A study of cooperative control of self-assembling robots in space with experimental validation

2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

A Study of Cooperative Control of Self-Assembling Robots in Space with Experimental Validation

Chiara Toglia, Daniel Kettler, Fred Kennedy, and Steven Dubowsky, *Fellow, IEEE*

Abstract—**Modular self-assembling on-orbit robotic and satellite systems can be more reliable, have lower launch costs, and be more easily repaired and refueled. However, when individual modules assemble, many challenges and opportunities make the control of the assembled system complex. These issues include changes in inertial properties, and redundancy of actuators and sensors. Optimal control methods may be used to coordinate the control of the modules after assembly, insure good performance, and best utilize the combined resources of the assembly of modules. Simulation and experimental results compare this Cooperative algorithm's performance to that of an approach in which the control of the individual modules is not coordinated. Cooperative optimal control methods prove well-suited for controlling redundant, modular space systems.**

I. Introduction

A modular approach to self-assembling on-orbit robotic systems and spacecraft has great potential for reducing costs, increasing long-term reliability, and providing the means for rapid repair and refueling. Such a spacecraft or space robot would consist of an assembly of a number of modules, each designed for specific tasks such as propulsion, payload, fuel storage, etc. See Fig. 1. The modules within a system could share resources such as power, sensors, computational capabilities, and data. The modules' small size and the maintenance of an on-orbit module inventory enables launching flexibility. Moreover, the possibility of mass producing simple modules rather than individually crafting unique satellites offers great potential for design and production savings [1]. The modular approach also provides redundancy. The ability to completely replace failed modules will greatly increase robustness to failure [2], [3].

In this concept, system assembly takes place in orbit. Hence, each module is required to carry sufficient attitude control actuators (thrusters and reaction wheels) and sensors to permit free-flying control and docking. These multiple sensors give the assembled system substantial sensor redundancy that can be used with sensor fusion techniques to minimize sensor errors. The actuator redundancy can provide the systems with greater agility and flexibility in fuel usage management. Moreover it enables the introduction of additional control constraints, for example those related to plume impingement.

Consequently, modular orbiting satellites and robotics systems have the potential to be simultaneously less expensive and more responsive, adaptable, and robust to the failure of one of its module. With these benefits come significant, and largely unaddressed, control challenges. These challenges result from the dynamic interactions between modules, the changing inertial properties, structural compliance, and sensor and actuator redundancy.

The control of formation flying orbital systems has been well studied, as well as the control of spacecraft and space robots maneuvering in close proximity for rendezvous and docking procedures. Orbital formation flying concepts distribute the functionality of large spacecraft over a set of cooperative, smaller, less expensive spacecraft which do not physically contact each other. Work in this field has focused on modeling, coordination and control, simulation, and autonomous formation reconfiguration [4], [5], [6]. However, the elements in these systems do not face the challenges of physical interaction found in the modular space robot concept addressed here.

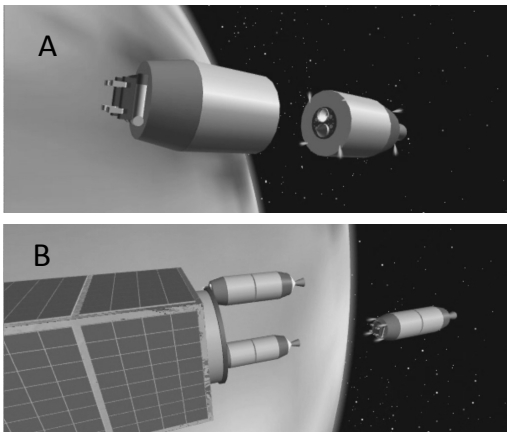

Fig. 1. Image A - two modules assembling. Image B - assemblies capturing a satellite

Substantial work has also been done on the dynamics and control of the rendezvous and docking of spacecraft and space robots [7], [8], [9], [10]. These works have focused on the period just before docking when the docking elements are free-flying or free-floating, or the first few moments after docking. The assumption generally made in these works is that, after docking, the control of the system is stable and well behaved. Moreover, the control of the system after docking is not adjusted or optimized to account for changed

