Auton Robot
DOI 10.1007/s10514-011-9238-z

Cooperative control of modular space robots

Chiara Toglia · Fred Kennedy · Steven Dubowsky

Received: 14 December 2009 / Accepted: 9 June 2011
© Springer Science+Business Media, LLC 2011

Abstract Modular self-assembling on-orbit robots have the potential to reduce mission costs, increase reliability, and permit on-orbit repair and refueling. Modules with a variety of specialized capabilities would self-assemble from orbiting inventories. The assembled modules would then share resources such as power and sensors. As each free-flying module carries its own attitude control actuators, the assembled system has substantial sensor and actuator redundancy. Sensor redundancy enables sensor fusion that reduces measurement error. Actuator redundancy gives a system greater flexibility in managing its fuel usage. In this paper, the control of self-assembling space robots is explored in simulations and experiments. Control and sensor algorithms are presented that exploit the sensor and actuator redundancy. The algorithms address the control challenges introduced by the dynamic interactions between modules, the distribution of fuel resources among modules, and plume impingement.

Keywords Space robots · Cooperative control · Modularity

C. Toglia (✉)
Dipartimento di Meccanica e Aeronautica, Università di Roma "La Sapienza", Via Eudossiana 18, 00184 Roma, Italy
e-mail: chiara.toglia@uniroma1.it

F. Kennedy
Joint Chiefs of Staff/J-8, US Air Force, 8000 Joint Staff Pentagon, Washington, DC 20318-8000, USA
e-mail: fred.kennedy@js.pentagon.mil

S. Dubowsky
Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
e-mail: dubowsky@mit.edu

Published online: 30 June 2011

1 Introduction

Modular self-assembling on-orbit robots and spacecraft can reduce costs, increase reliability, and permit the rapid repair and refueling of on-orbit systems. Such a space robot would consist of an assembly of self-sufficient modules, each with a specific function. Assembled modules would share resources such as power, sensors, computational capabilities, and data. A system composed of mass-produced modules would be less costly than a custom designed one (Pizzicaroli 1997). Because of each module's small size and low cost, modules could be launched using inexpensive boosters. An inventory of modules can be "parked" in orbit (Sweetman 2008), increasing launch schedule flexibility and permitting rapid operations for such missions as disasters response (e.g. tsunamis, earthquakes). The orbiting modules could be used to replace failed modules, extending a system life (Yoshida 2001; Hughes 1997). Figure 1 shows a concept of how three two-module assemblies could provide propulsion for a valuable disabled satellite.

For Modular Space Robots (MSRs), the modules self-assemble while in orbit to create larger satellites tailored to specific missions. Since each module carries sufficient attitude control actuators and sensors to permit free-flying control and docking, the assembled system has substantial sensor and actuator redundancy. Sensor fusion techniques can be used to minimize individual sensor errors. The actuator redundancy can give a system greater agility and flexibility in managing fuel usage. Moreover, it enables the introduction of additional control constraints. For example, in an assembly, some thrusters may be poorly positioned so that their thruster plumes hit other parts of the assembly, dissipating and misdirecting thrust as well as potentially damaging modules. Thruster redundancy allows the addition of plume impingement constraints to prevent the use of such thrusters.

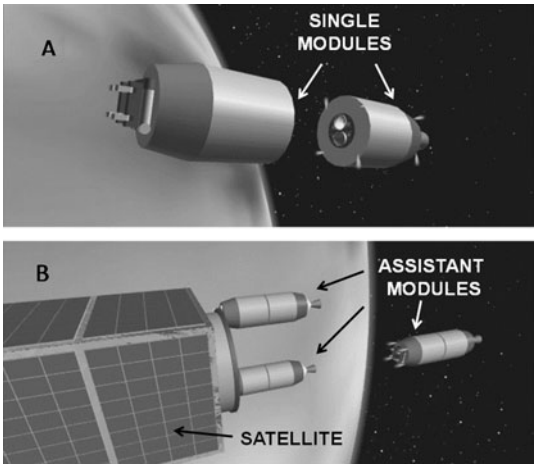

Fig. 1 Space robots assembling to recover a disabled communications satellite

The concept of reconfigurable space systems using simple modules has been proposed for some time, and simple systems have been developed and flown (Kennedy 2007; Mohan et al. 2009). These systems have the potential to be less expensive and more responsive, adaptable, and robust to the failure of one of its parts than conventional satellites. However, modular systems present numerous control challenges resulting from the dynamic interactions between modules, changes to system properties with the addition of a module, docking structure compliance, and sensor and actuator redundancy. While these challenges are largely unaddressed, substantial research exists on related problems.

The control of formation flying orbital systems consisting of a number of small satellites as well as the control of spacecraft and space robots maneuvering in close proximity for rendezvous and docking procedures has been well studied. In formation flying systems, a set of small inexpensive spacecraft cooperate to perform a mission without being in physical contact. Substantial progress has been made in the control, coordination and reconfiguration of these systems (Tillerson et al. 2003; Inalhan et al. 2000; Vadali et al. 2002). However, formation flying systems do not have the challenges of physical interaction found in MRS systems.

Substantial work has been done on the control of the rendezvous and docking of spacecraft and space robots (McCamish et al. 2007; Fehse 2003; Yoshida et al. 1995; Hirzinger et al. 2004). These works generally focus on the period just before docking when the spacecrafts are free-flying or the impact during docking. This contrasts with MSR systems where focus rests on the effective control of static configurations of modules after assembly. While recent studies have begun to address some of the issues of simple configuration MSR systems the problem is far from solved (Mohan and Miller 2008; Toglia et al. 2009).

In this paper, an organized sensing and control approach for generating optimal controllers for arbitrary configurations of assembled modular satellites is developed. The docking and assembly process is not considered. The analytical development of a Cooperative Control approach is presented, in which control efforts are coordinated between the modules. This approach uses LQR optimal control to coordinate the modules and ensure good system performance, and to best utilize the integrated system 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 the transfer of fuel between modules is difficult. Cooperative Control is compared to a control approach where the control of individual modules is not coordinated (Independent Control).

These control and redundancy algorithms are studied in simulation and experimentally using the MIT Field and Space Robotics Laboratory Free-Flying Space Robot Test bed (Boning et al. 2008; Ono et al. 2008). Cooperative and Independent Control performance are compared. Methods are also explored for handling plume impingement and the balancing of fuel consumption between modules within the proposed control architecture. Both the simulations and experimental results show the effectiveness of the proposed control approach as well as the performance increase that sensor fusion can provide. The Cooperative Control performs substantially better than Independent Control, yielding lower trajectory errors, and lower fuel consumption. Enhancements in state estimates and lower noise levels provided by sensor fusion further improve trajectory tracking accuracy and fuel consumption.

2 System model and assumptions

In modelling the system, the small effects of solar pressure, gravity gradient, and thermal warping are assumed negligible. The system is assumed to be compact, of microsatellite scale, and to have a Low Earth Orbit altitude of 600 km. Consequently, because of the small dimensions and high altitude of the system, aerodynamic effects are neglected. Circular orbits are also assumed. Assuming that the manoeuvre time is much shorter than the orbital period, effects of orbital mechanics are not considered; however, if trajectories executed over multiple orbits are considered, then the dynamics of the assembly can be described by the linear Hill–Clohessy–Wiltshire equations (Clohessy and Wiltshire 1960), for which the small motions of a MSR assembly is linearized about the orbit. This would not alter the substance of the method as follows. While a discussion of the designs of any proposed modules is beyond the scope of this paper,

preliminary studies suggest that these modules will be small, low mass, and have stiff latching mechanisms (Kennedy 2007). Hence, in an assembly consisting of a handful of modules, the structural resonances are expected to have little impact on the control or sensor fusion algorithms. Consequently, compliance of system elements is neglected. This assumption appears to be borne out by the experimental results shown in Sect. 4 of this paper. However if the modules are very large and the connections are no tight, then additional research is needed to extend the results of this paper to large flexible systems. Finally, the modules are assumed to have only thrusters as actuator, no reaction or momentum wheels nor CMGs that could be used for attitude control. Therefore thrusters are used both for orbital and attitude control. Thruster minimum impulse and saturation are not included in the control design.

