Experimental Demonstration of Technologies for Autonomous On-Orbit Robotic Assembly

Dr. Edward A. LeMaster* and Dr. David B. Schaechter†
Lockheed Martin Advanced Technology Center, Palo Alto, CA, 94304

Dr. Connie K. Carrington‡
NASA Marshall Space Flight Center, Huntsville, AL, 35812

The Modular Reconfigurable High Energy (MRHE) program aimed to develop technologies for the automated assembly and deployment of large-scale space structures and aggregate spacecraft. Part of the project involved creation of a terrestrial robotic testbed for validation and demonstration of these technologies and for the support of future development activities. This testbed was completed in 2005, and was thereafter used to demonstrate automated rendezvous, docking, and self-assembly tasks between a group of three modular robotic spacecraft emulators. This paper discusses the rationale for the MRHE project, describes the testbed capabilities, and presents the MRHE assembly demonstration sequence.

I. □ Introduction

Current spacecraft design methodologies generally revolve about the concept of a solitary, monolithic spacecraft bus. Such spacecraft are very capable, but also face significant limitations in view of future plans and visions for expanded human presence on-orbit, on the moon, and on Mars. Perhaps foremost is the spacecraft size limitation presented by current and next-generation launch vehicles, especially when considering the requirement to boost spacecraft beyond Earth orbit. Manned spacecraft clearly must carry additional resources and possess significant capabilities beyond those required for their unmanned counterparts, especially for extended-duration missions. This will likely require the assembly or construction of larger vehicles and/or structures in orbit, in order to concentrate these resources in locations where they are readily accessible and can be used effectively. This may take the form of large aggregate spacecraft that would then move out of low Earth orbit, or perhaps orbiting resource depots that would store consumables such as fuel, food, and water and provide large-scale power generation capabilities for future use. Besides merely allowing for the possibility of larger spacecraft, in-space assembly also offers potential benefits by allowing more-flexible use of launch vehicle types and launch dates.

Although we have experience with on-orbit construction in the form of Mir and the International Space Station, such tasks are still difficult, time consuming, and expensive. Frequently requiring astronaut extravehicular assistance, they can be dangerous as well. Automation using robotic vehicles has the potential to reduce cost, time, and risk for such tasks, although it will also require additional development of numerous supporting technologies. These technologies include improved local sensing and relative navigation; more-efficient microthrusters and manipulators; robust fault-detection and correction software for automated rendezvous, docking, and assembly tasks; adaptive planning and scheduling; human-robotic interaction for complicated tasks that cannot be completely automated; and mechanical mechanisms such as deployable booms, tethers, docking interfaces, and other construction tools.

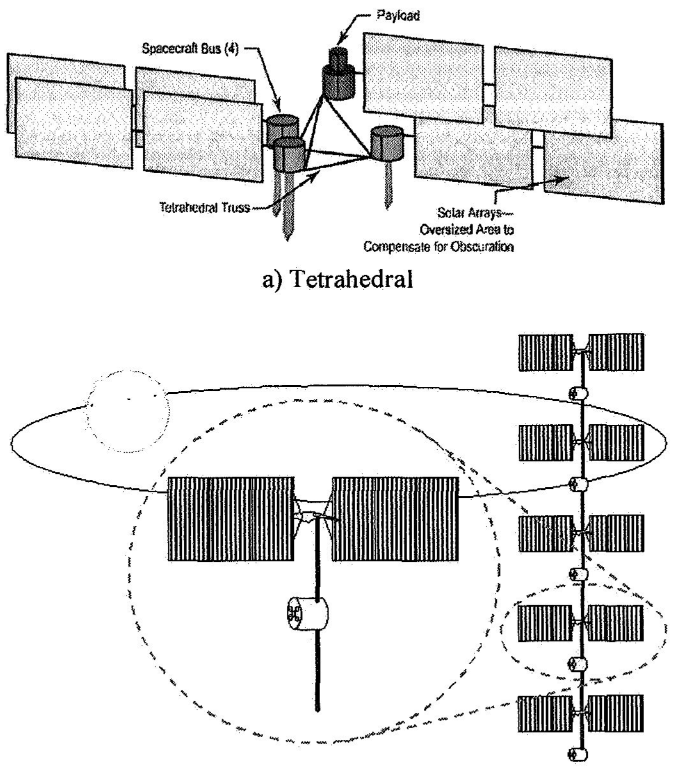
Development of modular or standardized hardware and software components and interfaces will also be instrumental towards reducing system costs, complexity, and development time. Although the complexity associated with spacecraft systems and the necessity to reduce on-orbit mass to a minimum may preclude a complete transition to the kind of "plug-and-play" architecture we use with desktop computers, significant modularization and

* Staff Controls Engineer, Precision Pointing and Controls, 3251 Hanover St., O/ABFS, B/201, Member AIAA.
† LM Fellow, Precision Pointing and Controls, 3251 Hanover St., O/ABFS, B/201, Associate Fellow AIAA.
‡ Aerospace Engineer, Advanced Sensors, EV21, Marshall Space Flight Center.

standardization can certainly be achieved. In addition to simplifying on-orbit construction, modularized components potentially allow for spacecraft repair and maintenance by replacing faulty parts, and reconfiguration for changing tasks and even missions by swapping payloads or sensors. The foremost examples of the tremendous benefits offered through modular design are the numerous upgrades and repairs that astronauts have performed on the Hubble Space Telescope, greatly enhancing its capabilities and prolonging its mission life.

II. □Modular Reconfigurable High Energy Project

In 2004, as a part of President Bush's Space Exploration Initiative, NASA Headquarters embarked upon a technology development program to return humans to the Moon and to send robotic science missions into the Solar System as pathfinder missions for eventual crewed expeditions. As a part of this effort, the Office of Exploration Systems' Human and Robotic Technology area solicited responses that would overhaul the conventional space technology development paradigm. Its objective was the rapid development and delivery of robust, adaptable, cost-effective spacecraft and flight hardware suitable for a broad variety of missions. The two-pronged effort toward this technology effort comprised the Advanced Space Technology Program and the Technology Maturation Program, the eventual goal being to advance the technology maturation level of key technologies to Technology Readiness Level (TRL) 6 by the completion of the project in 2008.



a) Tetrahedral

b) Linear

Figure 1. MRHE spacecraft concepts

The Marshall Space Flight Center (MSFC) identified modular reconfigurable spacecraft, assembled in orbit from identical building-block components, as a technology that was both critical and ideally suited to meet the demand for affordable, routine space exploration. Such a technology delivers sizeable cost savings through multiple-unit production, and provides options for affordable module replacement and inexpensive module redundancy. On-orbit assembly allows mission planners more flexibility in choosing launch vehicles and support hardware, increasing mission value and affordability. MSFC proposed to NASA HQ as a part of the intramural technology maturation competitive proposal process a Modular, Reconfigurable, High-Energy (MRHE) free-flying satellite concept as a mechanism toward developing these needed technologies. For this proposal MSFC teamed with several subcontractors, including the Lockheed Martin Advanced Technology Center (LM ATC), to advance the technologies associated with the development of a robust, scalable package complete with a high energy, high-efficiency, low-mass solar power system.

The basis of the MRHE concept, two possible variations of which are depicted notionally in Fig. 1, is to assemble aggregate groups of identical solar-powered buses with direct-drive electric propulsion systems in low Earth orbit and then boost them to high-Earth, lunar, or even interplanetary orbits or trajectories in a fuel- and cost-efficient manner. Each bus would have modular payload attachment points in order to accommodate multiple technology experiments or exploration payloads with flexibility and configurability. Efficient, lightweight solar concentrators drive the electric propulsion thrusters, and an innovative, solid-state "Supertube" thermal heat pipe and radiator technology would be used to provide unprecedented thermal management. Early concept definition trades selected the linear Solar Clipper, which provided a reference configuration for design of the MRHE assembly testbed demonstration.