C. Toglia is a PhD Candidate at "La Sapienza", Universita' di Roma, Rome, Italy (e-mail: chiara.toglia@uniroma1.it).

D. Kettler is a Masters student at MIT, Cambridge, MA 02139 USA (e-mail: kettler@mit.edu).

F. Kennedy is a Lt Col in the US Air Force and has been attached to the Defense Advanced Research Projects Agency.

S. Dubowsky is a Professor in the Mechanical Engineering Department, MIT, Cambridge, MA 02139 USA (e-mail: dubowsky@mit.edu).

mass properties of the joined system. These problems must be investigated to properly control self-assembling modular space robots and spacecraft. One of the few studies that have considered the control of assemblies uses the SPHERES platform [11].

In this paper, the analytical development of a Cooperative Control approach, in which control efforts are coordinated between the modules using linear quadratic regulator (LQR) methods, is presented. This approach uses optimal control methods to coordinate the control of the modules after assembly to insure good stable system performance, and to best utilize the combined resources. The algorithm balances trajectory error, plume impingement, total fuel consumption, as well as the distribution of fuel consumption among modules in determining actuator commands and thruster selection. It is important that the system balances the fuel usage between modules since fuel is difficult to redistribute. Simulation and experimental results are presented for the algorithm's performance and compared to those of a control approach in which the control of the individual modules is not coordinated after assembly, called here Independent Control. The simulations are done for a representative system consisting of an assembly of nanosatellite modules. The experimental results are obtained using the MIT Field and Space Robotics Laboratory (FSRL) Free-Flying Space Robot (FFR) Testbed [12], [13]. Both the simulations and experimental results show the effectiveness of the proposed control approach. The Cooperative Control approach performs better than Independent Control, yielding lower trajectory errors, and lower fuel consumption.

II. System Model and Assumptions

A number of simplifying assumptions are made in modeling the system. For clarity, a 2D planar case is considered, and in both simulations and experiments, trajectories consist of sequential, not simultaneous, translation and rotation elements. More complex trajectories have been tested in simulation but are beyond the scope of this paper [14]. The small effects of solar pressure, gravity gradient, and thermal warping are neglected. The orbit altitude is assumed sufficiently high so that aerodynamic effects are also negligible. Further, it is assumed that the time scale for an assembly's operations is much shorter than the orbital period, so that the effects of orbital mechanics are not included. The compliance of the system elements is also neglected. Finally, the modules are assumed to have thrusters but not reaction wheels. As a result, the dynamics of the assembly, or an individual module, may be approximated by the simple linear equation

$$\dot{x} = Ax + Bu \tag{1}$$

where x is the $n \times 1$ state vector and u is the $r \times 1$, where $r = p \times m$, p is the number of thrusters per module and m is the number of modules composing the assembly. Since there is no damping, the A matrix contains simple integrator dynamics, *i.e*. zeros and ones. All the information related to the assembly configuration is contained in the B matrix: inertial characteristics, number of thrusters, direction of each thrust, and the relative distance of the thrusters from the global center of mass, *i.e.* geometry of the thruster placement. Using such a compartmentalized approach, the dynamics and the control strategies can be easily adapted to any configuration, by updating the B matrix. Simulations demonstrating this adaptability to multiple modules have been done and will be covered in future publications.

A. The Control Problem

For a fixed system configuration, a stable and effective attitude and maneuvering control system can be designed using well known methods. However, when the individual modules assemble themselves, a number of factors enter, making this problem more complex. These include changes in inertial properties and redundancy of actuators and sensors. Each module could control itself as if it were independent, so that control would be distributed and not cooperative. However, this control algorithm would clearly be suboptimal and would potentially have robustness issues. For example, measurement errors and noise, as well as uncertainty in actuator thrusts could produce control errors that would cause the individual modules to "fight" against each other. This results in increased fuel consumption, higher trajectory errors, and higher internal constraint forces in the docking mechanism between modules. In extreme cases the result could be destabilization of the system.