For simplicity, the planar motion case is investigated. Results may be generalized to the three dimensional case if small attitude angles are considered. Under the assumptions accepted above, the dynamics of the assembly, or an individual module, may be approximated by:

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

x is the $n \times 1$ state vector, containing position, attitude, velocity and angular velocity coordinates:

$$x = \begin{bmatrix} X & Y & \theta & \dot{X} & \dot{Y} & \dot{\theta} \end{bmatrix}$$

u is the $r \times 1$ control vector of commanded thrusts, defined in each thruster reference frame, where $r = p \times m$, p is the number of thrusters per module and m is the number of modules composing the assembly.

As the system contains no damping, the A matrix contains simple integrator dynamics:

$$A = \begin{bmatrix} [0] & I_{3\times 3} \\ [0] & [0] \end{bmatrix} \tag{2}$$

where $I_{3\times 3}$ is the identity matrix.

All the information related to the assembly configuration is contained in the B matrix: inertial characteristics, number of thrusters, and geometry of the thruster placement. The B matrix is a $n \times r$ matrix which translates thruster inputs into net velocities and accelerations about the principle axes: linear velocities and accelerations in x and y directions and rotational velocity and acceleration about the center of mass. The B matrix takes the form:

$$B = \begin{bmatrix} [0] & [0] & \cdots & [0] \\ b_1 & b_2 & \cdots & b_r \end{bmatrix} \tag{3}$$

Each matrix b_i is a $n/2 \times 1$ matrix. The b_i matrix transforms the input thrust u_i of the ith thruster into the inertial reference frame and thus produces the x and y accelerations.

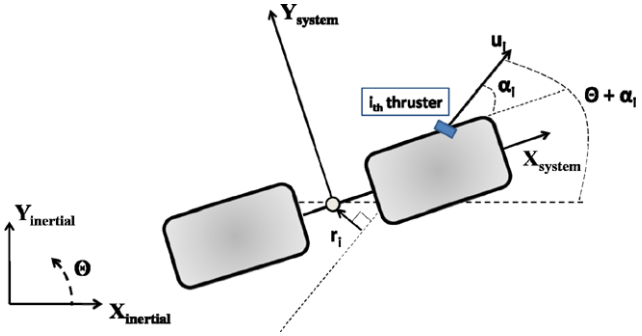

Fig. 2 Geometrical characterization of the ith thruster

It also calculates the rotational acceleration about the center of mass induced by u_i.

$$b_i = -\begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & J \end{bmatrix}^{-1} \begin{bmatrix} \cos(\theta + \alpha_i) \\ \sin(\theta + \alpha_i) \\ r_i \end{bmatrix} \tag{4}$$

The variables M and J are the mass and moment of inertia of the system. The constant α_i is the angular orientation of the actuator referred to the system's reference frame. The vector r_i is the (signed) distance between the ith thruster direction and a line parallel to it and containing the assembly center of mass, i.e. the lever arm for the exerted torque. See Fig. 2. The negative sign on all terms of the b_i matrix accounts for the fact that the thrusters produce forces in the direction opposite their orientation.

The B matrix holds all information defining system characteristics in the model. Consequently, system dynamics and control strategies can be easily adapted to changes in module assembly configuration, by updating the B matrix.

2.1 The control problem

A stable and effective position and attitude controller for a system with a fixed configuration can be designed using well known methods. However, when autonomous modular robots assemble themselves into a larger system, the problem becomes more complex.

In the assembly, each module could continue to control itself as if it were independent, so that control would be distributed and not cooperative. However, this control would be suboptimal. 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 forces in the docking mechanisms between modules. In extreme cases the system could become unstable. A more effective controller for the assembled modules, which minimizes the above problems, is Cooperative Control, i.e. a single integrated architecture that reflects the current configuration of modules.

Two metrics are used to evaluate control approaches. The first is the trajectory error on a selected reference maneuver. The second is the total amount of fuel consumed for control and the control algorithm's ability to balance fuel usage among the module.

Since the dynamics of a module assembly are time-varying and nearly linear (nonlinearity only enters though the B matrix), linear quadratic regulator (LQR) optimal control methods form the basis of both the Cooperative and Independent Controllers. Under Independent Control, each module's controller follows its own trajectory while attempting to minimize its own trajectory errors and fuel usage. Under Cooperative Control, one controller commands the entire assembly and minimizes trajectory errors and total fuel usage as well as balancing the fuel usage among modules. For both control approaches the cost function J to be minimized is:

$$J = \delta x^T(t_f) H \delta x(t_f) + \int_{t_0}^{t_f} \left(\delta x^T(t) Q \delta x(t) + u^T R u\right) dt \quad (5)$$

where $\delta x = x_{des} - x$ is the trajectory error. The term $\delta x^T(t_f) H \delta x(t_f)$ is the cost at the terminal time t_f. The first term in the integral penalizes errors in following the trajectory command, while $u^T R u$ is the cost on the fuel consumed by the thrusters. This cost function must be minimized before system operation to generate control gains tailored to the specified maneuver.

The resulting optimal solution is (Sidi 1997):

$$u(t) = -R^{-1} B^T \left[W(t_r) x(t) + \frac{1}{2} V(t_r) \right] \quad (6)$$

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

$$\frac{dW}{dt_r} = -\big[Q + W(t_r)A + A^T W(t_r) - W(t_r) B R^{-1} B^T W(t_r)\big] \quad (7)$$

The time-dependent matrix $V(t_r)$ can be found by integrating equation:

$$\frac{dV}{dt_r} = A^T V(t_r) - W(t_r) B R^{-1} B^T V(t_r) - 2 Q x_{des} \quad (8)$$

A closed loop control is obtained, using time-varying, pre-computed gains. This controller automatically selects thrusters from redundant sets to minimize fuel consumption at the module (Independent Control) or assembly (Cooperative Control) level.

Under Cooperative Control, to address such issues as plume impingement constraints and the balancing of fuel consumption, the proposed algorithm replaces the B matrix with an adjusted version, $\bar{B}$, for controller development. The $\bar{B}$ matrix is $n \times r$ and can be decomposed as follows:

$$\bar{B} = B \cdot B_{PIC} \cdot B_{FB} \quad (9)$$

B_{PIC} is an $r \times r$ diagonal selection matrix that introduces a Plume Impingement Constraint (PIC). It consists of an identity matrix with diagonal elements corresponding to undesirable thrusters set equal to zero. Exploiting actuator redundancy, B_{PIC} is used to remove poorly positioned thrusters which would experience plume impingement due to assembly geometry from the list of the thrusters the controller may utilize. B_{PIC} may also be used to remove malfunctioning thrusters from service.

B_{FB} is an $r \times r$ diagonal matrix that implements Fuel Balancing (FB). If no adjustments are desired for the relative rates at which different modules consume fuel, B_{FB} is an identity matrix. However, if a certain module is desired to consume fuel at a lower rate than other modules, the diagonal elements of B_{FB} corresponding to this module's thrusters may be set smaller than 1, in order to reduce its control authority. The fuel balancing matrix exploits the modularity of the assembly by assigning different control authority to each module. This allows the distribution of fuel reserves among modules to be evenly consumed. An alternative solution to redistribute fuel consumption among modules is to set the R weighting matrix accordingly, penalizing the use of thrusters on the robot with smaller fuel resources and vice versa. With this approach the R matrix is used to weight the relative control authority of each module.

Cooperative Control enables additional modules and constraints to be easily incorporated through the B matrix (or R matrix). This requires less human intervention than previous methods which search through the thrusters to assign one thruster to supply each component of force or torque (Mohan and Miller 2008). The Plume Impingement Constraint matrix B_{PIC} may be determined automatically from the assembly geometry. Similarly, the Fuel Balancing matrix B_{FB} may be set before a maneuver in response to existing fuel reserves. It is worth noting that after selecting a desired configuration of active thrusters and corresponding B_{PIC} matrix, a simple controllability test will demonstrate whether the desired configuration is viable. If motion in certain directions is not possible and this test is failed, the configuration may be redesigned to correct these problems. The Cooperative Controller then selects the best actuators to use for a given maneuver.