The MRHE project goal was to mature the technologies that support future development of such a modular, 100 kilowatt-class satellite suitable for on-orbit assembly and reconfiguration. The technology maturation effort was to integrate advanced solar power generation, high-voltage power management and distribution, direct-drive of electric propulsion systems, and advanced thermal management into system-of-systems demonstrations. The activity also was to address multi-satellite rendezvous and self-assembly, as well as distributed avionics and robotic reconfiguration in the event of satellite propulsion or power system failure.

As a part of the first year of the program plan, Lockheed Martin was tasked with creation and operation of the MRHE technology demonstrator platform. This effort entailed the implementation of an increasingly realistic sequence of system-of-systems laboratory demonstrations that support the development of the MRHE concept using a scaled-down cluster of three robotic "mini-satellite" emulators in order to address key functional issues of the MRHE program. The first of this sequence of demonstrations, the technology concept demonstration, occurred during the first year of the MRHE program and serves as the basis for the remainder of this paper. The demonstration system was assembled and testing was performed at the Lockheed Martin Controls and Automation Laboratory in Palo Alto, California.

III. □Controls and Automation Laboratory

The Controls and Automation Laboratory presents a unique environment for the development and testing of technologies associated with multi-vehicle autonomous operations for space environments. The laboratory assets include a very large air-bearing surface for the simulation of zero-gravity operations, a custom pseudo-starfield for the emulation of star-tracker-like navigation, numerous robotic vehicles, and associated computing and wireless communications infrastructure.

A. Air-bearing surface

The centerpiece of the laboratory is a large (12'x24') granite surface place, with a flatness of better than 0.0006" end to end (Fig. 2). The table rests on a three-point mount, and is leveled using a set of mechanical wedge blocks. This table provides an air-bearing surface of sufficient size to simultaneously operate a handful of medium-sized robotic vehicles, or potentially dozens of smaller vehicles, while still maintaining adequate separation for maneuverability. The extreme flatness and levelness of the table minimizes the amount of onboard compressed air used by the vehicles for flotation, thereby extending the time available for vehicle operations between refueling. A simple railing system prevents the vehicles from falling off the table in the event of system failures.

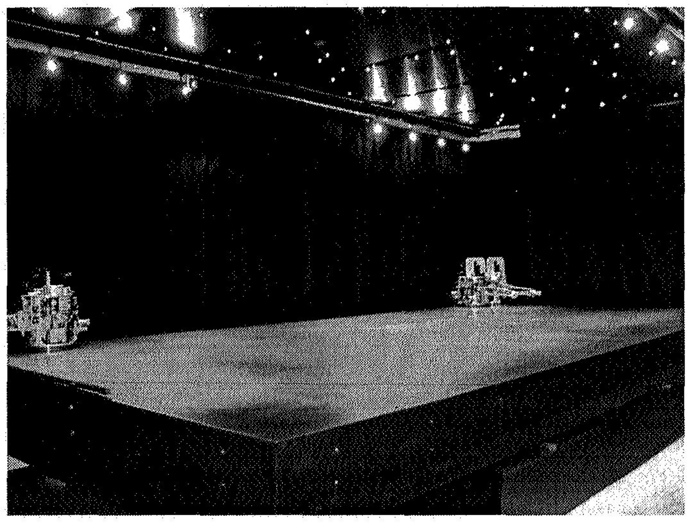


Figure 2. Granite surface plate

B. Robotic Vehicles

The Controls and Automation Laboratories currently features half a dozen free-floating robotic vehicles. Each vehicle is a cylinder roughly 40 cm across and 50 cm tall, with a mass of approximately 50 kg. Although the laboratory vehicles have been constructed at different times for different purposes, the basic bus design does not vary significantly from vehicle to vehicle. Systems present on each vehicle include actuators, sensors, computation, communications, and power, in addition to the specialized hardware for the current project associated with rendezvous and docking and boom deployment. A picture of one of the vehicles, as modified for the MRHE project, appears in Fig. 3.

Actuation for the vehicles comes from two separate systems, the pneumatic subsystem and the reaction wheel. The pneumatic system consists of a pair of high-pressure air tanks with distribution manifolds (used up to 2000 psi), a plenum-based air cushion with a computer-controlled flotation valve, and eight thruster valves distributed around the vehicle in four clusters of two. These thrusters do not face straight out from the vehicle; instead they are canted at specific angles optimized to minimize fuel use during translation maneuvers. The thruster valves vent through small pneumatic barb fittings, and can generate up to 1 N of force depending on the feed pressure. Maximum flotation lifetime is about 30 minutes, although this can be dramatically reduced by significant thruster use.

To minimize the use of compressed air, rotation control of solo (undocked) vehicles is achieved entirely using the onboard reaction wheels. Each wheel generates approximately 0.3 Nm of torque and has an operating speed of up to 3000 rpm, sufficient to rotate the vehicle through a 90 degree slew in under 5 seconds.

Each vehicle has two primary navigation sensors, an upward-looking "star-tracker" and a side-looking docking camera, both of which are described further in the next section. These cameras send their data to the computer over the vehicle Ethernet bus. In addition, each vehicle has sensors for monitoring the pressure in its onboard air tanks and its battery voltage. The vehicles also have onboard gyroscopes and accelerometers for inertial navigation, although these are not used for the current project.

The primary onboard computer stacks utilize x86-based processors in a PC-104 form factor and provide for power conditioning, analog and digital I/O, compact flash storage, and onboard Ethernet. Laboratory vehicles run on a mix of operating systems ranging from legacy DOS to the QNX real-time operating system. The laboratory ground station computers are Windows/Linux machines running custom software for communicating with and commanding the vehicles.

Communications both within and between vehicles and the ground station is accomplished through the use of 802.11g wireless Ethernet bridges. These bridges have a built-in hub/switch capability that allows connection of the onboard computers with the Ethernet-enabled cameras and other onboard components, while simultaneously allowing communications with the other vehicles and grounds stations. Encryption and other protocols insulate this wireless network from other computer networks that may be operating nearby.

In addition to the primary component systems described above, each vehicle has a power system consisting of a regulated 24V battery, several custom and COTS electronics boards to drive the actuators and filter the sensor signals, and a handful of switches and dials for operator feedback and computer override.

C. Vehicle Navigation

Accurate navigation is of critical importance for groups of spacecraft operating in close proximity. In addition to the generic task of orbital maintenance using global position and attitude knowledge, such vehicles may also need accurate relative navigation for tasks such as formation maintenance, collision avoidance, rendezvous and docking, fuel-use optimization, and distributed sensor calibration and operation. This navigation information could come from a variety of sensors such as star trackers, GPS receivers, horizon sensors, magnetometers and gyros, radio-frequency ranging and timing transceivers, laser or LADAR sensing systems, or visual cameras. For examples of current and proposed spacecraft navigation systems, we refer the readers to Refs 1-5.