B. The Control Concept and Performance metrics

A more effective way to control the assembled modules, and minimize the above problems, is Cooperative Control, *i.e.* a single well integrated architecture that reflects the current configuration of modules. Developed below, a Cooperative Controller provides good performance and stability while exploiting sensor and actuator redundancy.

Several metrics are used to develop and evaluate this control approach. The first is the trajectory error on a selected reference maneuver. The second is the controller's fuel consumption. The total amount of fuel consumed and the control algorithm's ability to direct fuel usage among modules are considered. Finally, the magnitude of forces between connected modules indicates the degree to which the thrust commands of modules conflict. Coordination of module control will minimize these forces.

C. Proposed Optimal Cooperative Control Algorithm

Since the system is time-varying linear, a linear quadratic regulator (LQR) optimal controller is used to minimize errors during a maneuver while minimizing total fuel usage and balancing the fuel usage among modules. The cost function J to be minimized is:

$$J = \delta x^T(t_f) H \delta x(t_f) + \int_{t_0}^{t_f} \left(\delta x^T(t) Q \delta x(t) + u^T R u \right) dt \tag{2}$$

where $\delta x = x_{des} - x$ is the trajectory error, $\delta x^T(t_f)\, H\, \delta x(t_f)$ is the cost at the terminal time t_f, Q, and R are square matrices. The first term in the integral penalizes errors in following the

trajectory command, while $u^T Ru$ is the cost on the fuel consumed by the thrusters. The control strategy has to be computed ahead of time. The optimal solution is [15]:

$$u(t) = -R^{-1}B^T\left[W(t_r)x(t) + \frac{1}{2}V(t_r)\right] \quad (3)$$

where $t_r = t_f - t$ is the remaining time. The matrix $W(t_r)$ is obtained integrating the Riccati equation in (4) .

$$\frac{dW}{dt_r} = W(t_r)A + A^T W(t_r) - W(t_r)BR^{-1}B^T W(t_r) + Q \quad (4)$$

The matrix $V(t_r)$ can be found by integrating equation (5).

$$\frac{dV}{dt_r} = A^T V(t_r) - W(t_r)BR^{-1}B^T V(t_r) - 2Qx_{des} \quad (5)$$

A closed loop control can thus be obtained from (3), using the time-varying gains $W(t_r)$ and $V(t_r)$ that are computed *a priori*. This controller automatically selects thrusters form redundant sets to minimize fuel consumption.

In order to address such issues as plume impingement and the balancing of fuel consumption, the proposed algorithm adjusts the B matrix. B is a $n \times r$ matrix, that can be decomposed as follows

$$B = \overline{B} \cdot B_{PIC} \cdot B_{FB} \quad (6)$$

$\overline{B}$ is a $n \times r$ matrix which translates thruster inputs into net forces and torques about the principle axes. It describes the behavior of the system including all thrusters without any special adjustments. B_{PIC} is a $r \times r$ matrix that introduces the Plume Impingement Constraint (PIC), by removing the contribution of those thrusters that are poorly positioned. Thus B_{PIC} acts as a selection matrix: all the non diagonal elements are zero. The diagonal elements related to unusable thrusters are zero, while those diagonal elements related to well positioned thrusters are one. Note that the methods developed for PIC may also be used to compensate for malfunctioning thrusters. B_{FB} is a $r \times r$ matrix that introduces a weighting based on a Fuel Balancing (FB) constraint. This insures balanced fuel consumption among the included modules so that resource levels stay evenly apportioned. It is a diagonal matrix whose elements are weights based on relative fuel levels related to each module. The weighting is the same for all the thrusters belonging to the same module and has to be computed at every instant, taking into account possible leaks of fuel or differences between actual and commanded thrusts. A different approach to fuel balancing can be followed if the difference in fuel levels is known *a priori*: the fuel usage distribution can be regulated using different weights in the LQR gains computation: in particular the R matrix, which is a diagonal matrix, can be written using weighted gains for each robot, so the fuel distribution is optimal and computed ahead of time.