2.2 Sensor fusion

Exploiting the sensor redundancy of MSRs enables a more accurate state estimation and consequently better performance. Given the system's nearly linear structure, an Extended Kalman-Bucy Filter is applied to implement sensor

fusion. It produces a nearly optimal solution and is easy to implement using the same basic structure and coordinate transformations used for the Cooperative Controller.

A previously developed hybrid filter is implemented with sensor updates of state and covariance estimates handled in discrete time, while propagation through time is done using continuous integration (Stengel 1994). The filter estimates the state of the assembly's center of mass. Denavit-Hartenberg transformations are used to transform the sensor data from the coordinate frames of individual modules to the coordinate frame of the assembly (Spong and Vidyasagar 1989).

At each discrete sensor update instant t_k, the filter gain $K(t_k)$ is computed using:

$$K(t_k) = P(t_k^-)C^T\left[CP(t_k^+)C^T + R\right]^{-1} \tag{10}$$

where R is the sensor noise covariance matrix, C is the output matrix $I_{6\times 6}$, and $P(t_k^-)$ is the estimate of covariance at instant t_k before inclusion of current sensor information $z(t_k)$. This gain is used to fuse the new sensor data and update the state and covariance estimates:

$$\begin{aligned}\hat{x}(t_k^+) &= \hat{x}(t_k^-) + K(t_k)\left[z(t_k) - C\hat{x}(t_k^-)\right]\\ P(t_k^+) &= P(t_k^-)\left[I - K(t_k)C\right]\end{aligned} \tag{11}$$

where $\hat{x}(t_k^-)$ and $P(t_k^-)$ are the state and covariance estimates before inclusion of the new sensor information and $\hat{x}(t_k^+)$ and $P(t_k^+)$ after.

Between sensor measurement updates, the state and covariance estimates are propagated through time using numerical integration of the nonlinear dynamics:

$$\begin{aligned}\hat{x}(t_{k+1}^-) &= \hat{x}(t_k^+) + \int_{t_k}^{t_{k+1}} \left(A\hat{x}(\tau) + B(\theta(\tau))u_k\right)d\tau\\ P(t_{k+1}^-) &= P(t_k^+) + \int_{t_k}^{t_{k+1}} (AP(\tau) + P(\tau)A^T\\ &\quad + B(\theta(\tau))Q_d B(\theta(\tau))^T)d\tau\end{aligned} \tag{12}$$

where Q_d is the input disturbance covariance matrix representing disturbances injected into the system by actuator errors. Actuator performance and therefore Q_d are assumed to have no time dependence. For simplicity, disturbance sources besides the actuators are not considered.

3 Simulations

Simulations are used to study Cooperative and Independent Control. Performance is measured in terms of trajectory error and fuel consumption, computed as the mass flow integrated over time necessary to produce the required thrust.

Two different simulation cases were considered. First an assembly of six identical modules is used to compare the performance of Cooperative and Independent Controllers and demonstrate Fuel Balancing. The study of Plume Impingement Constraints is left to experiment because the complex interactions of thruster plumes and assembly structure necessitating PIC are difficult to capture in simulation. The second simulation case represents the rescue of large high-value satellite by two much smaller MSR modules. Here again, Cooperative and Independent Control are compared. In both simulations the same nanosatellite module is used. Each module has a mass of 10 kg and dimensions on the order of 0.5 m. These modules are assumed to be rigidly lock into assemblies using manipulators or docking mechanism. Each module has eight thrusters and its own position and attitude sensors. For all controllers, both Cooperative and Independent, in these simulations, the following LQR weighting matrices are used:

$$H = 0_{n\times n}, \qquad Q = \begin{bmatrix} 200\cdot I_{n/2\times n/2} & 0_{n/2\times n/2} \\ 0_{n/2\times n/2} & 100\cdot I_{n/2\times n/2}\end{bmatrix}$$

$$R = I_{r\times r}$$

3.1 Case 1: symmetric MSR assembly

Initial simulations consider a symmetric assembly of 6 modules arranged radially about a central point. The assembly is commanded to follow a circular arc trajectory while maintaining a constant, radially-aligned orientation relative to the center of the arc. See Fig. 3.

The simulation results comparing Cooperative and Independent controllers may be seen in Table 1. Trajectory tracking performance for both controllers is nearly identical with negligible differences. However, there are substantial differences in fuel consumption of the two controllers, with Independent Control using 12% more.

Figure 4 plots the cumulative fuel consumption of each controller normalized by the total fuel consumption of the Cooperative Controller. Both controllers demonstrate an initial period of high fuel consumption as the assembly accelerates into the commanded trajectory. This is followed by a

Table 1 Cooperative control vs independent control

	Cooperative	Independent
Total Fuel Consumption (Normalized)	1	1.12
X RMSE	0.02 cm	0.02 cm
Y RMSE	0.01 cm	0.01 cm
Θ RMSE	0.03°	0.02°
X′ RMSE	0.001 cm/s	0.001 cm/s
Y′ RMSE	0.015 cm/s	0.012 cm/s
Θ RMSE	0.06°/s	0.02°/s

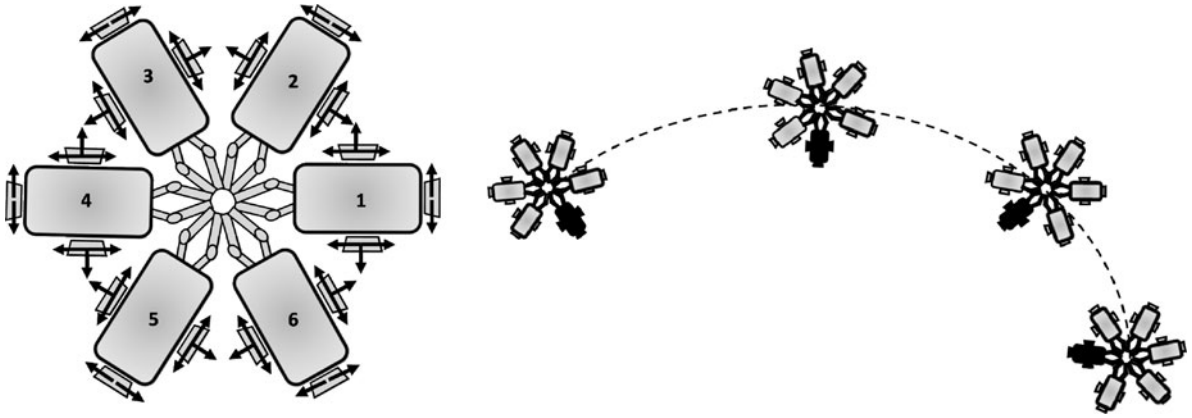

Fig. 3 Six-module assembly configuration and reference trajectory

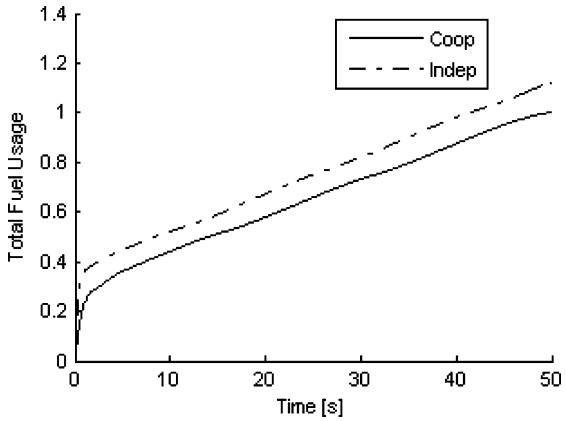

Fig. 4 Comparison of fuel usage by Cooperative and Independent Controllers

long coasting period where fuel is used at a much lower rate to keep the system on the arc trajectory. The difference in total fuel consumption develops during the initial acceleration phase. This occurs because under Independent Control the assembly's thrusters are not used optimally to reduce initial trajectory errors.