Accurate navigation is no less important for the laboratory testbed. In addition to the capability to mount a variety of sensors onboard the vehicles in order to test various navigation systems, the Controls and Automation Lab provides a generic navigation system for all vehicles in the form of an array of light-emitting diodes (LEDs) on the ceiling above the granite table that mimics a field of stars (Fig. 4). This array gives an absolute position and heading reference for vehicles

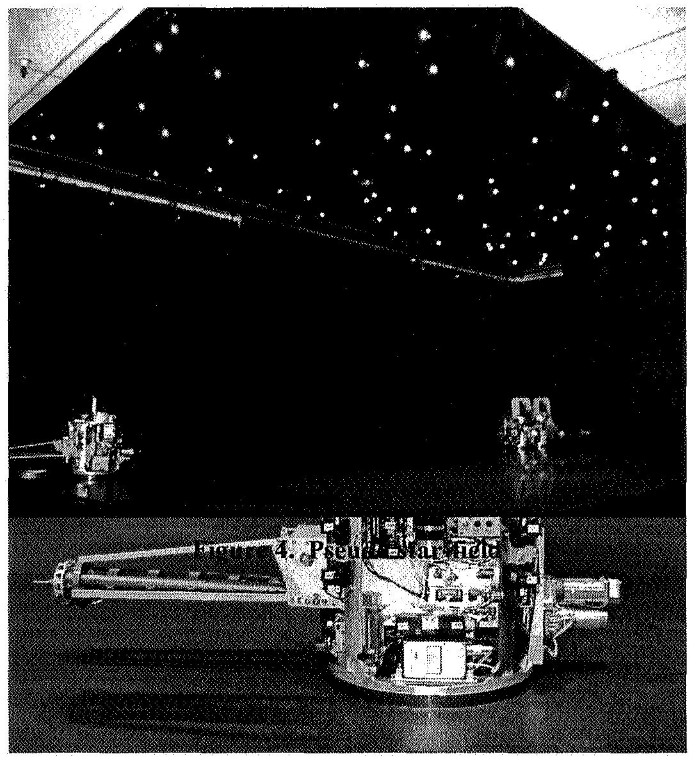


Figure 4. Pseudo Star field

Figure 3. Robotic vehicle

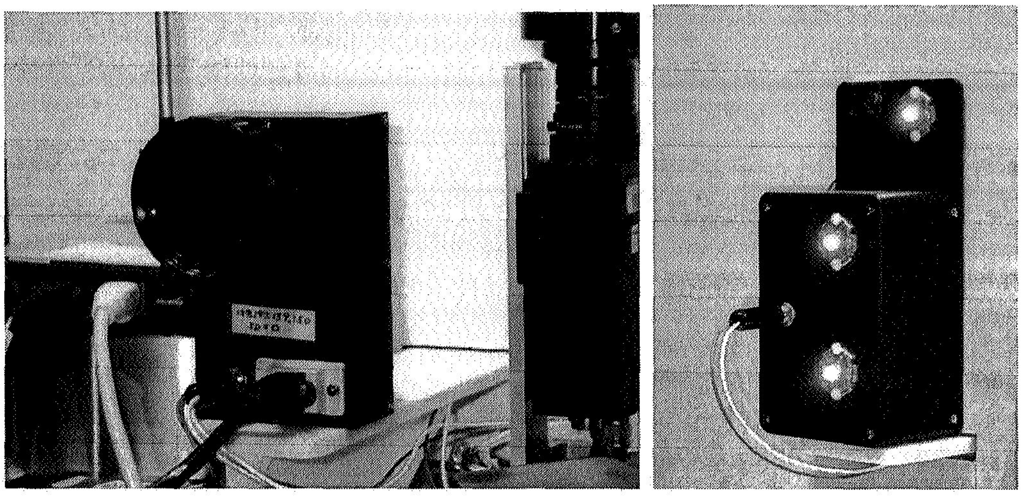


Figure 5. Docking camera and target

operating on the table through the use of upward-looking emulated "star tracker" cameras located on each vehicle. The array itself consists of 81 individual cyan LEDs, each outputting up to 80 lumens. These LEDs are arranged in a randomly-generated pattern such that every point on the table exhibits a unique pattern of "stars" overhead, enabling each vehicle to determine its position and heading anywhere on the table to a resolution of approximately 1 mm and 0.1 deg, respectively. The camera algorithms for determining position and heading from the star field leverage Lockheed Martin's extensive experience in star-tracker development for actual spacecraft. Narrowband filters on the camera lenses remove light from other sources such as the room ambient ceiling lighting and provide for a clear image.

This star-field navigation system emulates actual spacecraft navigation systems in three primary ways. First, it acts as a direct analog for a star tracker on a spacecraft by providing orientation information with respect to a fixed point in space. Second, it provides absolute position knowledge to the vehicles analogous to the way that GPS provides position knowledge. Although actual spacecraft do not have access to this type of information at this resolution, it is necessary in the laboratory setting to ensure that the vehicles – in very close proximity to each other at all times – do not collide or drift off the edge of the table. Third, it can be used to emulate any number of different relative navigation systems by appropriately combining and filtering the position information from each of the vehicles. In addition to these emulation capabilities, the star-field system also acts as a truth reference system when evaluating the performance of other vehicle-mounted relative navigation systems that would be more-representative of those to be used for proximity operations and formation flying between actual spacecraft.

As an example of the additional sensors that can be mounted on the vehicles, each one is currently equipped with a vision-based relative navigation system that it uses for rendezvous and docking. This system consists of a side-looking Ethernet-equipped smart camera on each vehicle, together with a corresponding target pattern of 3 amber LEDs similar to those used in the star-field (Fig. 5). The docking camera runs a software algorithm that uses the location of the three LEDs in the image to determine the relative position and orientation of the target and the camera. The camera sends the relative range, heading, and bearing information to the host vehicle, which then uses the known location of the camera and target on the vehicles to determine the relative position and orientation of the vehicles themselves.

Because this relative sensing system utilizes cameras, resolution and accuracy degrade quickly with distance: relative angular resolution with the inverse of the distance, and range with the inverse of the distance squared. Up close, however, the resolution is reasonably good: 1.2 mm and 0.1 deg at a range of 1 m. Overall accuracy depends upon the quality of the system calibration, but even with only a rudimentary calibration this system exhibits accuracies on the order of a few mm and a fraction of a degree at docking range. Naturally this configuration with three LEDs is only sufficient for operation in two dimensions, but by adding an additional LED the concept is easily extendible to three dimensions. Note that while this proximity sensing system works well for laboratory use, a similar vision system for actual spacecraft would have to contend with the extreme variability in lighting conditions on-orbit.

D. Specialized Hardware

The MRHE technology demonstration program required the addition of a few specialized hardware components to the stock laboratory vehicles. These mechanical components include docking latches used to mate the vehicles, deployable collapsible booms, and boom deployment mechanisms.

The motorized docking latch used for the current project is an adaptation of an earlier Lockheed Martin kinematic latch design. While the original specifications for this highly precise and repeatable device differ significantly from those for a spacecraft mechanical docking interface, the design is adequate for the present laboratory system. This latch, depicted in Fig. 6, consists of an active component mounted to the chaser vehicle in the docking scenario and a passive half mounted to the target vehicle. The active half includes a 1" cone-shaped lead-in region, spring-loaded jaws, and a jaw retraction motor. The passive half includes a docking probe with a spherical load-bearing surface mounted on the end of the deployable boom. Sensors within the latch motor mechanism detect the presence of the docking probe and command the motor to turn, pulling in the capture jaws and locking the vehicles together. An additional limit switch verifies hard-docking between the two vehicles. To undock the vehicles the motor simply reverses direction, opening up the jaws and releasing the probe.

