System Design of Robots for Application to In-Space Assembly

Harshit Suri, Peter Will, Wei-Min Shen
University of Southern California
Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292, USA
harshitsuri@ieee.org, will@isi.edu, shen@isi.edu

Abstract - **This paper presents the design of an experimental system for assembly applications in space. The prototypical application is the assembly of mechanical trusses. The system used an air-hockey table to simulate a frictionless two-dimensional space. Assembly robots fly on the surface finding, gathering and assembling the relevant parts to perform the construction. The system design involved building the FIMER Robots, the test bed, the sensing system for position and velocity feedback and the control scheme. This paper describes the hardware and software used in the various sub-systems and includes calibrations and measurements and the results of experiments.**

Index Terms – Self-Assembly and Space Robots.

I. Introduction

This work was done as part of a project (FIMER, Free-flying Intelligent Match-makER robots, NSF/NASA Grant No. 0233364) to study the feasibility of techniques for the assembly of large structures in space; in this particular case, structures that would be capable of carrying very large arrays of solar cells and beaming the collected power back to earth. The basic premise of in-space assembly is that those structures would be far too large to be put in space in any single launch and thus would have to be assembled in situ from components and subsystems that had previously been placed in space. A prior paper described the use of the FIMER system in an assembly application [3]. The discussion here is focused on describing the architecture and design of the underlying system that allowed that and other work to be done

Our methods for assembly in space are based on technologies developed for our CONRO self-reconfigurable robots [1,2]. Technologies from other reconfigurable robot projects could be applied to the problem but CONRO technology was local, developed and convenient.

We chose the beam as the component for the in-space assembly structure since it is the most primitive structural element. Beams are then assembled into triangles. The triangles are then made into trusses and trusses can be made into larger structures. Two FIMER robots flying cooperatively and emulating two spacecraft assemble the beams. Both beams and spacecraft are equipped with docking stations that are the exact docks used in the CONRO robot [4,5]. These connectors were designed so that any two connectors can dock or de-dock at will. Here, rather than grasping each beam, the spacecraft fly to the beam, attach to it using the CONRO dock to form a unit, fly to rendezvous with another spacecraft carrying another beam or a structure, add its beam, undock, and then fly off to collect another beam to enlarge the assembly

The body of this paper deals with the design of an experimental systems comprising an air-table to simulate zero-G space operation, two assembly robots (FIMERS) and dock equipped beams plus the sensing, actuation and control methods needed to perform the assembly.

A. Assembly and FIMER Philosophy

One of the major lessons from work in automated assembly is that the critical issue is to keep the component parts to be assembled under control at all times. On earth, we do this by keeping parts oriented by feeders and trays using gravity to aid in holding them in place. In space, in zero gravity, this is not possible in the same form.

Astronauts doing in-space assembly are tethered against flying off into space. For similarly reasons, and others to be discussed below, we feel it is desirable to tether the FIMER assembly robots.

Tethers are known to have complex behaviours in space [10]. In major differences from prior work on tethering[1] where they are used as power generators or as assists in raising orbits, the tethers in FIMER play an *active* role in assembly. Tether entanglement is a serious issue in practice especially in 3-D but the use of tethers is worth investigating in 2-D because of its interesting properties (see the paper [6] describing the use of FIMERs in an assembly application).

The FIMER robots, although tethered, can fly autonomously and communicate with other robots. In flying they use thrust and expend power. In space the thrust would come from cold gas or other devices. FIMERs use the tether not only for keeping connected but also as a source of thrust (or more correctly) pull through a reel mechanism that can impart tension on the tether. Pulling on a tight tether imparts acceleration to both the object pulling and, reactively, to the object being pulled. Pushing a string is not advisable. Deceleration always requires expending fuel. In a typical assembly task with the sub-tasks; accelerate to go to get a part; decelerate to grasp; accelerate to fly to assemble; decelerate to assemble; pulling on the tether can save conceptually the "accelerate to assemble" stage which promises to be ~¼ of the total power expended. This savings is worth investigating but more important is the over-all capability of keeping parts under control that a tether gives to the assembly process.