Cooperative Control enables the easy incorporation of additional modules and constraints in an optimal fashion through the B matrix. An alternative approach is developed in previous literature [11]. It searches through the thrusters to assign one thruster to supply each component of force or torque. The forces and torques to be applied at the center of mass are computed and the components commanded from the specified thrusters. This approach is simple, reliable, and does not require an expensive gain calculation, with a consequent saving in computational cost. However, it is not optimal and does not enable the easy addition of constraints such as Fuel Balancing. Consequently, it is less general than the Cooperative LQR approach and does not provide means of minimizing fuel consumption.

III. Simulations

A. Case Study: Description and assumption

To facilitate the design of controllers, the case of two nanosatellite modules is considered. See Fig. 2. Each module is equipped with two manipulators and eight thrusters, resulting in a total of 16 thrusters for the assembly. Commands are continuous. The modules have a mass of 10 kg, dimensions on the order of 0.5m, and a maximum manipulator reach of 20 cm An assembly composed of two nanosatellites with identical characteristics (mass, moment of inertia, thrusters location and saturation threshold) is considered. The assembly configuration is symmetric, although the method extends to asymmetric assemblies and non-identical modules. Manipulators connect the modules. These manipulators are locked and are assumed rigid. For simplicity, modules are equipped only with thrusters in these simulations. The reference trajectory is planar and composed of two sequential parts: a translation in the y direction, starting and ending with zero velocity, and a rotational motion around the axis normal to the plane, Fig. 2. The mission duration is 30 s, each parts lasts 15 s.

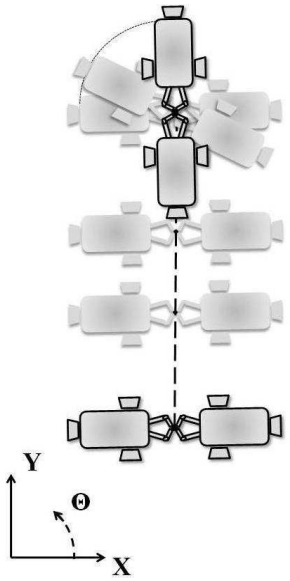

Fig. 2. Reference Trajectory.

B. Simulation Results

The Cooperative Controller and the Independent Controller were investigated and compared. Both the control strategies had good trajectory tracking performance, with the Cooperative Controller producing slightly better results. Fig. 3 shows a comparison of the tracking performance. The Independent Controller produces small oscillations when the system is commanded to keep constant attitude angle, because of the antagonism of the two independent controllers. Similarly, a small drift along X and oscillations about the reference trajectory in the Y direction can be seen in Fig. 3. Fig. 4 shows that the total fuel consumption for the

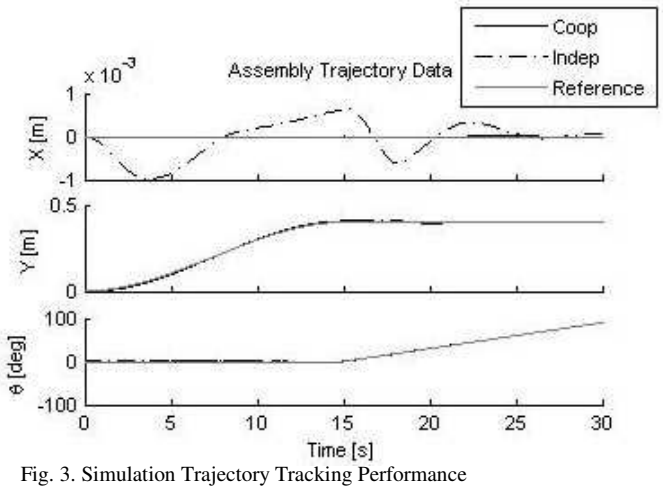

Fig. 3. Simulation Trajectory Tracking Performance

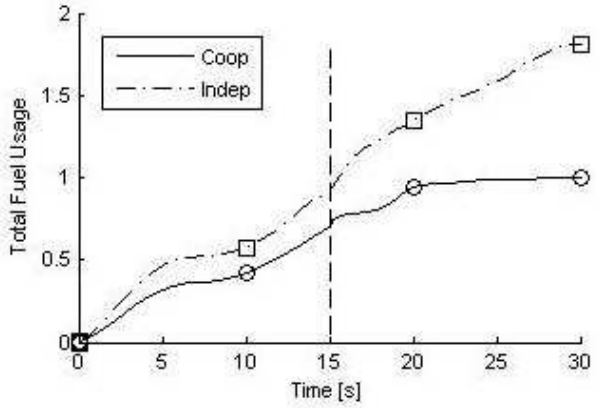

Fig. 4. Simulation total fuel consumption. The vertical line indicates the switch from translation to rotation