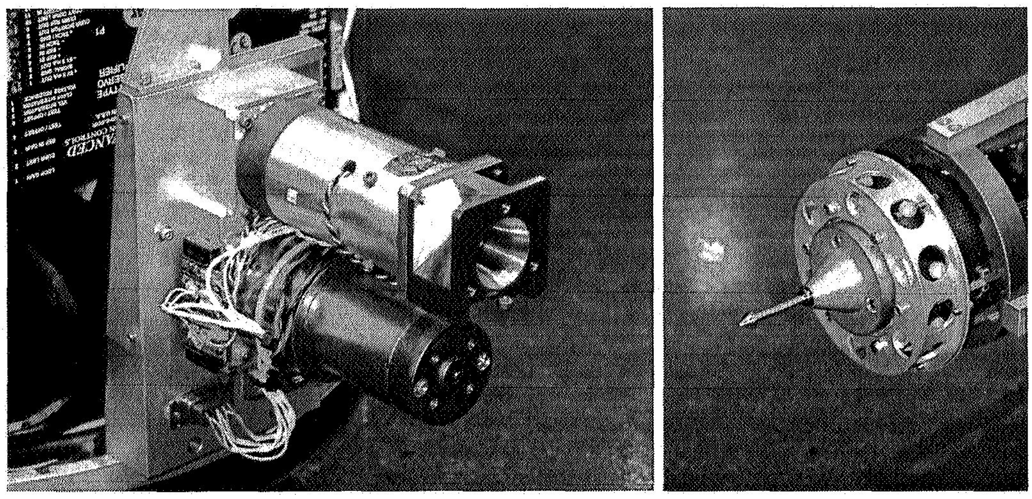


Figure 6. Docking latch and capture probe

The aspect of the MRHE demonstration testbed that is most unique is the presence of the deployable booms that are mounted on two of the three vehicles (Fig. 7). With these booms in place the vehicles, when docked together, can then vary their center-to-center spacing to create an expandable aggregate structure. These 2.5" diameter booms, built by ATK, employ Collapsible Rollable Tube (CRT) technology developed by ATK for Lockheed Martin. The goal of the CRT technology is to achieve a deployable and retractable, flight-like boom with a high

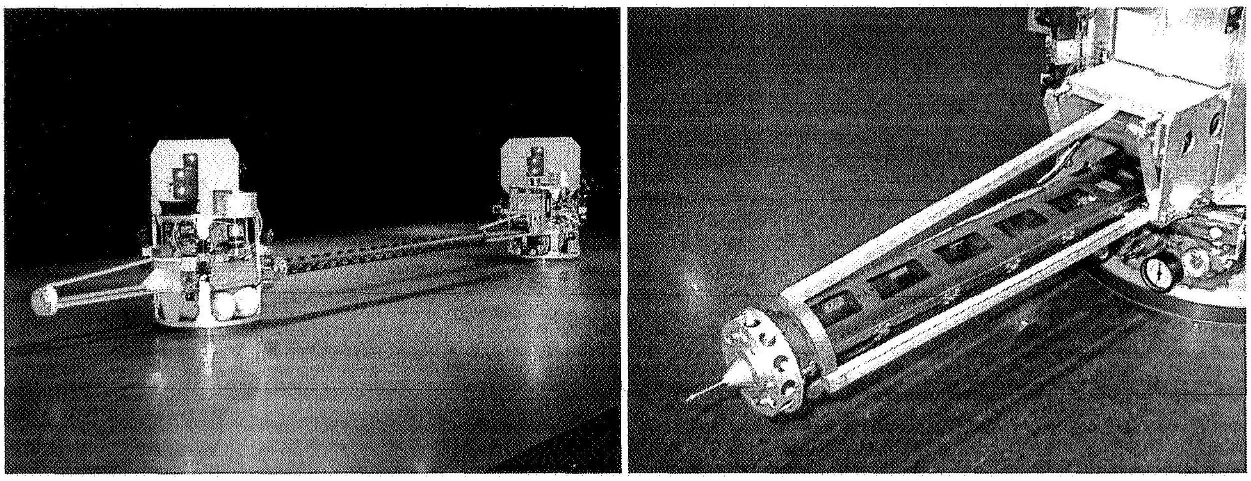


Figure 7. Deployable boom and mechanism

packing ratio and good stiffness/weight characteristics throughout its deployable range. The booms are assembled from two halves of carbon fiber material and roll up onto a compact storage spool, similar to the packing used on a standard metal tape measure. The booms can be deployed and retracted multiple times (dozens to hundreds of cycles), with a current deployable length of greater than 1.5 meters.

The Lockheed Martin-developed deployment mechanism features a positive traction drive, encoder-based position control, and end-of-travel switches.

IV. Software Capabilities

Software algorithms compose some of the most-critical aspects of the MRHE demonstration program. Although the vehicles have a baseline set of operational command and control code, this had to be significantly enhanced in order to provide the functionality necessary for motion planning and obstacle avoidance, task scheduling, boom and latch operation, docking operations, and coordinated control of connected multi-body aggregate spacecraft.

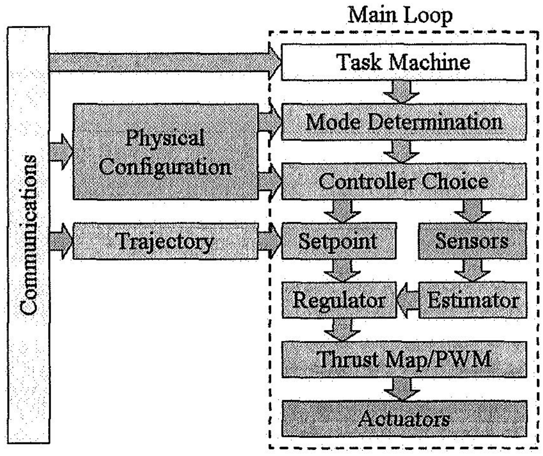


Figure 8. Vehicle software architecture

Figure 8 presents a view of the vehicle software architecture, including some of the major process flow paths. As the figure indicates, most of the activity occurs within the main control loop. A task machine – analogous to a finite state machine – determines the mode of operation for the vehicle. Together with the physical configuration of the vehicle, the current task determines which of a number of possible sensors, regulators, and estimators the vehicle will use. Each of the many different control behaviors of the vehicle – including stationkeeping, trajectory following, and docking – is achieved through judicious choice of the position and velocity setpoints and control gains. The only exception to this rule occurs when the vehicle is docked as a non-leading vehicle in a multiple-vehicle aggregate, in which case its control outputs are radioed to it by the lead vehicle. A number of software components operate outside of the main control loop in a secondary loop. This includes communications and graphical output. The communications engine calls appropriate operators on the controller object to affect its operation.

Code portability, reuse, data protection, and abstraction are maintained through the use of an object-oriented design implemented in C++. Primary modules added for the development effort include the controller (pink in Fig. 8), the communications engine (blue), the task engine (yellow), modules to operate the boom and latch hardware, and a system clock. Specific public operations on these objects allow cross-communication between the modules, while maintaining data separation.

A. Motion Planning

One of the primary software components for vehicle operation is the integrated motion planner. It is the job of this planner to take in the initial starting locations of the vehicles at the beginning of the demonstration sequence and plan collision-free trajectories to get them to their designated starting locations (in an evenly-spaced line near the center of the table). The motion planner currently resides as a centralized planner on the ground station in order to simplify the implementation, and it therefore does not have to deal with the complexities of sequential planning, voting, or other methods used for distributed task-planning. Note that there is conceptually no difference between locating a centralized planner on the ground station as opposed to onboard a mobile vehicle.