Additional simulations were run with Cooperative Control and Fuel Balancing (FB) to adjust the distribution of fuel reserves among the modules by apportioning greater control authority to modules with larger fuel. The modification of the control authority is made through the weighting matrix B_{FB}. Module 3 is assumed to have the largest initial fuel reserve and module 2 the lowest one. Consequently, module 3 is assigned a fuel use weight of 1.1, while module 2 is given the weight 0.8. The choice of the weight depends on the value of the fuel resource with respect to the average among modules.

Recalling that each module is equipped with eight thrusters, the matrix B_{FB} is written as

$$B_{FB} = \mathrm{diag}\left(\begin{bmatrix}\mathbf{I_8} & \mathbf{0.8}\cdot\mathbf{I_8} & \mathbf{1.1}\cdot\mathbf{I_8} & \mathbf{I_8} & \mathbf{I_8} & \mathbf{I_8} & \mathbf{I_8} & \mathbf{I_8}\end{bmatrix}\right)$$

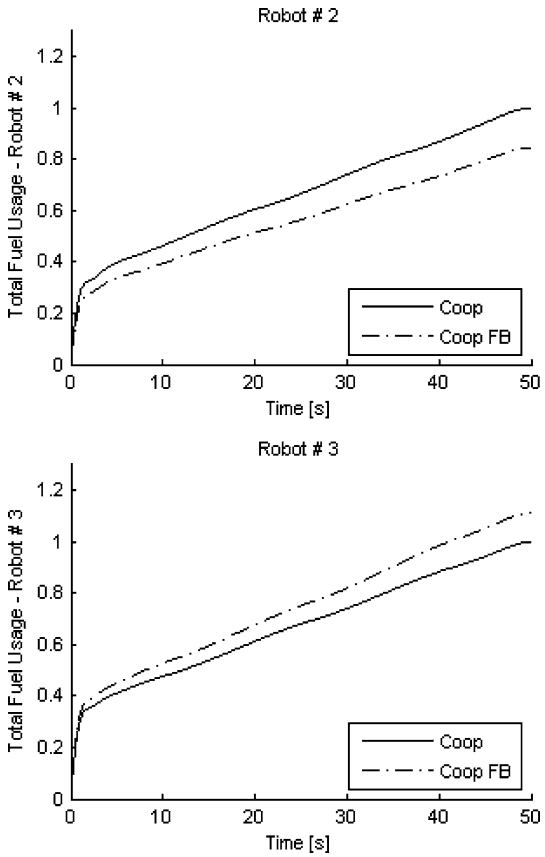

Fig. 5 Module fuel usage—Robot 2 and 3

Figure 5 shows the individual fuel consumption of robots 2 and 3. Fuel usage is normalized with respect to the total fuel used by the same robot under Cooperative Control without FB. With FB, module 2 uses only 85% of the fuel it uses without FB. Conversely, module 3 consumes 111% of its pure Cooperative Controller quantity. The fuel consumption of the other modules is unchanged. The total fuel consumption for the assembly increases by 6% when FB constraint is implemented.

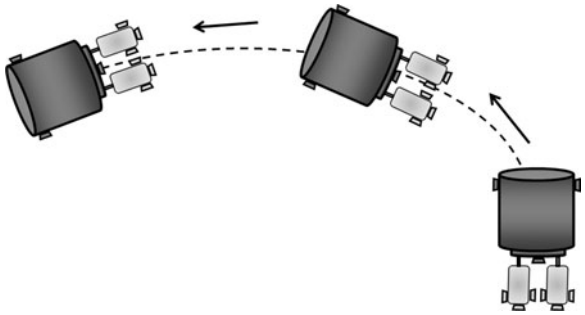

Fig. 6 Rescue assembly reference trajectory

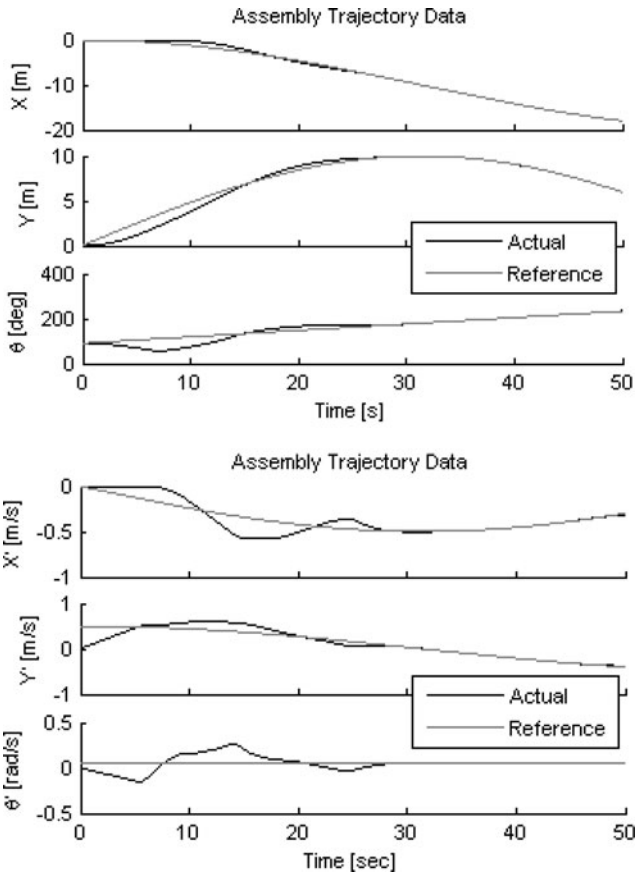

Fig. 7 Cooperative controller trajectory performance

3.2 Case 2: satellite rescue

The second set of simulations considers the rescue of an existing high-value satellite. Two rescue MSR modules assemble with and manoeuvre a larger satellite that has run out of fuel. The smaller MSR modules serve as a propulsion subsystem of the larger payload, characterized by a mass that is 2.5 times the mass of each MSR module. Figure 6 shows the assembly and its simulated trajectory.

Figure 7 shows trajectory performance under Cooperative Control. After some initial transient errors, the assembly tracks the trajectory well. These transient errors occur because the high inertia of the large disabled satellite causes the rescue modules' thrusters to saturate during the initial acceleration phase. In this case, additional modules or modules designed to provide higher thrust to a large load are required. While a redesigned configuration is required in this case, the value of the MSR rescue approach is suggested.

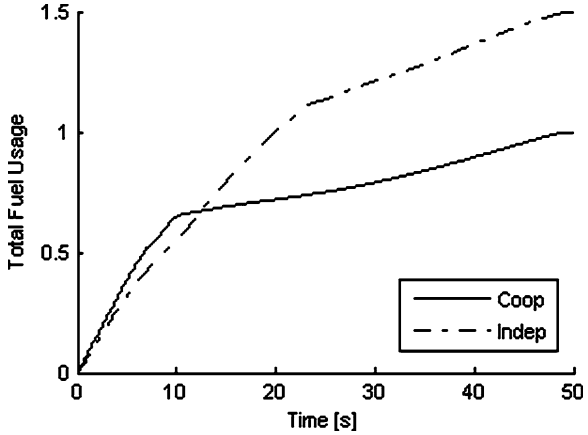

Fig. 8 Total fuel usage—cooperative vs. independent

For comparison, Independent Control of this case was also simulated. Independent control demonstrated significantly decreased performance in this case. Most notably, Fig. 8 shows that Independent Control consumes 50% more fuel. This likely resulted from fighting between the two independent module controllers as they tried to respond to the large initial trajectory errors resulting from high system mass and limited thrust.

4 Experimental validation

The MIT Field and Space Robotics Laboratory's experimental Free-Flying Space Robotics (FFR) test bed was used to study the proposed control algorithms, As in simulation, the performance of Cooperative Control and Independent Control are compared. Additional tests demonstrate the performance effects that adding Plume Impingment Constraints (PIC) or Fuel Balancing (FB) to a Cooperative Controller. Finally, benefits of sensor fusion are demonstrated on the test bed.