two controllers is very different: the Independent control used 82% more fuel than Cooperative Controller. In the figure, the fuel consumption has been normalized with respect to the total fuel used under the Cooperative Control.

Adding the Plume Impingement Constraint (PIC) to the Cooperative Controller did not change performance significantly. Fuel consumption was roughly the same.

Applying the fuel balancing constraint to the controller produced the desired inequality in fuel consumption rates as well as in an increase of the total fuel usage of approximately 6%. This result is not unexpected, since Fuel Balancing constraints can alter the penalty on overall fuel consumption.

IV. Experimental Validation

To validate the simulation results, an experimental study of the Cooperative Controller was performed, using the MIT Field and Space Robotics Laboratory (FSRL) Free-Flying Space Robotics (FFR) Testbed [12], [13]. See Fig. 5.

The FFR Testbed consists of two multi-arm, 6.5 kg robots floating on CO_2 bearings to emulate microgravity in two dimensions. For these tests the robots floated on a 1.3 m x 2.2 m polished granite table. The robots are equipped with two Scara-type two-joint manipulators, eight thrusters, two module position sensors that provide position and velocity data, four manipulator joint angle encoders, and two force/torque sensors. The robots have 7 DOF in total (2 module DOF's in translation and 1 DOF in rotation, and 4 DOF's for the manipulators' joint motions), all of which are controllable and observable. Throttled thrust commands are achieved by commanding individual thrusters with a pulse width modulated signal. All actuators and sensors are controlled by an on-board computer and powered by onboard batteries, so that the robots can work without any externally connected cables. Under Cooperative Control, both modules in an assembly are controlled by the processor on one module. While not currently used, the additional processor of the remaining module is available for computationally expensive tasks such as gain recalculation. The experimental operator can access the onboard computer using wireless LAN adapters. The maximum thruster force is approximately 0.1 N.

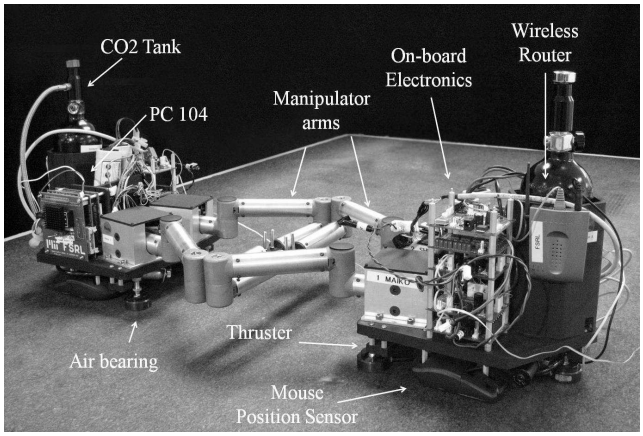

Fig. 5. FFR Testbed showing two robot modules in an assembly.

For the experiments discussed here, the robots' manipulator end-effectors were magnetically connected during the tests. The manipulators attempted to maintain constant module internal geometry during the tests. However, compliance in the manipulators and in the connections introduced some flexibility.

A. Cases studied

Two controllers were investigated: Cooperative Control, and Independent Control. A number of variations on Cooperative Control, including Plume Impingement Constraint (PIC) and Fuel Balancing (FB) were tested.

For the majority of tests, the reference trajectory is a short translational maneuver of a distance of 0.75 m using a constant-velocity step in the Y direction, starting from rest. When testing the Cooperative Control with, and without Plume Impingement Constraints, a 90° rotation trajectory was used, as the simple translational maneuver created minimum plume impingement potential. In this case, the robot assembly started at rest and was commanded to follow a step command in rotational velocity.

