series of lunar flybys and orbiters for imaging and science data collection http://nssdc.gsfc.nasa.gov/planetary/lunar/lunarussr.html