The FFR test bed consists of two 6.4 kg robot modules that float with CO_2 bearings on a 1.3 m $\times$ 2.2 m polished granite table to emulate microgravity in two dimensions (Hirzinger et al. 2004; Mohan and Miller 2008). See Fig. 9. Both robots are equipped with two Scara-type two-joint manipulators and eight thrusters. The CO_2 thrusters are pulse width modulated with a maximum thruster force of 0.1 N. On each module, two optical mice provide position, orientation, and velocity data, which have been shown to be very low noise (Bonarini et al. 2005). The manipulators have joint

angle encoders, and base force/torque sensors. The 7 DOF motions for each module (two translational and one rotational base motions and four manipulator joint motions) are controllable and observable. Each module is controlled by an on-board computer and has an onboard power supply. The onboard computers communicate with a fixed command computer using a wireless LAN, so that the modules are completely untethered.

For these experiments, the two robots are assembled into the configuration shown in Fig. 10 by magnetically connecting the manipulator end-effectors. While the manipulators are commanded to hold their pose, compliance in the manipulator structure and in the magnetic connection introduces some flexibility. Under Independent Control each module is controlled by its own onboard computer. For Cooperative Control the entire assembly is controlled by one module's processor. The remaining processor is available for computationally expensive tasks such as gain calculation. For both control approaches, the following weight matrices were used in the LQR design process:

$$H = 0_{n\times n}, \qquad Q = \begin{bmatrix} 10 \cdot I_{n/2\times n/2} & 0_{n/2\times n/2} \\ 0_{n/2\times n/2} & 30 \cdot I_{n/2\times n/2} \end{bmatrix}$$

$$R = I_{r\times r}$$

System performance is quantified using the previously introduced metrics of fuel consumption and trajectory tracking error. The amount of the fuel (CO_2 gas) consumed by the individual robots during each test is estimated from the thruster command history. These values do not include the CO_2 gas used to float the robots. For each test, trajectory tracking performance is summarized by the root mean square (RMS) position and orientation errors of each controller with respect to the commanded trajectory. For velocity tracking error only data from the final two thirds of each test is considered to permit initial transients to decay.

Two different reference trajectories are used. Most experiments utilize a simple constant-velocity translation 0.75 m in the Y direction with fixed orientation. A fixed-position constant-rate 90° rotation is also used. See Fig. 11.

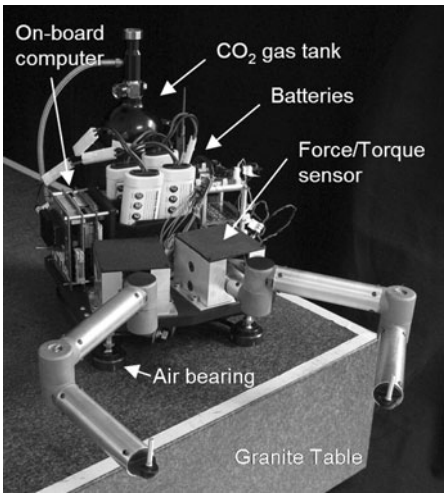

Fig. 9 FFR Test bed—module detailed description

4.1 Cooperative vs. independent control

The performance of Cooperative and Independent Controllers is compared using the linear translation trajectory.

Table 2 summarizes the experimental fuel consumption and trajectory RMS errors for the Cooperative and Independent Controllers. Cooperative Control significantly improves fuel consumption performance using 43% less fuel (45 g) than Independent Control (79 g).

These improvements are also visible in Fig. 12 which shows the cumulative fuel consumption of both controllers over time for one linear trajectory test. Notably, this plot shows a significant change in the slope of the Cooperative Controller consumption record at 1.5 s. The fuel consumption of the two controllers is nearly identical during the first

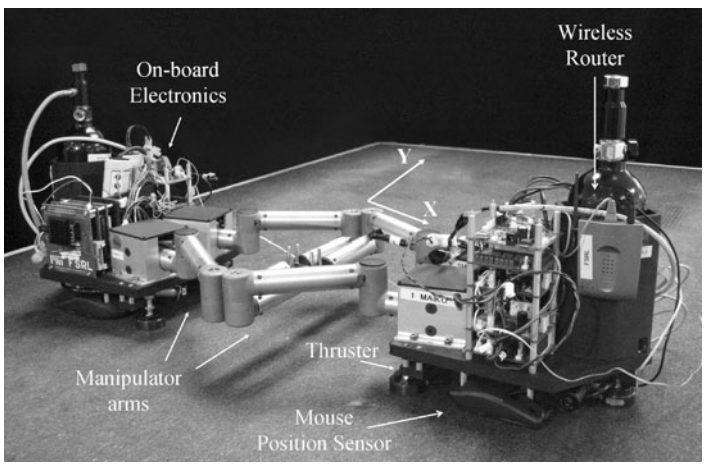

Fig. 10 FFR Test bed with a two module assembly

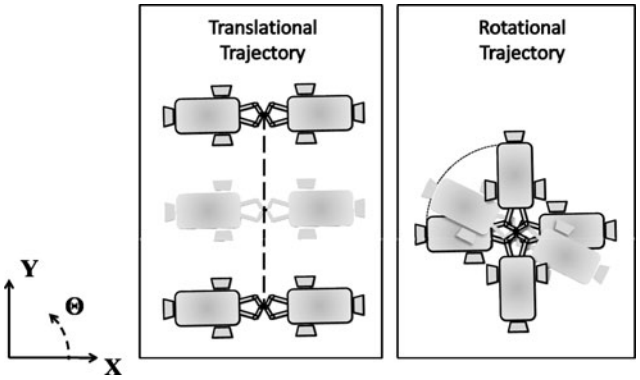

Fig. 11 Translational and rotational reference trajectory for FFR testbed

Table 2 Cooperative control vs independent control

	Cooperative	Independent
Total Fuel Consumption	45 g	79 g
X RMSE	0.9 cm	1.2 cm
Y RMSE	6.2 cm	6.3 cm
Y′ RMSE	2.4 cm s^{-1}	2.4 cm s^{-1}
Θ RMSE	0.164°	1.900°

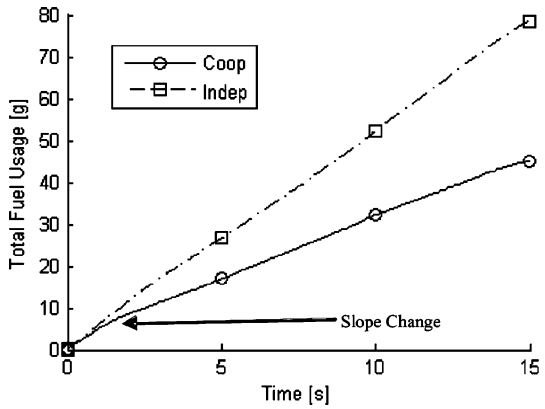

Fig. 12 Fuel usage for cooperative and independent control

1.5 seconds as the assembly accelerates to the trajectory's constant velocity. After this initial acceleration phase, under Cooperative Control the system enters a coasting phase indicated by the reduced slope. Thrusters fire at a much lower rate to maintain trajectory. Qualitatively, during experiments this transition was accompanied by a sudden reduction in thruster noise. However, under Independent Control, this coasting phase is never entered. Assembly miss-alignments and compliance coupled with the fighting between two controllers trying to correct trajectory errors result constant corrections that waste fuel.