B. Emulating Zero Gravity

The workspace for FIMER assembly is a commercial, several hundred dollar, air-hockey table that works by blowing

[1] The theory, design and use of tethers in space is a rich subject but outside of the scope of this paper

air upwards to float an object placed on the surface to give inexpensive zero-gravity solution. A better but much more expensive, several thousand dollar solution, is to use a polished granite block with air blown downwards from the spacecraft as used in NASA. Overhead cameras connected to a base station monitor the workspace and give the position and orientation of spacecraft in the environment.

The first task was to setup the airbed mechanical system used to test the FIMER robots. The basic principle, as in a hovercraft, uses a plenum chamber to trap air under the object. A hovercraft uses a skirt around the perimeter to create a chamber to contain the pressurized air and blows air downwards into the chamber. The pressure differential causes a positive lift to be produced which counters gravity. Since the object is 'flying' on air, it experiences very low lateral friction. We tried different geometries of the skirt around the plenum chamber and also different materials. Empirically we found that the flatness of the material that floats over the air surface is the most important parameter. A simple flat piece of glass worked very well without requiring a skirt and was able to support a large load. The shape of the glass bottom was unimportant for levitation, initially we used a square piece glass bottom to match the square array of holes on the air table, but a focus on the assembly task led us to use a circular bottom plate. The circular base made a very convenient geometric assembly reference.

The choice of an air jockey table as the base, though a low cost approach, gave major experimental problems and this should be taken as a warning to future experimenters. A non-horizontal table surface causes the robot motion to be biased towards one side of the table but this can be corrected easily. Our table has spatially noisy airflow that could not be corrected easily and this caused a non-actuated robot to move around randomly on the surface unlike what would occur in space. The result of these effects was the addition of a significant amount of environmental noise that had to be overcome by the control system. The solution was to make the force generated by the fans large enough to counteract these perturbations We also provided a large dynamic bandwidth and fast changes in fan speed/force levels very quickly and these design choices aided in system control.

The net effect of the perturbations, and the mode of operation required in the control system, was to make the system somewhat unrealistic as a detailed space simulation. In space (even in LEO: Low Earth Orbits) friction and perturbations are to a first order negligible. Spacecraft need very small amounts of delta V to do station keeping as compared to the thrusts required for normal orbital maneuvers. Control is done in the impulse mode. However in our case, the large perturbations caused the control equations to deviate from the simpler forms used in normal attitude control and forced us to operate in continuous mode.

Given this restriction that caused us to solve a harder than usual problem and operate in an unrealistic control mode, FIMERS operate effectively.

II. FIMER

Figure 1 shows two FIMER spacecraft robots connected by the reel-able tether and with a beam docked to each FIMER via CONRO docks. Each FIMER is circular and about 25 cm in diameter and is supported by a flat base that in turn floats on the up-currents from the air-table. FIMERs are battery powered and are connected to a base station by radio for control. Communication between them is obtained by sending messages via the base station.

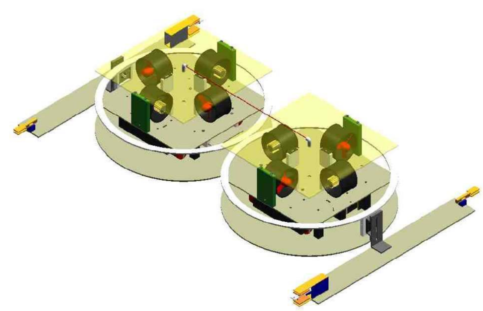

Fig. 1 Two FIMER robots and attached beams.

Each FIMER has 4 orthogonal fans as shown. The fans emulate the cold nitrogen gas thrusters used in space and, as in spacecraft, are arranged to thrust tangentially. Each fan can run forward and backward so that the various combinations of thruster actuations allow forward and backward motion in each of the canonical directions. Rotation about the center of gravity is obtained by running all fans simultaneously at the same calibrated thrust. (Experiments were done on using the minimum set of two fans but the servo and control complexity caused by non-holonomicity was distracting to the overall assembly goal.)