Randomized Kinodynamic Motion Planners

The particular planner implemented for MRHE is a Randomized Kinodynamic Motion Planner (RKMP), adapted from the planner developed at the Stanford University Aerospace Robotics Laboratory.[6] An RKMP works by creating a tree of possible trajectories from the starting state (position, orientation, and time) to the goal state. Branches on this tree are generated by randomly selecting the end of a branch in the tree and then applying a random control input, from the range of possible control inputs, for a random amount of time. Branches that would collide with other moving vehicles or fixed obstacles, or would move the vehicle into a forbidden region (i.e. off the table), are discarded, while all others are retained. Branches that end close enough to the goal location – i.e. that pass some prescribed "end-game" criteria – are subsequently completed to yield a continuous trajectory from start to goal.

Planning generally continues for either a specified number of complete trajectories, a maximum number of branches, or a maximum amount of time, and the lowest-cost plan is returned as the final vehicle trajectory.

While such randomized motion planners will not in general provide optimal plans, they do offer a number of advantages that make them worthy of consideration. First, they are extremely fast. Planning times for point masses moving in uncluttered environments are generally measured in milliseconds, while multi-vehicle plans, plans with collision avoidance of booms and other appendages, and plans with multiple moving obstacles can still usually be generated in a fraction of a second. (The typical total planning time for the MRHE demonstration is around one quarter of a second). While this might not be important for many missions, it is critical for applications involving vehicle motion in dynamic, changing, or otherwise unpredictable environments in which rapidly-moving obstacles may require fast decision making.

Second, while the plans themselves are random, performance is actually somewhat predictable. As proved by Hsu and Kindel,[7] although an RKMP is not a complete planner, it is probabilistically complete in suitably expansive environments. This means that the chance of not finding a suitable path drops exponentially to zero as planning time increases. Moreover, if a suitable plan is not found in a reasonable period of time, the planner will tell you so. This is in contrast to other fast but non-complete planners such as those utilizing potential fields, which may not be able to detect deadlock, local minima, or other undesirable solutions. In no plan to the goal is found, it is generally trivial to return at least a viable plan to a region devoid of obstacles – a so-called escape plan – to give the vehicle additional time to either plan a longer route or contact a human operator for assistance.

Third, Randomized Kinodynamic Motion Planners can inherently satisfy all dynamic constraints, including the complicated relative motions exhibited by formations of spacecraft in orbit, by simply including the relevant forces within the dynamic propagator. Such applicable forces include, but are not limited to, gravitational forces, atmospheric drag, solar radiation pressure, and electrostatic and magnetic forces and moments.

It is important to note that although randomized planners are very fast and do not require significant computation, they do require extensive available memory because of the necessity of retaining a large number of branches in the randomized motion tree. This may preclude the use of randomized planners on some systems. In a space environment the ideal situation may actually be a mix of planners, perhaps a fast randomized planner operating in the case of emergencies or when under severe time constraints, along with a slower complete planner to optimize time or fuel use constraints if sufficient time for its operation is available.

MRHE Planner Implementation

While the previous section describes this class of randomized planners in general, this section deals with the implementation used for MRHE. Planning for the multiple vehicles is handled sequentially, with priority depending upon the distance of each vehicle from its goal location. Once a vehicle is planned, it is treated as a moving obstacle by all remaining vehicles. (Planning can also be done concurrently, but this can yield problems when trying to choose which branch of the planning tree to expand next.) The available duration for the maneuvers is usually set for two or three minutes for tabletop operation in the Controls and Automation Laboratory.

The planning tree for each vehicle starts with the root at the vehicle start location. At each loop through the planner – hundreds to thousands of loops being typical – a random node in the planning tree as it currently exists is chosen for expansion. Choosing this node properly is a bit of an art, and is frequently what differentiates successful from unsuccessful planners. As a practical matter, probabilistic completeness is generally achieved only if the nodes chosen for expansion uniformly span the available region of space-time, whereas choosing a completely random node will tend to cluster the branches very closely together and will not search the space very effectively: As the branches get denser, there are more branches in that region to choose from, which makes it more likely you will choose one of them, and so on. The alternative approach proposed by Kindel and Hsu is to choose nodes with a probability inversely-proportional to their density in space-time. For MRHE we emulate this process by gridding space-time into discrete cells, with the relative probability of choosing a node in that cell inversely proportional to the number of nodes in that cell. This yields a tree that quickly spans space-time, thereby finding many routes to the goal.

After the node for expansion is chosen, the branch is "grown" by randomly applying a thrust direction (in the inertial frame) and a constant random vehicle torque. The thrust magnitude is also a random constant, but the maximum value varies with vehicle velocity such that vehicle maximum velocity constraints are not exceeded. (This simple step cuts computation time in half by eliminating many "out-of-bounds" control inputs.) Branches that result in collisions or other kinematic or dynamic constraint violations are discarded. Once a viable branch is chosen, it added to the tree. Finally, that branch is checked vs. the endgame criteria. In this case, the endgame is satisfied if a single cubic spline in position (i.e. linearly-varying acceleration, or constant jerk) can connect the end of the branch to the goal location while passing the dynamic, kinematic, and collision constraint criteria. Any complete trajectories are added to a list of viable options, and the minimum-fuel trajectory is returned upon planner completion. Note that proper endgame criteria selection is a second factor that can have tremendous influence on the speed and convergence properties of randomized planners, and can vary significantly depending upon the physical situation.

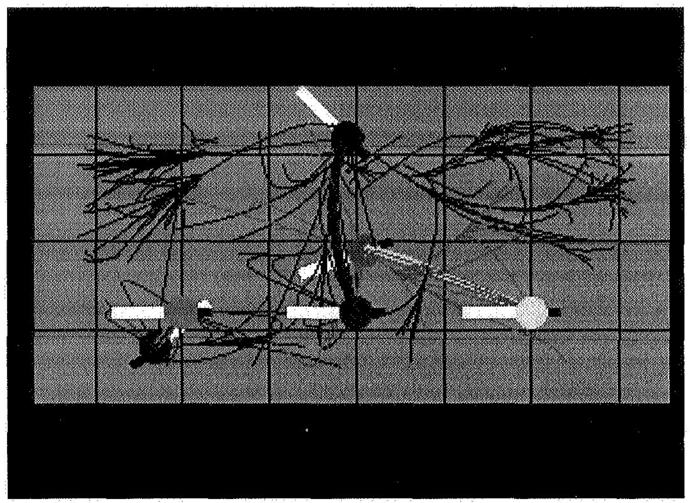


Figure 9. Motion plan

Motion planning with extended objects – such as the vehicles with booms extended – presents some interesting challenges. One important issue is how to efficiently handle collision checking. Approximating the bounds of each vehicle, component, or appendage with a collection of bounding spheres is a relatively simple and effective approach, because collision checking between spheres is a trivial operation. For MRHE we bound each vehicle body with a single oversized sphere, and each boom with a number of smaller overlapping spheres. For more-complicated geometries we recommend the approach of Quinlan,[8] who presents an efficient algorithm that recursively covers an arbitrary object with a binary tree of progressively smaller spheres. A second issue associated with extended objects is the proper choice of random rotations during the planning process. Some random rotational degree of freedom is necessary in order to move the vehicles through smaller passages. Too much allowed rotation, however, is disconcerting to human observers and may be undesirable for other reasons. Because these particular vehicles have significantly greater control authority in heading than in position, we find that it is necessary to restrict the allowed rotational control to prevent trajectories with extreme and inappropriate rotational components.