Cooperative Control also exhibits slightly better trajectory tracking performance. Most notably, Table 2 shows that

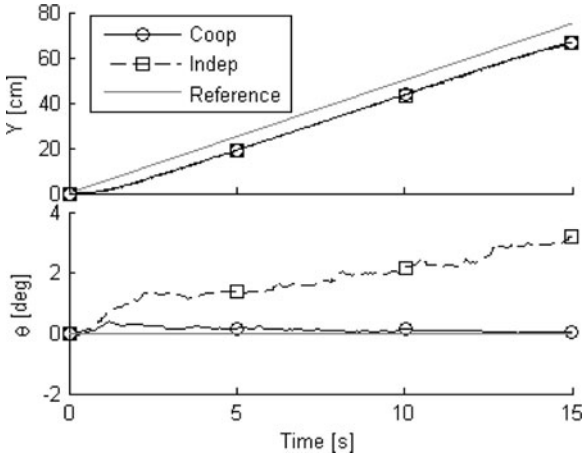

Fig. 13 Median Y and Θ RMS error for cooperative and independent control. Y performance is equivalent. Cooperative control has much lower Θ error

the Θ RMS error is reduced by 91%. Figure 13 plots representative trajectory tracking performance for the two controllers. This orientation performance improvement may be again attributed in part to the fact that multiple controllers are not issuing antagonistic commands. More fundamentally, a Cooperative Controller, directly monitors and corrects for assembly orientation error while an Independent Controller does not. Under Independent Control, the two modules attempt to individually execute adjacent trajectories. If both modules are properly positioned along their trajectories, the orientation results to be correct. Consequently, small position errors along the prescribed paths can combine into an orientation error for the assembly that is not monitored by the controller.

In order to verify that everything is in good agreement, data from the experiments are compared with those from the simulations. Figure 14 plots representative trajectory tracking performance for the experiments and simulations in the case of Cooperative and Independent Control. It is worth noting that the different performance in terms of orientation holds when comparing the two controllers, since both in experiments and simulation the performance of the Independent Control is significantly worse than results obtained with Independent Control.

Angular velocity data are reported in Fig. 15. Besides the high frequency noise that is not included in simulations, the sets of data show good agreement between experiments and simulation.

4.2 Plume impingement constraints

The effects of PIC are demonstrated by comparing the performance of a Cooperative Controller with and without PIC executing the stationary 90° rotation maneuver. Thrusters' position and orientation are showed in Fig. 16. Thrusters that

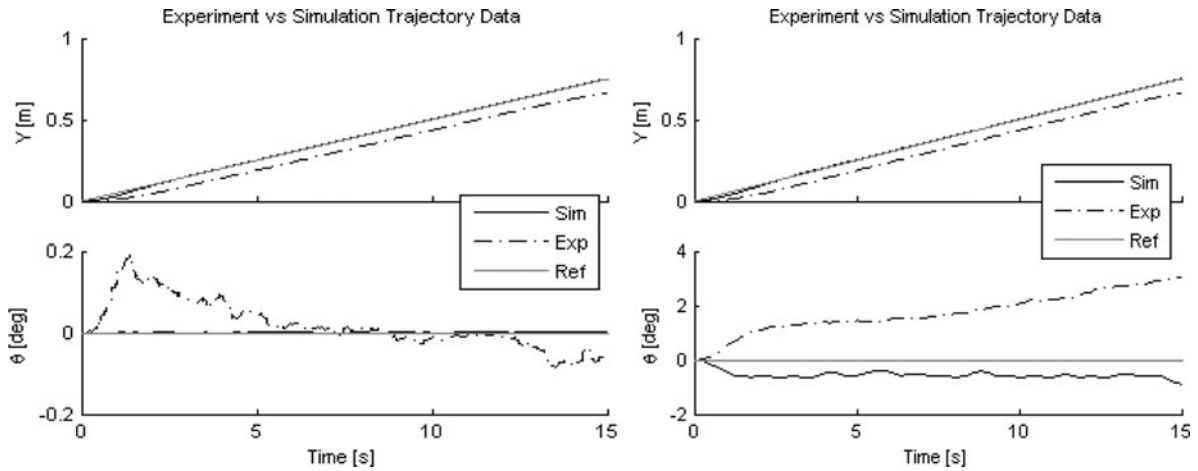

Fig. 14 Median Y and Θ trajectory for cooperative (*left*) and independent (*right*) control in experiments and simulations

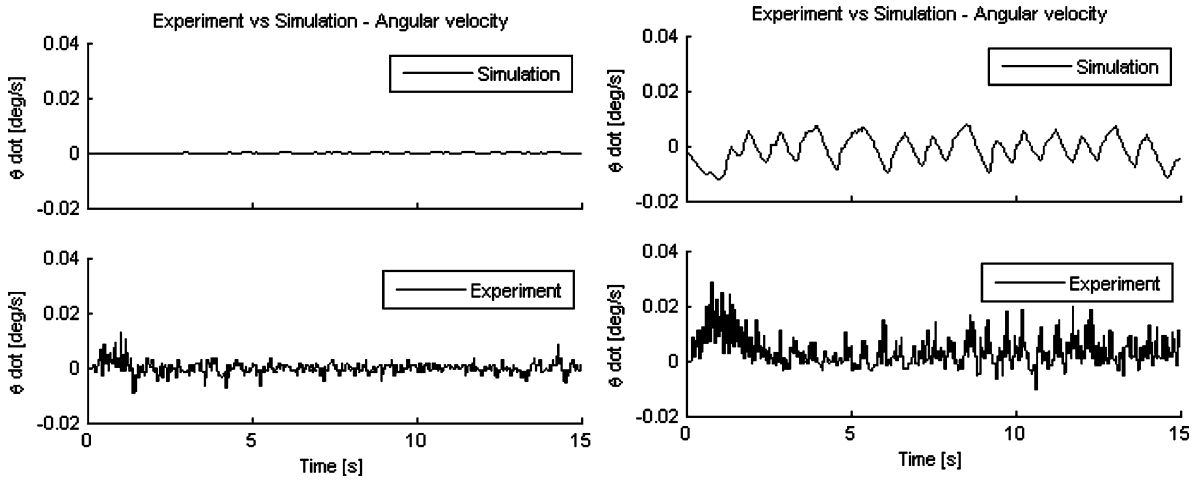

Fig. 15 Angular velocity trajectory for cooperative (*left*) and independent (*right*) control in experiments and simulations

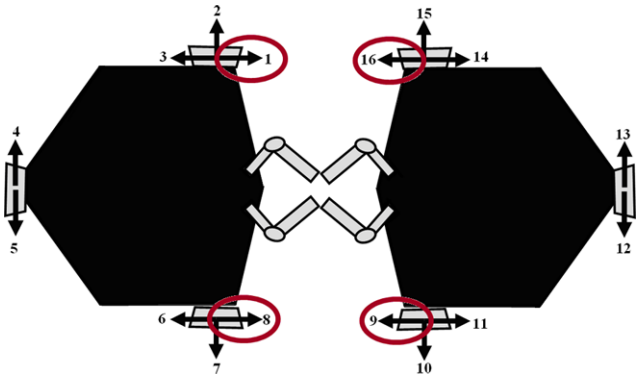

Fig. 16 Thursters position—FFR testbed

are poorly positioned and thus removed by PIC are highlighted.

Once the numeration of thrusters is defined as in Fig. 16, the matrix B_{PIC} is written as

$$B_{PIC} = \text{diag}\left(\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}\right)$$

Table 3 Cooperative control vs cooperative plume impingement constraint control

	Cooperative	Cooperative PIC
Total Fuel Consumption	147 g	120 g
X RMSE	0.7 cm	0.7 cm
Y RMSE	0.2 cm	0.3 cm
Θ RMSE	2.86°	2.95°
Θ′ RMSE	0.05° s^{-1}	0.05° s^{-1}

Table 3 shows the median experimental results for fuel consumption and trajectory RMS error. The Plume Impingement Constraint effectively prevented the use of poorly positioned thrusters and there were negligible reductions in trajectory tracking performance. However, PIC reduces total fuel consumption by 18%. This reduction occurs because thruster plumes are no longer directed against assembly surfaces, therefore there are no more reflected and dissipated thrust and wasted fuel.