Spacecraft attitude control is derived from an overhead camera connected to the base station. The camera measures FIMER position and orientation plus their derivatives. Control commands for assembly and docking are sent from the base station for execution by the servomotors running fans and the various other on-board command and control systems on FIMER.

A. *Docks and Docking*

The FIMERS and all the structures to be manipulated by FIMERS were fitted with docking mechanisms. The FIMERS carried the male docks and the other parts carried female docks. Hermaphroditic docks would have been preferred but the docks used, Figure 2, had a long history of successful use in the CONRO project [4].

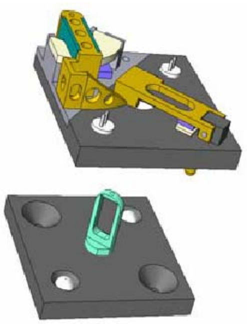

Fig. 2. Docks Female (top) and Male (bottom).

Docking is done in stages. First the two objects, FIMER and the target are moved relatively coarsely to an approach point from which, secondly, a fine motion is executed. The approach point is usually made so that the fine motion is made as orthogonally as possible to the plane of the dock. CONRO docks shown can accommodate approximately 10 degrees of misalignment.

B. *Optical Target*

A camera looking downward provided synoptic vision. Each FIMER had a target place on it that was visible to the camera. The target form is shown in Figure 3 and identifies both the center of gravity and a canonical direction. The image processing algorithms thus determined the location and orientation of FIMERS and both implicitly and explicitly the carried beams. The locations and orientations were sent wirelessly from the base station-to the FIMERS and incorporated into control loops for positioning the FIMERS and doing the assembly.

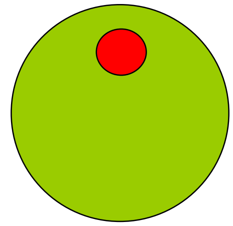

Fig. 3. The target.

The FIMERS also had a reel-in mechanism using a servo-controlled take-up spool fitted with a tether tension sensor. The control system for this mechanism, actuated on command, was a simple PID controller.

C. *Vision for Position and Attitude Control*

The vision system was the primary position and velocity sensor for FIMER and images the robots from an elevation of around 6 feet above the air hockey table (~6ft x ~3.5ft), so a single camera sufficed. The camera was a Unibrain Fire-i with a resolution of 659 x 494 and a Firewire interface to a PC.

The vision task was the detection of the position and heading information of the two FIMERs from their colored targets shown above. The system was iteratively optimized for speed to give a fast ‘enough’ iteration rate for the closed loop position system. The camera could sample up to 60Hz however the processing needed for two robots limits this cycle time to about 15 frames per second. This is above the Nyquist frequency of the system and is fast enough to keep the entire control system stable.

The vision system used software developed in-house using open source video libraries to find the centre of mass of the targets placed on the robots. Two different color schemes are used to differentiate between the robots. The targets used two circles, one small one inside the other (non-concentric) to give both position and orientation. The image is raster scanned and edge detected to find the change in colors. The algorithm then finds the circles, both inner and outer. It then computes the centre of mass of each circle and then reports the 4 coordinates as a return value. The vector between their centers gives the direction vector of the robots. The rate of change of linear distances of the centre gave the velocities in the X-Y plane. We used a simple first order differential approximation of the rate of change of the angle to get the angular velocities. The overall cycle time included frame capture, IO for camera-PC, control algorithm execution and transmission of the final command to the remote robot. The detected resolution referred to the FIMER was 8mm. This accuracy (plus white noise) was sufficient for the FIMER spacecraft to home into and dock with a beam using the CONRO docking connectors.

D. *Thruster and FIIMER Calibration*

Fans were used to emulate the thrusters. The fans were model aircraft parts and were advertised to be able to produce up to 0.009N of thrust, enough to produce lift for a medium sized model aircraft. The profile of the vanes is such that thrust produced in the forward direction is much greater than the thrust produced in the reverse direction. The fans were selected to produce enough thrust to overcome the large spatially dispersed background effects in the air hockey table. Unlike real spacecraft thrusters, fans have a non-linear relationship between rotation velocity and thrust. The map was measured in a test rig similar to those used to calibrate micro-thrusters for space. A speed versus force profile was measured for the fan in both directions of rotation. The fans were measured using the same circuitry and control mechanism (PWM) as in the actual board for accuracy. The graph in Figure 4 demonstrates the non-linear thrust Vs RPM relationship of the fans.