Figure 9 shows the user interface display of the complete planning trees returned by the motion planner during one run of the MRHE demonstration scenario. Each vehicle is represented by a color-coded circular disk, with a black bar representing the docking latch and a while bar representing the boom. The darker versions of the vehicle icons correspond to the initial locations and orientations, while the brighter versions represent the goal locations (all in a line). The trees of explored motion segments (narrow, dark lines) are clearly visible, as are the final chosen trajectories (thicker, brighter lines). The tree for the blue vehicle is the most expansive because it planned last, requiring it to dodge the other vehicles and resulting in a more-difficult planning task.

E. Vehicle Controllers

Although the vehicles in the Controls and Automation Lab float on an air-bearing surface to emulate zero-gravity conditions in two-dimensions, this emulation is not complete with respect to on-orbit relative dynamics. Such dynamics are frequently described in terms of the Clohessy-Wiltshire or Hill's equations for near-circular orbits[9] and with Lawden's equations for elliptical orbits,[10,11] and can result in very counter-intuitive relative motions arising from selected control inputs. Tillerson[12] provides a good summary of relevant work in related orbital dynamics. Because the specific equations will vary with the orbital conditions, and because of the difficulty associated with trying to emulate these orbital effects in the lab, the vehicle controllers merely work with the standard non-rotating Newtonian reference frame attached to the table. While this does simplify the controller design, it does not significantly impact the higher-level task and configuration management functions that forms the core of the demonstration sequence.

The regulators used for the vehicles are full-state regulators designed using standard Discrete Linear Quadratic Regulator (DLQR) methods.[13,14] Input error is position and velocity error (x, y, theta), and output is an applied thrust vector and reaction wheel torque. Position error comes from comparing the upward-looking camera data with the current position setpoint, while the current velocity error is estimated using a corresponding discrete Kalman filter. In addition to the position and velocity feedback, these regulators have been enhanced to include optional integral terms with saturation cutoffs and optional position and velocity deadbands. Each of these parameters is independently configurable and can be changed on the fly to modify behavior. Each regulator and estimator includes all three degrees of freedom. Although this is not necessary for the solo vehicles because the axes can be decoupled, it makes it more convenient to implement the partially-coupled multi-body estimators and regulators necessary when the vehicles are docked.

The use of Linear Quadratic Gaussian (LQG) techniques was chosen because of the ease it provides towards generating Multiple-Input Multiple Output (MIMO) controllers for arbitrary vehicle configurations. Using known mass and inertia properties, available control authority, and sensor noise, the method quickly provides a well-balanced controller that is executable on the hardware at hand. Fuel use can be easily optimized with respect to desired performance, a feature of great value when using robots with limited fuel supplies. The method does require an estimate of process noise, but once an educated guess has been verified it can be used again and again even if other parameters change. The greatest disadvantage with LQG techniques is that the (nonlinear) optimization only holds for a given physical configuration, and becomes invalid if attempts are made to non-dimensionalize the controller and apply it to differing vehicle mass and inertias. This is a problem for the current project, wherein docking and boom deployment can vary these parameters greatly. In order to retain some degree of optimality, the approach taken is to store a number of regulators and estimators that have been pre-computed for solo, 2-vehicle, and 3-vehicle combinations with various boom lengths, and then to choose the closest match in a process analogous to gain scheduling.

The controller sends the output of the regulators to the vehicle actuators. Desired torque is sent directly to the reaction wheel, while the thrust vector is sent through an optimized thrust mapper and a software pulse-width modulation module before application to the thrusters. For solo vehicles, the reaction wheel alone is used to generate torque in order to minimize fuel use.

F. Adaptive Multibody Controllers

Autonomous control reconfiguration to account for changes in vehicle topology and dynamics is one of the key functional capabilities of the MRHE technology demonstration testbed. The first requirement for such reconfiguration is knowledge of the current vehicle and system configuration. In MRHE we accomplish this through the use of a software entity called the Connectivity Manager (CM). The CM has an internal model of the physical connectivity of itself and all other vehicles: in other words, which vehicles are moving solo and which are currently docked to each other, and in what relative location.

Although the vehicles have enough sensory capability to detect when they have latched onto another vehicle, they cannot explicitly tell onto which vehicle they are latched. The CM resolves this ambiguity by broadcasting the completion of any latch or unlatch activities to all other vehicles. The CMs on these other vehicles, in turn, compare the broadcast location of the latching vehicle to their own current location to determine whether they are the target of the latching operation. They then update their own local connectivity map with the new information about the event, and broadcast the map to the other vehicles, which in turn update their local maps. Received changes in connectivity are retained as long as they are consistent with the local map; otherwise they are discarded. Consistency is defined as states differing by a single latching operation. Repeated broadcasts of the current state of each latch insures that the vehicle connectivity maps stay synchronized, even if vehicles come online or offline while latched to each other. In such a case, connectivity changes iteratively travel from vehicle to vehicle until synchronization is achieved.

Once the connectivity map is established, each vehicle uses its own location in the map to determine its role: solo, master in an aggregate structure, or slave in an aggregate structure. This in turn dictates which controller is applied. If the vehicle is unattached and solo, the regulators and estimators used are those described previously. If it is in a designated slave position, it turns off its local controller and instead receives control commands from the master vehicle. If it is in a designated master location (head of a two-vehicle aggregate, or in the middle of a three-vehicle aggregate) then the appropriate multi-body regulators and estimators are employed. The gains for these components are chosen depending on the known mass and inertia of the vehicles in the aggregate and the current length of the connecting booms.

The multi-body regulators and estimators are similar to the solo versions described above, with the exception that each vehicle's thrusters and reaction wheel are available for actuation. Thus a three-vehicle aggregate can use up to three independent thrust vectors applied at the individual vehicle center of gravity locations, and the sum total of all three reaction wheels. This effectively allows the use of thrusters for attitude control. Although this uses more fuel, it is necessary for aggregate motion because the increased inertia – up to several hundred times larger than a solo vehicle in the case of a 3-vehicle aggregate with full boom extension – is well beyond the control authority of the reaction wheels.

Sensing and estimation for the aggregate vehicle is accomplished using the sensors on the lead vehicle only, and the setpoint is applied at that vehicle center of gravity. For regulation, both the vehicle and setpoint positions and velocities are then translated to the vehicle center of gravity to compute the error of the aggregate vehicle. This two-step process, although somewhat cumbersome, has the advantage of maintaining a continuous setpoint and seamless transition when a vehicle is stationkeeping during a docking or undocking operation.

G. Task Engine

The final major component of the vehicle controller architecture is the task engine, which is responsible for all event sequencing throughout the demonstration. The structure of the task engine is primarily that of a sequenced list. Each vehicle maintains its own task sequence, which can be started, stopped, appended to, or dynamically modified at any time. Jumping to out-of-sequence tasks is also supported. The sequence can be set to advance upon expiration of a timer, upon completion of some event, or a combination of the two. In general, each vehicle will have a parallel, although potentially different, set of tasks within its task sequence. Simultaneous advancement through the task lists is accomplished by designating one vehicle as the lead for each particular task entry. Upon task completion, the lead vehicle then radios the other vehicles to advance to the next task.

Each task is itself implemented as a software object, derived from a virtual base class that incorporates the hooks operated upon by the task engine. In addition to its update routine, each task has an entry and an exit routine to handle variable initialization, controller mode switching, and other one-time operations associated with the task. These are analogous to the entry and exit routines commonly used in finite state machines. The tasks implemented for the MRHE demonstration are Idling (passively waiting for additional commands), Stationkeeping, Trajectory Following, Docking, and Boom Deployment.

The sequential task engine described herein is merely the first step towards an autonomous high-level task-engine capable of replanning operations in response to changing situations or fault conditions. Future projects in the Controls and Automation Lab will incorporate a much more advanced autonomous planning architecture.