Table 4 Cooperative control vs cooperative FB control

	Cooperative	Cooperative FB
Robot 1 Fuel Consumption	22 g	20 g
Robot 2 Fuel Consumption	22 g	23 g
X RMSE	0.9 cm	1.0 cm
Y RMSE	6.2 cm	6.3 cm
Y′ RMSE	2.4 cm s^{-1}	2.5 cm s^{-1}
Θ RMSE	0.16°	0.19°

4.3 Fuel balancing

Fuel Balancing is demonstrated by a Cooperative Controller with Fuel Balancing following the linear translation trajectory. In order to redistribute fuel consumption, the R weighting matrix was set to:

$$R = \begin{bmatrix} 2I_{r/2\times r/2} & 0_{r/2\times r/2} \\ 0_{r/2\times r/2} & I_{r/2\times r/2} \end{bmatrix}$$

penalizing the use of thrusters on robot 1 twice as much as on robot 2. For comparison the results of the Cooperative Controller without Fuel Balancing on the same trajectory are used.

Table 4 shows the experimental fuel consumption and trajectory RMS errors for the Cooperative Controller and the Cooperative Controller with Fuel Balancing (FB). Fuel consumption is listed for each module individually. Fuel Balancing shifts the consumption ratio between the two modules from 1:1 with the Cooperative Controller to 1:1.15 with the Cooperative FB Controller. With FB, robot 1 uses 91% of the fuel it used without fuel balancing. Robot 2 consumes 105% more in response to the FB adjustment. Total fuel consumption is slightly lower with FB, however, trajectory errors are marginally larger.

4.4 Sensor fusion

Sensor fusion was tested experimentally using the translational trajectory. Two sensing schemes are considered. The non-redundant approach estimates the assembly's state using base position and velocity data from only one module filtered with a Kalman-Bucy filter. Data from the second module's sensors is ignored. The redundant approach fuses sensor data from the two modules using the Kalman-Bucy filter. The FFR test bed obtains exceptionally clean position data from its converted optical mice sensors. Consequently, white noise is added to the raw sensor output. This raw sensor output is also saved and used as true baseline record of the assembly's state. Filter performance is evaluated by comparing the filtered state estimates with this baseline record and extracting RMS errors for each state variable. In all tests, the assembly is controlled using the same Cooperative Controller and gains.

Table 5 Sensing performance: error from baseline sensing

	Non-Redundant	Redundant
X RMSE	0.74 cm	0.02 cm
Y RMSE	2.24 cm	0.60 cm
Y′ RMSE	3.69 cm s^{-1}	2.33 cm s^{-1}

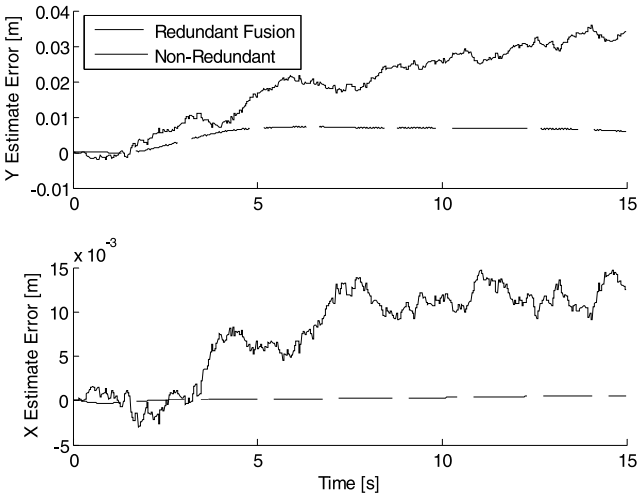

Fig. 17 Comparison of estimation errors with and without redundant sensor fusion

Table 6 Tracking performance: error from desired trajectory

	Non-Redundant	Redundant
Total Fuel Consumption	75 g	31 g
X RMSE	0.61 cm	0.23 cm
Y RMSE	10.24 cm	7.61 cm
Y′ RMSE	2.28 cm s^{-1}	2.33 cm s^{-1}

In Table 5, the differences between state estimates corrupted by noise and the original "baseline" state record are reported. The fusion of the redundant sensor data produces significantly better state estimates with lower RMS errors.

Errors in X and Y state estimates are compared in Fig. 17. As expected, using redundant sensor data greatly improves the accuracy of an assembly's state estimation and its ability to reject sensor noise.

These differences in filtered data quality have measurable effects on trajectory tracking performance. Table 6 shows the fuel consumption and trajectory tracking performance of the assembly with non-redundant and redundant filtering. Comparisons of fuel usage and trajectory tracking as functions of time are also shown in Fig. 18 and Fig. 19, respectively. With redundant sensing, the assembly uses less than half as much fuel as with non-redundant sensing. Similarly, the better state estimate produced by the redundant sensor

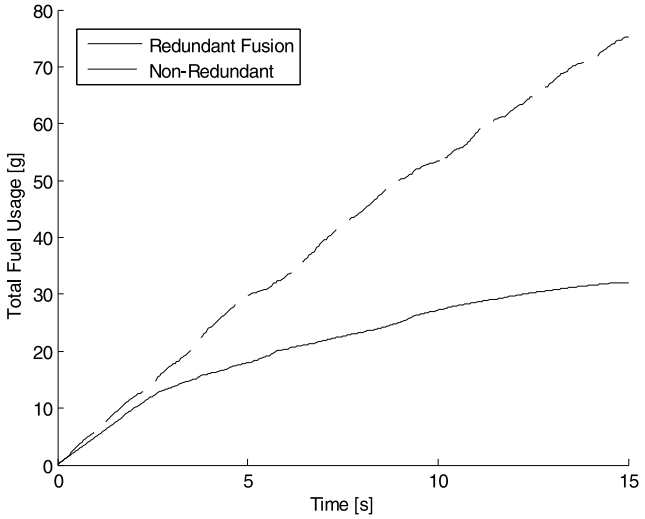

Fig. 18 Total assembly fuel usage histories

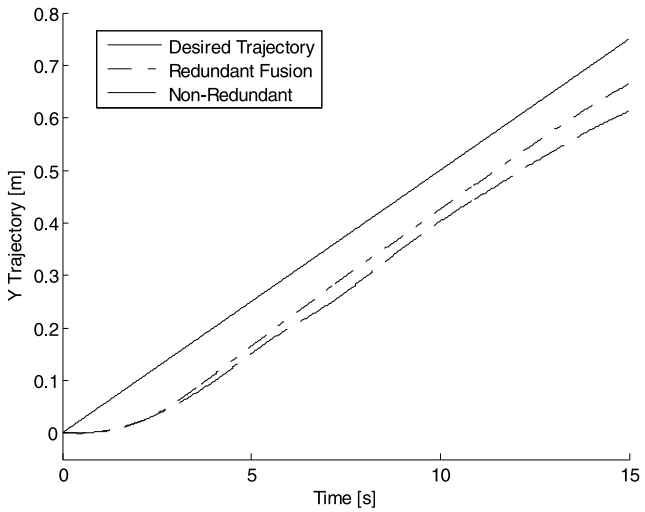

Fig. 19 Trajectory tracking: performance versus the desired trajectory

approach results in decreased tracking errors. The greater noise rejection that sensor fusion provides also prevents the ripple seen in the velocity of the system with non-redundant sensing in Fig. 19. This assembly accelerates and decelerates as it translates down the prescribed path in the slightly wavy quality of the position-time plot. These velocity errors resulted from the noisy state estimates produced by the non-redundant filter.

5 Conclusion

This work demonstrates effective control and sensing approaches for assemblies of spacecraft and space robots. The Cooperative Control method is developed and shown to be an effective control strategy for modular assemblies. Cooperative Control reduces conflicting thrust commands from the different modules promoting low fuel consumption. This is especially true in the presence of disturbances and errors that would cause Independent Controllers to fight when individually implementing corrective commands. A 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 Control can be autonomously updated without offline, human intervention by simply updating the system model. It can also easily include Plume Impingement Constraints, and Fuel Balancing. Similarly, sensor fusion through Kalman-Bucy filtering fully exploits the sensor redundancy provided by modular assemblies. By fusing sensor data, better noise rejection and more accurate estimates of the assembly state can be obtained. Cooperative Control and Kalman filtering are a unified, methodical, and general approach to the implementation of sensing and control for assemblies of spacecraft and space robots. These results are demonstrated in simulation and experimental studies.