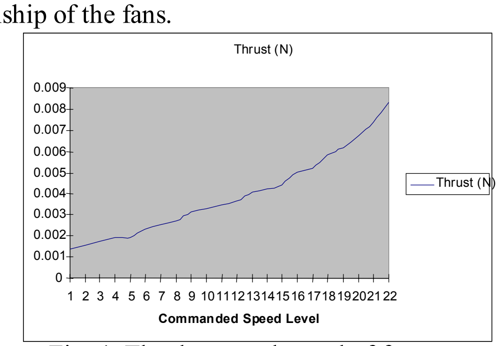

Fig. 4. The thrust and speed of fans.

We assume that once the fans speed up to a certain rpm the thrust produced is constant. Transient thrust produced when the fans were accelerating was not considered deeply. The control algorithm maintains a map of the commanded air speed versus the actual thrust produced in Newtons in order to compute the needed thrust. A look up table is employed to find the index speed value necessary to produce the required force and the look-up value is sent to the robot wirelessly to produce it. The control law produces a force vector that is decomposed into X and Y force components. Note that yaw (torques about the Z axis) are also decomposed into X and Y forces. Yaw actually needs only two fans to be active and four are available.

FIMER as in normal spacecraft requires balancing. The moment of inertia about the Z axis (axis of rotation) of FIMER was also required for both Matlab simulations of the control algorithm and the actual servo code. The method proposed by [Richard Murray MVWT] was used in this measurement. FIMER was hung from a nail and then nudged to start oscillations. The moment of inertia was calculated using

$$I = mL^2(((gT^2)/(4\pi^2 L)) - 1)$$

where L is the distance from axis of rotation to the FIMER center of gravity. T is the time period of the oscillations. 'g' is the constant of acceleration. 'm' is the mass of the robot. The moment of inertia was found to be 0.0055192 ($kg.m^2$). The mass of the robot was measured to be 0.2695 kg.

Our spacecraft, FIMER, had to be dynamically balanced. Geometrically the FIMER centre of gravity had to coincide with its axis of rotation. Since the components were not homogenous we had to adjust the robot so that the COG went through the axis of rotation. We did this by first locating geometrical centre of symmetry. The fans were located with good precision about the axis of symmetry. We then knew where the COG should be. A balancing mass (a piece of brass) was added appropriately so that the COG passed through the center of gravity. In actual spacecrafts dynamic spin balancing is used to check weight, center-of-gravity, dynamic balance, and moment-of-inertia [8]. Static balancing was sufficient for FIMER.

E. Control

A three dimensional rigid body has 6 degrees of freedom in free space. These are X, Y, Z and the 3 rotations about the 3 axes. In space applications the attitude of the spacecraft is controlled by impulses of force. The impulses are obtained from many, at least 6 but usually more than a dozen impulse generators, either gas jets or reaction wheels. The pure friction-free dynamics of in-space operation do not apply well enough to air table operation because of imperfections in the table and its interaction with FIMER. This meant that FIMER had to be driven by continuous not impulsive control. The end result, servo control, was therefore dynamically realistic in the end result but no comparison relating to "fuel consumption" should be made.

In our planar case we had only 3 degrees of freedom, X, Y and rotation about the Z-axis. We used 4 X and Y thrusters placed symmetrically as shown in Figure 2, They operated independently and the same thrusters were used also to produce a yaw (rotation about Z axis). Simple aero-hobbyist fans were used to generate thrust. They were driven by on-board servos that were calibrated to provide thrust in forward and backward directions for effective spacecraft control. The calibration process included consideration of the fan characteristics; the initial lack of closed loop speed control of the fans, the asymmetry of the fan blades in the forward and backward direction and the discharge rate of the motor drive batteries. In addition different supposedly identical fans produced different torques even at the same commanded drive speed. This was corrected in 2 ways. First the fans were servo-controlled and calibration constants were maintained for each fan for closed loop commanded drive speed versus the force in Newtons and in a second-level control loop, external synoptic vision from an overhead camera allowed actuating one of the fans at a higher command speed than the other to decrease torque during a motion to achieve the correct trajectory.