V. MRHE Concept Demonstration

The MRHE concept demonstration was designed to demonstrate the feasibility of the fundamental technologies necessary to autonomously assemble large space structure and to show initial implementation of a subset of these technologies on a representative testbed. The selected technologies include autonomous path planning, rendezvous, and docking; automated reconfiguration of the attitude and position control system from the pre- to post-docked configuration; automated deployment and retraction of ATK-supplied Collapsible Rollable Tube (CRT) booms; and, at a rudimentary level, automated detection and recovery from a failed docking attempt.

The final demonstration sequence is executed in a fully autonomous manner by the vehicles. As was described above, each vehicle has a sequential list of tasks and behaviors, together with associated entry and exit criteria and actions for each task. Inter-vehicle messages keep the vehicles synchronized and advancing from task to task in a unified manner. During the course of this sequence, any vehicle that is not actively involved in a given maneuver or task is instead stationkeeping at a prescribed location. The ground station operator initiates the planning and the execution of the sequence, but thereafter does not need to intervene with the system and the vehicles operate in a fully-autonomous manner.

Although the entire assembly sequence has been successfully performed numerous times on the laboratory hardware, the sequence steps in Fig. 10 are presented in graphical form for clarity.

A. Motion Planning to Goal Locations

At the beginning of the demonstration, each vehicle starts in an unknown random location and orientation (Fig. 10a). After determining their location and heading on the table, they radio this information to the system ground station computer. The ground station uses the Randomized Kinodynamic Motion Planner to plan paths for each vehicle to its prescribed goal location while avoiding all potential obstacles, and radios these plans to the vehicles. In this case, the prescribed goal configuration is a line of three evenly-spaced vehicles, all facing the same direction. Because of the boom assemblies and latches protruding from the vehicles and the confined maneuvering space available on the table, the rotational motion and collision checking in the planner is just as important as the translational motion. Although the motion planner currently resides on the ground station for simplicity, it could also be distributed onboard the vehicles themselves in either a centralized or decentralized implementation.

B. Trajectory Following to Goal Locations

At a command from the ground station, each vehicle then proceeds along the trajectory plan to the desired goal location (Fig. 10b).

C. Failed/Aborted Docking Attempt

The first docking maneuver is now attempted (Fig. 10c). The docking vehicle (on the left) switches to its side-looking docking camera, and controls itself with respect to the LED docking target on the target vehicle. In this case the docking fails because this particular target vehicle purposely does not have the appropriate docking probe, effectively mimicking a hardware failure that the system must first detect and then recover from. The docking vehicle times out of its docking attempt, determines that docking was unsuccessful, and automatically executes a back-off maneuver to put a safety buffer between it and the target.

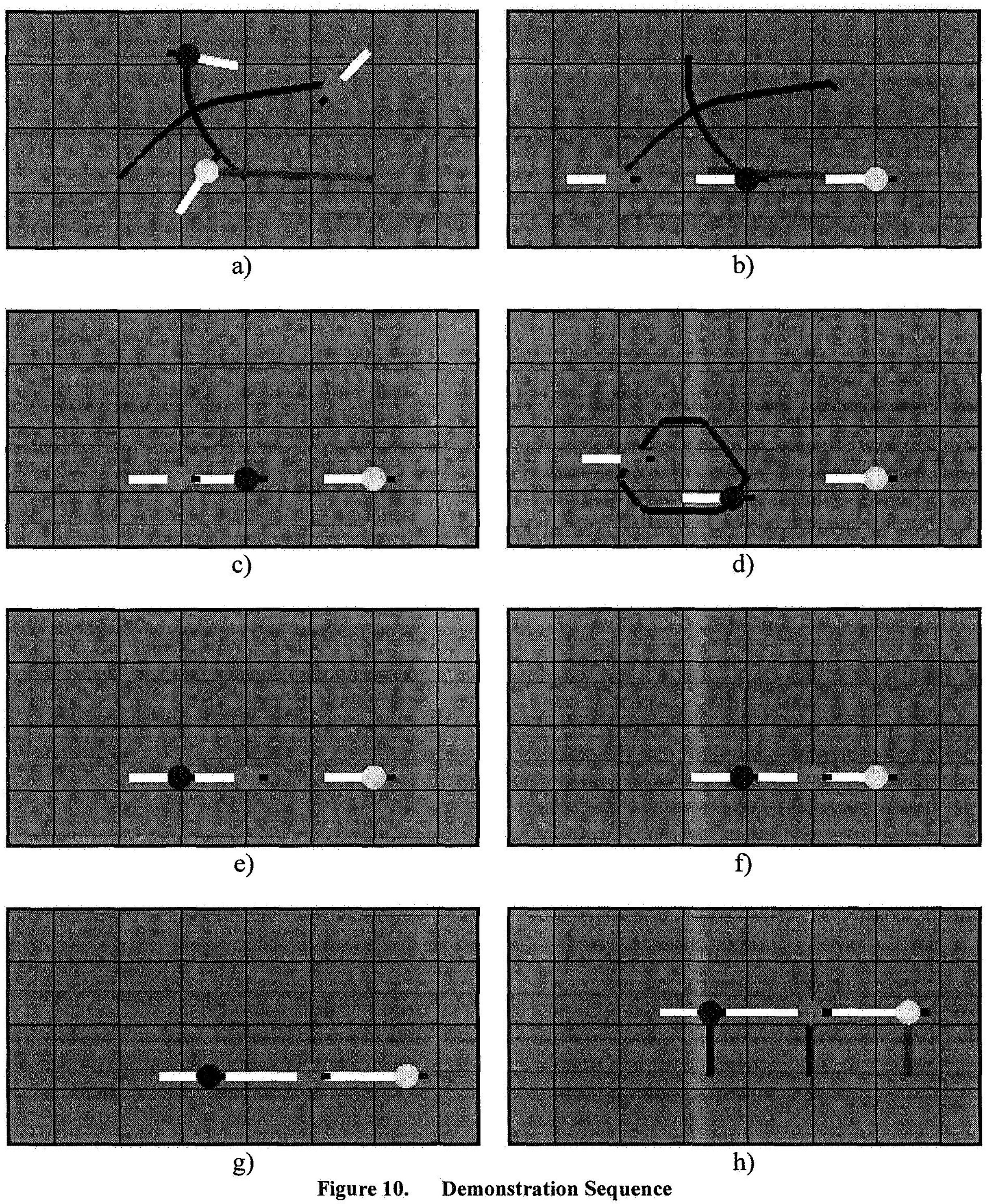


Figure 10. Demonstration Sequence

D. Vehicle Location and Role Exchange

Because the previous docking attempt could not take place, the two vehicles now change locations and switch roles, docker vs. target, for a new docking attempt (Fig. 10d). The rational behind this maneuver is that if the vehicles are truly modular, then any other vehicle can take the place of a damaged or non-functional one in the event of a failure. In this particular demonstration the maneuver and system reconfiguration is scripted, and is effectively hard-coded within the vehicle task engines. Development of fully-autonomous fault-detection and correction strategies is the subject of future development using a more-adaptive planning architecture; therefore this "fault-detection" behavior is currently for demonstration purposes only.

E. Docking with Controller Reconfiguration

After the positional and role reorganization above, the demonstration sequence proceeds as if the task sequence had been uninterrupted, but with two of the vehicles now in different roles. The new docking vehicle now docks with the new target vehicle (Fig. 10e). After successful docking, the docking vehicle examines sensors in its latch mechanism to determine that it is now latched onto another vehicle, and announces this event to the other vehicles. The vehicles then compare their knowledge of their relative physical "connectivity" so that each vehicle knows who is connected to whom.