B. Performance metrics

The performance of the investigated control methods were compared using the following metrics.

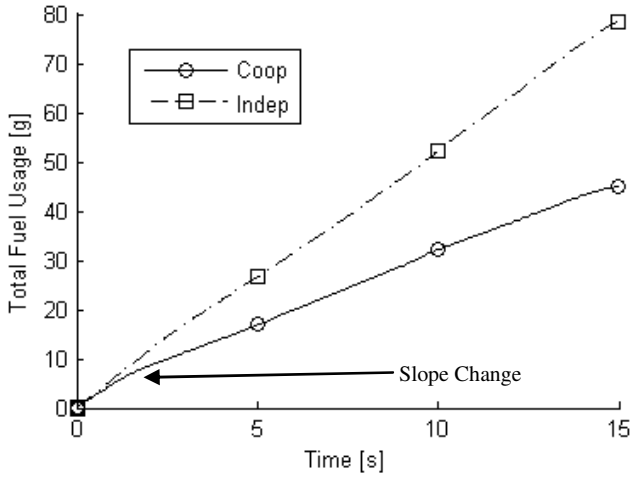

Fig. 6. Median fuel consumption for Cooperative and Independent Control. Note the distinctive reduction in slope of the Cooperative line at time 1.5 s.

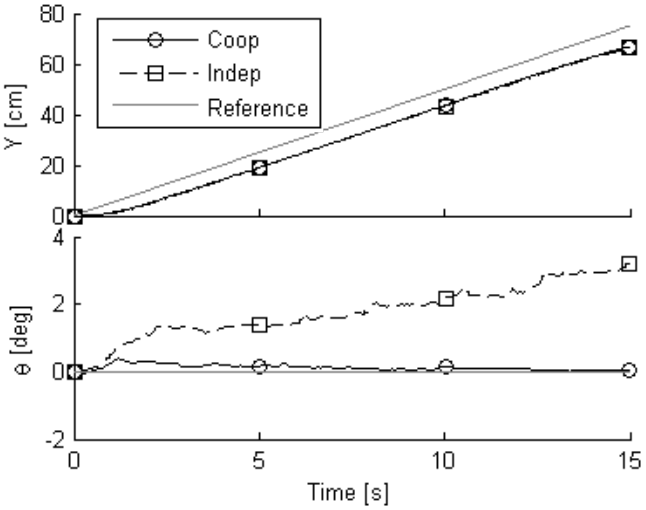

Fig. 7. Median Y and Θ RMS error for Cooperative and Independent Control. Y performance is equivalent. Cooperative Control has much lower Θ error.

Fuel consumption: The total amount of the fuel (CO_2 gas) consumed by the individual robots during the test was estimated from the thruster command history. These values do not include the CO_2 gas used to float the robots.

Trajectory Error: The root mean square error (RMS error or RMSE) that each controller achieves on the commanded trajectory was also considered. Position and orientation errors were collected in addition to their derivatives. The velocity error metrics were calculated using data from only the final ⅔ of the duration of a test in order to allow initial transients to decay.

To evaluate each controller, a total of 10 trials were run using the appropriate trajectory. The means of collected fuel usage and trajectory RMS error values were compared for statistically significant differences using *t*-tests. In all cases, statistical significance corresponds to $p < 0.05$, indicating that there is a 5% probability that the observed results could occur when there is no difference in the means.

C. Results

Table I lists mean fuel consumption and trajectory RMS error for experimental results on the Cooperative and Independent Controllers. These results show that the Cooperative approach is superior. The Cooperative Control used 43% less fuel (45 g) than the Independent Control (79 g). See Fig. 6. The change in slope indicates where the Cooperative Controller reduces fuel consumption after initial acceleration. Much of this savings resulted because under Cooperative Control antagonism between the module propulsion systems is avoided, allowing the assembly to enter a coasting mode and fire thrusters only occasionally to maintain trajectory. Under Independent Control, small miss-alignments and compliance in the connecting links foster small oscillating fluctuations in assembly geometry and wasted thrust. The Cooperative Control also had better trajectory tracking performance (see Fig. 7) with significant improvements in X RMS error and, most notably, Θ RMS error where the Cooperative LQR controller reduced mean error by 91%.