A simple but well-damped PID controller drove each FIMER with the position and velocity in translation and rotation given by the overhead camera and the fans providing calibrated thrust. The commended motions were segments of straight lines interspersed with commanded rotations. The prototypical action was to go to an approach point clear of the target, stop, align rotationally and then move forward slowly to dock.

III. System Description

The hardware for the PC Host side was a Pentium III with a firewire and UART port. The firewire port interfaced to the camera, and a serial port interfaced via a RS232 transceiver chip to a wireless transmitter for sending signals to the FIMER robots. The host software grabbed a frame from the camera, processed it and sent the positions and orientations of the targets on the robots. A first derivative in time of the linear and angular positions gave the linear and angular velocities. The combined error signal was translated to a motor actuation value and sent over the wireless for actuation by the microprocessor. The vision system could also detect that the robots had reached a certain stage in the construction process and could instruct the robots to start a new movement using the reel-in mechanism.

The requirements for onboard processing in the robots computer are very small. The robots basically have three major tasks: 1) Actuate and control the thrusters 2) Actuate and control the reel-in mechanism including sensing the tension on the string between the robots 3) Communicate with the host PC. The on-board tasks do not require high computing resources hence we chose the basic STAMP [6] as the single board computer. In turn it uses internally a PIC microcontroller. The development environment was the proprietary Parallax software and PBASIC the programming language. The basic STAMP uses a PIC microcontroller, runs at 20MHz, has 32 Bytes of RAM and has other peripherals and memories interfaced to it. It has 2Kbytes of EEPROM that holds the program to be stored and can be programmed using the serial RS232 connection directly without any transceiver. It has 16 general-purpose I/O pins. It draws 3mA at 5V while running and 50uA during sleeping. It had just sufficient I/O and memory after minimization and packing for the control actuation and decoding commands although more resources would have been preferable.

The majority of the computation was done at the PC host end, which handled the sensing, control algorithm and sending commands over the wireless to the robot for actuation. The STAMP was responsible for handling communications between the processor and the wireless link and controlling the motor drivers. It received and decoded the commands over the wireless channel and then actuated the fans according to the force levels sent by the host PC.

The load on the STAMP made it necessary to off-load the servo driving function to external DS1052 PWM (pulse width modulation) devices that internally used a STAMP. The motor actuator hardware consisted of a four individual Philips PWM generation chips that each implementing one single channel of PWM for the gates of the H-Bridge. They communicated with the main STAMP using the I2C protocol over the two-wire serial interface address bus setting the internal registers of the DS1052 PWM generating chips, one chip per motor being driven. The nominal frequency of the PWM signal was 100KHz. The PWM chips send a signal to the gates of an H-Bridge implemented using MOSFETs. The signal from the PWM to H-Bridge gates went through a de-multiplexer to allow for bi-directional control of the motor direction. The H-Bridges used a dedicated power supply from NiCad rechargeable batteries so that supply to the microprocessor could be kept as isolated as possible from the inductive noise generated by the motors.

The software implemented on the basic STAMP consisted of code implementing the general control pseudo code shown below

1) Listen for preamble.
2) If it is for assigned robot then proceed else go to 1
3) Decode incoming command. If it's a thrust command go to 4 else go to 5
4) Set PWM parameters using the I2C bus. Return to 1
5) Start the closed loop reel in mechanism that uses the optoelectronic sensor and the SMA as an actuation control.
6) Return to 1

The STAMP was also responsible for interfacing to the reel-in mechanism. This controlled by the STAMP actuating the reel based on a binary sensor that indicated whether or not there was tension in the tether that connected the two robots.

A. *Communications*

Communications is required on two levels. One level is between the remote PC host implementing the control algorithm and FIMER. The other was on the FIMER between the micro-controller and the peripheral chips.