In a real-world assembly, the control architecture can be implemented in a variety of ways. Designers should consider the need for one of the modules to assume a leadership function and the resulting tradeoffs between inter-module communication bandwidth, computational capabilities, and robustness to individual module failures. The details of this implementation architecture are beyond the scope of this study but should be considered as a subject for future research.

Acknowledgements Past and present members of the MIT Field and Space Robotics Laboratory have made substantial contributions to this research.

References

Bonarini, A., Matteucci, M., & Restelli, M. (2005). Automatic error detection and reduction for an odometric sensor based on two optical mice. In *Proceedings of the IEEE international conference on robotics and automation*, Barcelona, Spain, April.

Boning, P., Ono, M., Nohara, T., & Dubowsky, S. (2008). An experimental study of the control of space robot teams assembling large flexible space structures. In *Proc. of the 9th international symposium on artificial intelligence, robotics and automation in space*, Los Angeles, CA.

Clohessy, W. H., & Wiltshire, R. S. (1960). Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, *27*(9), 653–658.

Fehse, W. (2003). *Cambridge aerospace series: Vol. 16. Automated rendezvous and docking of spacecraft*. Cambridge: Cambridge University Press.

Hirzinger, G., Landzettel, K., Brunner, B., Fischer, M., Preusche, C., Reintsema, D., Albu-Schäffer, A., Schreiber, G., & Steinmetz, B. M. (2004). DLR's robotics technologies for on-orbit servicing. *The International Journal of the Robotics Society of Japan, Advanced Robotics*, *18*(2), 139–174.

Hughes, G. V. (1997). *The orbital express project of Bristol aerospace and MicroSat launch systems*. Washington: AIAA.

Inalhan, G., Busse, F. D., & How, J. P. (2000). Precise formation flying control of multiple spacecraft using carrier-phase differential GPS. In *Proc. guidance, control and navigation conference*.
Kennedy, F. (2007). Protection. In *Proceedings of DARPATech, DARPA's 25th systems and technology symposium*, Anaheim, California, August 8.
Kennedy, F. (2007). Oral presentation, DARPA's 25th Systems and Technology Symposium, August 8, Anaheim, California.
McCamish, S., Romano, M., & Yun, X. (2007). Autonomous distributed control algorithm for multiple spacecraft in close proximity operations. In *AIAA guidance, navigation and control conference*, Hilton Head, South Carolina, Aug. 20–23.
Mohan, S., & Miller, D. (2008). SPHERES reconfigurable control allocation for autonomous assembly. In *AIAA guidance, navigation and control conference and exhibit*, Honolulu, Hawaii, 18–21 Aug.
Mohan, S., Saenz-Otero, A., Nolet, S., Miller, D. W., & Sell, S. (2009). SPHERES flight operations testing and execution. *Acta Astronautica*.
Ono, M., Boning, P., Nohara, T., & Dubowsky, S. (2008). Experimental validation of a fuel-efficient robotic maneuver control algorithm for very large flexible space structures. In *Proc. of the IEEE international conf. on robotics and automation (ICRA)*, May, Pasadena.
Pizzicaroli, J. (1997). *Launching and building the Iridium® constellation*. IAF Workshop on Mission Design and Implementation of Satellite Constellations, Toulouse, France.
Sidi, M. (1997). *Spacecraft dynamics and control*. Cambridge: Cambridge University Press.
Spong, M., & Vidyasagar, M. (1989). *Robot dynamics and control*. New York: Wiley.
Stengel, R. (1994). *Optimal control and estimation*. New York: Dover.
Sweetman, W. (2008). Tiny independent coordinated spacecraft or TICS in space. www.aviationweek.com, August 8.
Tillerson, M., Breger, L., & How, J. P. (2003). Distributed coordination and control of formation flying spacecraft. In *Proceedings of the IEEE American control conference*, June.
Toglia, C., Kettler, D., Kennedy, F., & Dubowsky, S. (2009). A study of cooperative control of self-assembling robots in space with experimental validation. In *IEEE international conference on robotics and automation*, Kobe, Japan, 12–17 May.
Vadali, S. R., Vaddi, S. S., & Alfriend, K. T. (2002). An intelligent control concept for formation flying satellite constellations. *International Journal of Robust and Nonlinear Control*, *12*(2–3), 97–115.
Yoshida, K. (2001). ETS-VII flight experiments for space robot dynamics and control. *Experimental Robotics*, *VII*, 209–218.
Yoshida, K., Mavroidis, C., & Dubowsky, S. (1995). Dynamics and control of a space robot capturing a floating target. In *Proc. ISAS workshop on astrodynamics and flight mechanics* (pp. 108–114).

Chiara Toglia received the bachelor's degree in Aerospace Engineering (2004), the master's degree in Space Engineering (2006) and the PhD degree in Theoretical and Applied Mechanics (2010) from the University of Rome "La Sapienza". During her MSc she spent six month at TU Delft. She spent one year at the Field and Space Robotics Laboratory, M.I.T., as visiting PhD student. She attended the Space Studies Program (SSP09) of the International Space University in 2009 at NASA Ames Research Center. Her research interests concern Space Robotics and Control, UAV, Orbital Multibody Dynamics, Structures. She is currently working at Thales Alenia Space as AOCS engineer.

Fred Kennedy attended M.I.T. on an AFROTC scholarship and was commissioned a second lieutenant in June 1990. After receiving his master's degree in aeronautical and astronautical engineering, he was assigned to the Phillips Laboratory, Kirtland AFB, NM (1993–1996). He was selected as an Air Force Intern in 1996. Over the next two years, he served in three offices—the Air Force Secretariat, Office of the Secretary of Defense (Acquisition and Technology), and Joint Staff (J-8). After a stint at Squadron Officers' School in Montgomery, AL, he was reassigned to the National Reconnaissance Office in Chantilly, VA, in 1998 for three years. In 2001, he was selected to attend the University of Surrey, Guildford, UK, under the Air Force's Fellowship and Grant Program. He completed his PhD in 2004 and returned to the DC area as a liaison for the Air Force's Space Superiority Program Office and then as a program manager at DARPA. From 2004 to 2008, he led several space-based initiatives for DARPA's Tactical Technology Office. As the program manager for the Orbital Express program, he and his team demonstrated the ability to autonomously rendezvous, refuel, repair, and upgrade satellites on-orbit. He received the 2008 National Space Club's award for Best Collaboration between industry and government teams on a satellite program. He attended the US Army War College in Carlisle, PA, during the 2008-9 academic year and was reassigned to the Joint Staff in July 2009.

Steven Dubowsky received his Bachelor's degree from Rensselaer Polytechnic Institute of Troy, New York in 1963, and his M.S. and Sc.D. degrees from Columbia University in 1964 and 1971. He is currently a Professor of Mechanical Engineering at M.I.T. He has been a Professor of Engineering and Applied Science at the University of California, Los Angeles, a Visiting Professor at Cambridge University, Cambridge, England, and Visiting Professor at the California Institute of Technology. During the period from 1963 to 1971, he was employed by the Perkin-Elmer Corporation, the General Dynamics Corporation, and the American Electric Power Service Corporation. Dr. Dubowsky's research has included the development of modeling techniques for manipulator flexibility and the development of optimal and self-learning adaptive control procedures for rigid and flexible robotic manipulators. He has authored or co-authored nearly 200 papers in the area of the dynamics, control and design of high performance mechanical and electromechanical systems. Professor Dubowsky is a registered Professional Engineer in the State of California and has served as an advisor to the National Science Foundation, the National Academy of Science/Engineering, the Department of Energy, and the US Army. He has been elected a fellow of the ASME and IEEE and is a member of Sigma Xi, and Tau Beta Pi.