Table II lists mean fuel consumption and trajectory RMS error for experimental results on the Cooperative and Cooperative with Plume Impingement Constraint (PIC) Controllers. While the additional constraint effectively prevented the use of poorly-positioned thrusters as desired, there were negligible reductions in trajectory tracking performance. A significant difference was only found between the mean Θ RMS errors. However, this difference was only 3%.

Unexpectedly, the addition of the plume impingement constraint reduced total fuel consumption by 18%. This occurs because with PIC the assembly no longer directs thruster plumes against its own surfaces thereby wasting thrust and fuel.

Table III lists mean fuel consumption and trajectory RMS error for experimental results on the Cooperative and Cooperative with Fuel Balancing (FB) Controllers. Experimental results demonstrate, with statistical

TABLE I
COOPERATIVE CONTROL VS INDEPENDENT CONTROL

	COOPERATIVE	INDEPENDENT	*P* VALUE
Total Fuel Consumption	45 g	79 g	<0.001
X RMSE	0.9 cm	1.2 cm	<0.001
Y RMSE	6.2 cm	6.3 cm	=0.456
Y' RMSE	2.4 cm s^{-1}	2.4 cm s^{-1}	=1.000
Θ RMSE	0.164°	1.900°	<0.001

Note $p < 0.05$ corresponds to statistical significance.

TABLE II
COOPERATIVE CONTROL VS COOPERATIVE PIC CONTROL

	COOPERATIVE	COOPERATIVE PIC	*P* VALUE
Total Fuel Consumption	147 g	120 g	<0.001
X RMSE	0.7 cm	0.7 cm	=0.255
Y RMSE	0.2 cm	0.3 cm	=0.253
Θ RMSE	2.86°	2.95°	=0.387
Θ' RMSE	0.05 °s^{-1}	0.05 °s^{-1}	=0.003

Note $p < 0.05$ corresponds to statistical significance.

TABLE III
COOPERATIVE CONTROL VS COOPERATIVE FB CONTROL

	COOPERATIVE	COOPERATIVE FB	*P* VALUE
Robot 1 Fuel Consumption	22 g	20 g	<0.001
Robot 2 Fuel Consumption	22 g	23 g	=0.068
Consumption Ratio	1.00	1.18	<0.001
X RMSE	0.9 cm	1.0 cm	=0.033
Y RMSE	6.2 cm	6.3 cm	=0.779
Y' RMSE	2.4 cm s^{-1}	2.5 cm s^{-1}	=0.031
Θ RMSE	0.16°	0.19°	=0.640

Note $p < 0.05$ corresponds to statistical significance.

significance, that Fuel Balancing affected fuel consumption patterns shifting the consumption ratio between the two modules from 1.00 with the Cooperative Controller to 1.18 with the Cooperative FB Controller. The higher penalties on control introduced by Fuel Balancing also resulted in a 2% decrease in total fuel consumption. Trajectory tracking performance remained nearly unchanged. There were only slight but statistically significant increases in X and Y' RMS error.

V. CONCLUSION

This work demonstrates that the proposed Cooperative Control methods are an effective means of planning and implementing control strategies for modular assemblies of spacecraft and space robots. A Cooperative Controller inherently reduces conflicting thrust commands from the different modules. An LQR approach naturally determines optimal commands for any given thruster configuration, including those with thruster redundancy or asymmetry. Consequently, as assembly and thruster geometries change, the Cooperative Controller may be updated without offline, human intervention by simply updating the system model. This approach is clean to implement and optimal, insuring low fuel consumption. It can also easily include Plume Impingement Constraints, and Fuel Balancing. Cooperative Control is a unified, methodical, and general approach to the control of assemblies of spacecraft and space robots.