We implemented a wireless scheme of communications since the robots were to fly autonomously on the air hockey table There are many commercial-off-the-shelf wireless components available. They include Bluetooth, WLAN etc. However since we had chosen to keep all the processing and IO on the remote PC host low bandwidth yet moderately reliable wireless link filled the need. The communications between the PC and the Robot was done using a RXM Linux (tm) wireless module chip that implemented an amplitude shift keying mechanism i.e. a carrier was sent when input bit was of one polarity and carrier was absent when input bit was of the opposite polarity. The link used ISM frequency of 433 MHz and was running at 2400 Baud.

The serial bus protocol over the wireless link used the RS232 signalling scheme. It used a start bit, 8 data bits and a terminating stop bit to signal end of a data byte. RS232 is a very simple yet robust scheme for low bandwidth asynchronous communications. Asynchronous use allows a clock less system and that in turn makes the system easy to deploy.

The system was deployed without parity checking in order to keep the speed high and to avoid a potential bottleneck in controlling the PWM devices. A delay in the control loop has serious stability implications. A slow communication delay implies a slower control loop cycle time and also increases the transit time between instrumentation and actuation of the system that is being controlled, in this case the robot. The transmitter and receiver mutually controlled the baud rate. The wireless chip simply keyed the carrier onto the digital built stream being sent to it. The baud rate of 2400 bps gave a good trade-off b/w signal integrity and speed.

Noise immunity was increased drastically when the commands were preceded by a simple known preamble. This preamble was required for two purposes. 1) It gave the receiver a way to discriminate between noise and actual data 2) It was used as an address decoder for the robots. Since the Linux modules were simply transmitting on the same channel frequency, we needed a way to talk to multiple robots. This was accomplished by assigning a predetermined address to each robot in the code in their micro-controllers. Both the robots would listen to the channel however they would only be triggered by their respective addresses. The interface to the wireless chips was a simple TTL logic level single wire channel, making it very simple to use. On the robots side, the basic STAMP supported a serial bus command in its instruction set. This basically shifted the digital stream into the wireless chip that in turn modulated the signal and radiated it out through a splatch antenna. The radiation pattern was essentially isotropic and hence independent of the actual direction and position of the antenna. The robots could be moving and facing different directions without loss of communication.

The PC side of the communications had a RS232 transceiver and a wireless transmitter. For the PC side a small board was built that used a RS232 transceiver to convert RS232 signalling to TTL that fed into the wireless chip for further modulation and transmission. The antenna was a simple monopole lamda/4 whip antenna. The antenna could be a directional antenna to concentrate radiated power towards the air hockey table, but the whip antenna was easiest to make and match with the wireless chip. A serial command sent to the UART port of the PC activated the wireless transmitter. The software to implement the communications channel wrote to the UART port of PC using standard Linux API for handling serial communications and was written in C. For the Windows implementation we used a freeware serial bus driver API.

Communications between the microprocessor and the PWM chips was done over a two wire serial interface available in the STAMP. An address bus mechanism was implemented; each peripheral chip assigned an address. The master microprocessor was responsible for generating the clock for the peripheral chips.

One of the problems at the receiver was a switching noise induced from inductive effects of the motors. Good grounding practice was imperative in remedying this source of noise.

IV. Application Results

The FIMER robots were used to assemble a triangle, the base element of a truss, from elementary beams. Figure 5 shows the entire experiment environment with the air-hockey table and the camera on the ceiling. Reference [6] describes the use of the FIMER robots to do a basic construction under closed loop control; First the targets and the robots are acquired by the main controller using vision. Then the robots go and attach themselves to the target truss beams using the docks. This process is called to the 'go-get' stage it relies on global sensing (see Figure 5.1). Once the robots have docked to their respective beams, the main controller commands them to start reeling in. The completion of this task is indicated by the tension of the tether and indicates that they are touching each other on the periphery of the circular base plate (see Figure 5.2). The main controller through the vision system also senses this condition. The robots now hold the tether taut and start rotating. In this mode the robots rotate about their centers so that the there is rolling motion (also shown in Figure 5.2). This rotation while in contact we have named the mirror roll and its utility in assembly depends on the design specification of a circular perimeter in FIMER, to make a pair of them essentially a kinematic mechanism similar to a toothless gear. This action connects the two beams together. After the mirror roll is completed the robots repeat the previous behavior to connect the 3rd beam to connect to the existing structure to make the triangle (Figure 5.3 and 5.4).