Using this connectivity information, the docking vehicle relinquishes control to the target vehicle it has docked to. The target vehicle (on the right) assumes the leading role in a master-slave control configuration, and reconfigures its regulators and estimators to account for the new mass and inertia properties of the aggregate vehicle, the change in center of mass location, and the additional actuators available to it from the second vehicle. Thereafter the lead vehicle computes the necessary control for both itself and the follower vehicle, and radios to it the desired reaction wheel and thruster commands.

F. Cooperative Docking

The two-vehicle aggregate now docks with the third vehicle using the new master-slave controller configuration (Fig. 10f). After docking the vehicles again compare connectivity and reconfigure their respective control strategies accordingly. In this case, the center vehicle retains control of the aggregate, while the two end vehicles act as followers.

G. Boom Deployment

The deployment mechanisms now synchronously deploy the booms connecting the three vehicles, half a meter each to obtain a center-to-center spacing of over 1.5 meters (Fig. 10g). Note that the leftmost vehicle does not have a boom to deploy. During the deployment, the lead vehicle senses the boom length and adjusts its control gains to accommodate the change in vehicle inertia and maintain stability, while stationkeeping at its current location. Although the current booms on the vehicles allow for over 1.5 meters of deployment each, the combined length of the aggregate vehicle in the case of full deployment (over 6 meters) would not give it sufficient room to maneuver on the table.

H. Multi-Body Lateral Translation

The aggregate vehicle translates 1.0 meters laterally, followed by stationkeeping at the new location (Fig. 10h). Each vehicle contributes to the control necessary for the maneuver, with the end vehicles receiving appropriate control commands via radio link from the leading center vehicle.

I. Boom Retraction

Following the conclusion of the fully-autonomous sequence, the booms are retracted to their stowed configurations. This action is again coordinated by the leading center vehicle.

VI.□ Conclusion

The modular and reconfigurable spacecraft architecture developed during the MRHE project represents one possible way to leverage the benefits associated with modular hardware to get large payloads to distant orbits in a cost-effective manner. Additional capabilities enabled by this approach include the ability to replace components and/or elementary spacecraft buses on-orbit in order to replace faulty hardware, upgrade instruments, or reconfigure mission objectives. The assembly sequence presented herein experimentally demonstrates in a laboratory setting a number of the rudimentary capabilities necessary for the construction of such an aggregate spacecraft including motion planning, rendezvous and proximity operations, docking operations, and controller reconfiguration. Of special interest is the integration of the new Collapsible Rollable Tube technology booms and deployment mechanisms into the vehicle testbed, a technology concept that has significant flight potential.

Although the MRHE program ended in 2006, the completed demonstration sequence gives a tantalizing view of the potential for such spacecraft operations in the future. Development of related technologies continues at both NASA and its industry partners. At Lockheed Martin, current work at the Controls and Automation Lab is focusing on formation flying by outfitting the test vehicles with an onboard inter-vehicle interferometer to enable optical-level metrology and formation control. Other work is focused on the areas of intelligent task planning, software modularity, and distributed autonomy.

Acknowledgments

Funding for the technical work and hardware infrastructure presented in this paper has come from a number of different sources. The bulk of the design of the demonstration system and the related software development was conducted by Lockheed Martin under a NASA Human and Robotic and Technology: Technology Maturation Program contract with NASA Marshall Space Flight Center (Modular Reconfigurable High-Energy Technology Demonstrator Program – NASA Contract NNM05AA99C). Development of the docking interfaces and sensors by Lockheed Martin, as well as the presented Collapsable Rollable Tube boom technology by ATK, was also performed under this contract. The authors would like to thank NASA for funding this work and enabling its publication. The Controls and Automation Laboratory infrastructure including the lab space, robotic vehicles, overhead starfield sensor system, boom deployment mechanism, and lower-level vehicle software was all developed by Lockheed Martin under internal funding, and represents an ongoing investment in furthering Lockheed Martin's capabilities in the areas of in-space robotics and autonomous operations. This project could not have been successfully completed without the hard work and dedication of a number of people at the Lockheed Martin Advanced Technology Center including – but not limited to – Fred Rehnmark, Buck Holmes, Kyle Brookes, Greg Burton, Fred Baisa, and Mike Bishop.

References

[1]Mitchel, I. T., Gorton, T. B., Taskov, K., Drews, M. E., Luckey, D., Osborne, M. L., Page, L. A., Noris, H. Lee. III, Shepperd, S. W., "GN&C Development of the XSS-11 Micro-Satellite for Autonomous Rendezvous and Proximity Operations", *Proceedings of the 19th Annual AAS Guidance and Control Conference*, AAS, San Diego, CA, 2006.

[2]Lightsey, G. E., Campbell, T., Mogensen, A., Burkhart, P. D., Ely, T., and Duncan, C., "Expected Performance of the Electra Transceiver for Mars Missions", *Proceedings of the 2005 AIAA Guidance, Navigation, and Control Conference*, AIAA, Washington, DC, 2005.

[3]Carpenter, J. R., Folta, D. C., Moreau, M. C., and Quinn, D. A., "Libration Point Navigation Concepts Supporting the Vision For Space Exploration", *Proceedings of the 2004 AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, AIAA, Washington, DC, 2004.

[4]Stadter, P. A., Asher, M. S., Kusterer, T. L., Moore, G. T., Watson, D. P., Pekala, M. E., Harris, A. J., and Bristow, J. O., "Half-Duplex Relative Navigation and Flight Autonomy for Distributed Spacecraft Systems", *Proceedings of the 2004 IEEE Aerospace Conference*, IEEE, New York, 2004, pp. 565-573.

[5]Inalhan, G., Busse, F., and How, J. "Precise Formation Flying Control of Multiple Spacecraft Using Carrier-phase Differential GPS", *Proceedings of the 2000 AAS/AIAA Space Flight Mechanics Conference*, AAS, San Diego, CA, January 2000.

[6]Kindel, R., Hsu, D., Latombe, J., and Rock, S., "Kinodynamic Motion Planning Amidst Moving Obstacles", *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, Vol. 1, IEEE, New York, April 2000, pp. 537-543.

[7]Hsu, D., Kindel, R., Latombe, J., and Rock, S., "Randomized Kinodynamic Motion Planning with Moving Obstacles", *International Journal of Robotics Research*, Vol. 21(3), Sage Publications, Thousand Oaks, CA, Mar 2002, pp. 233-255.

[8]Quinlan, S., "Efficient Distance Computation Between Non-Convex Objects", *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, Vol. 4, IEEE, New York, 1994, pp. 3324-3329.

[9]Kaplan, M., *Modern Spacecraft Dynamics and Control*, John Wiley & Sons, New York, 1976, Chap. 3.

[10]Lawden, D., *Optimal Trajectories for Space Navigation*, Butterworths, London, 1963.

[11]Carter, T., and Humi, M., "Fuel-Optimal Rendezvous Near a Point in General Keplerian Orbit," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 6, AIAA, Washington, DC, 1987, pp. 567-573.

[12]Tillerson, M., *Coordination and Control of Multiple Spacecraft using Convex Optimization Techniques*, S.M. Dissertation, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, June 2002.

[13]Gelb, A. (ed.), *Applied Optimal Estimation*, The M.I.T. Press, Cambridge, MA, 1974.

[14]Franklin, G., Powell, J., and Workman, M., *Digital Control of Dynamic Systems*, 2nd ed., Addison-Wesley, Reading, MA, 1990.