The results of this work have identified a number of topics for future investigation. The comparison of Independent and Cooperative Controllers suggested that compliance and vibration had a large effect on performance. Methods of minimizing assembly oscillations through control could increase performance. Results showed that the burden of resource consumption could be distributed through Fuel Balancing. The best way to chose Fuel Balancing weights needs to be determined. In order to make more self-sufficient assembly controllers, the development of online system identification methods is essential. Identification could be used to determine the properties of a new assembly, or respond to failures such as thruster malfunctions. Finally, the exploitation of redundant sensors through sensor fusion should also greatly improve performance.

ACKNOWLEDGMENT

Past and present members of the MIT Field and Space Robotics Laboratory have made substantial contributions to this research. In particular we would like to thank Patrick Barragan, Peggy Boning, Dan Lopuch, Dr. Jamie Nichol, Tatsuro Nohara, and Masahiro Ono.

REFERENCES

[1] J. Pizzicaroli, "Launching and Building the Iridium® Constellation", *IAF Workshop on Mission Design and Implementation of Satellite Constellations*, Toulouse, France, 1997.
[2] K. Yoshida, "ETS-VII Flight Experiments For Space Robot Dynamics and Control - Theories on laboratory test beds ten years ago, Now in orbit", in D. Rus and S. Singh (Eds.): *Experimental Robotics VII*, LNCIS, Springer-Verlag Berlin Heidelberg, 2001, pp. 209–218.
[3] G. V. Hughes, "The Orbital Express Project of Bristol Aerospace and MicroSat Launch Systems", Inc., Published by AIAA, Paperback, 1997.
[4] M. Tillerson, L., Breger, L., and J. P. How, "Distributed Coordination and Control of Formation Flying Spacecraft", *Proceedings of the IEEE American Control Conference*, June 2003.
[5] G. Inalhan, F. D., Busse, and J. P. How, "Precise formation flying control of multiple spacecraft using carrier-phase differential GPS", *Proc. Guidance, Control and Navigation Conference*, 2000
[6] S. R. Vadali, S. S., Vaddi, and K. T. Alfriend, "An intelligent control concept for formation flying satellite constellations", *Int. J. Robust Nonlinear Control*, 12 (2–3), 2002, pp 97–l15.
[7] S. McCamish, M. Romano, and X. Yun, "Autonomous Distributed Control Algorithm for Multiple Spacecraft in Close Proximity Operations", *AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, South Carolina, Aug. 20-23, 2007
[8] W. Fehse, *Automated Rendezvous and Docking of Spacecraft*, Number 16 in Cambridge Aerospace Series, Cambridge University Press, Cambridge, UK, 2003.
[9] Yoshida, K., Mavroidis, C., and Dubowsky, S. "Dynamics and Control of a Space Robot Capturing a Floating Target (Experiments of Impact Estimation)." *Proceedings of the 1995 ISAS 5th Workshop on Astrodynamics and Flight Mechanics*, pp. 108-114, August 1995.
[10] G. Hirzinger, K. Landzettel,B. Brunner, M. Fischer, C. Preusche, D. Reintsema, A. Albu-Schäffer, G. Schreiber, B. M. Steinmetz, " DLR's robotics technologies for on-orbit servicing", *RSJ The International Journal of the Robotics Society of Japan, Advanced Robotics*, Special Issue Service Robots in Space, Vol. 18, No. 2, 2004, pp. 139-174, VSP, Netherlands
[11] S. Mohan and D. Miller, "SPHERES Reconfigurable Control Allocation for Autonomous Assembly", *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, 18-21 Aug 2008.
[12] P. Boning, M. Ono, T. Nohara, S. Dubowsky, "An Experimental Study of the Control of Space Robot Teams Assembling Large Flexible Space Structures", *Proc. of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS,* 2008, Los Angeles, CA, USA
[13] M. Ono, P. Boning, T. Nohara, S. Dubowsky, Experimental Validation of a Fuel-Efficient Robotic Maneuver Control Algorithm for Very Large Flexible Space Structures, *Proc. of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*, May 19-23, 2008, Pasadena.
[14] C. Toglia, Phd. Thesis. Expected: Nov 2009.
[15] M. Sidi, *Spacecraft Dynamics and Control,* Cambridge, United Kingdom: Cambridge University Press, 1997.