Fig. 5.1.

Fig. 5.2

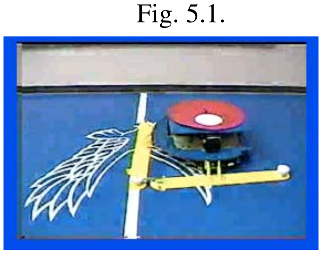

Fig. 5.3.

Fig. 5.4

V. Conclusions

We built an experimental system that can be used to demonstrate in-Space assembly of basic structures. Our current experiments operate in two dimensions on an air table and use centralized control.

Accurately simulating space was a difficult task because of the noise and differential pressures generated in our test bed. In our system we tried to mitigate this noise by making sure that the gain and damping of the system were large enough to overcome the noise.

The airbed inaccuracies forced the use of continuous mode control in our experiments rather than the impulse control normally used in space. This affected the verisimilitude of our work but not the end results.

Our docking system was well developed for 3-D in a prior project and performed well here in 2-D.

As usual in imaging applications, there were lighting problems. Our biggest problem was with the colors on the targets. Slight change in lighting conditions or replacing the targets with new ones caused a change in detection accuracy. A better, intuitively more appealing, scheme would have been to mount cameras on top of the robots and make them point towards the roof to image predetermined markers to emulate star tracking systems used in real spacecraft for attitude determination and control. The advantage of this would be that sensing would be autonomous and distributed. On the down side, an upward pointing camera would have consumed higher initial resources than using the roof mounted one. A less tight schedule or re-examination of the issues might lead a future researcher to the upward pointing camera.

The work has shown that automated assembly in space, albeit with much development, is a practical proposition.

References

[1] Castano, A., W.-M. Shen and P. Will, ``CONRO: Towards Deployable Robots with Inter-Robot Metamorphic Capabilities. *Autonomous Robots Journal*, 8(3), 309-324, 2000.

[2] Castano, A., A. Behar and P. Will, The CONRO modules for self-reconfigurable robots, *IEEE Transactions on Mechatronics*, 7(4), 403-409, Dec. 2002.

[3] Shen, W.-M., P. Will, B. Khoshnevis, Self-Assembly in Space via Self-Reconfigurable Robots, *International Conference on Robotics and Automation*, Taiwan, 2003.

[4] Shen, W.-M. and P. Will, Docking in Self-Reconfigurable Robots, *International Conference on Intelligent and Robotic Systems*, *2001*.

[5] B. Khoshinevis, R. Kovac, W.-M. Shen, and P. Will, Reconnectable Joints for Self-Reconfigurable Robots, *International Conference on Intelligent and Robotic Systems*, *2001*.

[6] Everist, J., Mogharei, K., Suri, H., Ranasinghe, N., Khoshnevis, B., Will, P., Shen, W.-M., A System for In-Space Assembly, *International Conference on Intelligent and Robotic Systems*, Sendai, Japan, 2004.

[7] Basic Stamp: www.**parallax**.com

[8] Timothy Chung, Lars Cremean, William B. Dunbar, Zhipu Jin, Eric Klavins, David Moore, Abhishek Tiwari, Dave van Gogh, Stephen Waydo. A Platform for Cooperative and Coordinated Control of Multiple Vehicles: The Caltech Multi-Vehicle Wireless Testbed. *Conference on Cooperative Control and Optimization*, 2002.

[9] Boeing: see website: http://www.boeing.com/defensespace/space/bss/factsheets/isf/isf.html

[10]Cosmo, ML, and Lorenzini, E.C. Tethers In Space Handbook - Third Edition, prepared for NASA/MSFC by Smithsonian Astrophysical Observatory, <http://www.tethers.com/papers/TethersInSpace.pdf>, Cambridge, MA, Dec 1997.