Laboratory Experimentation of Multiple Spacecraft Autonomous Assembly

Riccardo Bevilacqua¹, Andrew P. Caprari², Jason Hall³, and Marcello Romano⁴ Naval Postgraduate School, Monterey, California, 93943

This work introduces a novel approach and its experimental verification for propellant sub-optimal multiple spacecraft assembly via a Linear Quadratic Regulator (LQR). The attitude dynamics of the spacecraft are linearized at each time step, about the current state vector, and the relative dynamics between two spacecraft are assumed as a double integrator. This allows for implementation in real-time of a LOR that computes the optimal gain matrix depending on the current phase of the spacecraft's mission. As a result, both the attitude and position are sub-optimally controlled. The presented logic compensates for the structural evolution related to an incremental assembly by updating the system's dynamics matrices. The actuators' reallocation and command of the assembled structure is dealt with through inter-robot wireless ad-hoc communication. Each spacecraft runs symmetric algorithms, differing only in the number of docking ports that each possesses for the mission, which are related to the number of assembling spacecraft and the final structure's desired shape. Once the spacecraft are assembled, one acts as master by performing the required navigation and control of the new structure through real-time wireless commanding of the other spacecraft's actuators. The improved third generation (3G-i) of spacecraft simulators developed at the Spacecraft Robotics Laboratory SRL of the Naval Postgraduate School (NPS) is presented to demonstrate experimental verification of the proposed methodology. Features of the (3G-i) robots include an unique customized construction of rapid prototyped thermoplastic (polycarbonate) that incorporates a lightweight modular design with a small footprint, thus maximizing the entire surface of the SRL robotic testbed.

Nomenclature

а	=	Scaling Factor in R LQR Weighting Matrix
A	=	Dynamics Matrix
B	=	Control Matrix
С	=	State-Output Mapping Matrix
D	=	Control-Output Mapping Matrix
α	=	Docking Safety Cone semi-aperture
β	=	Commanded Orbiting Angle around target Docking Port in Docking Phase
$\begin{bmatrix} \varphi & \phi \end{bmatrix}$	$\vartheta]$	= Euler Angles
DOF	=	Degrees Of Freedom
LQR	=	Linear Quadratic Regulator

¹ Ph. D., US National Research Council Research Associate, Mechanical and Astronautical Department, code AA/RB, 699 Dyer Rd., AIAA Member, <u>rbevilac@nps.edu</u>.

² Master Thesis Student, US NAVY LT, Mechanical and Astronautical Department, code AA/AC, 699 Dyer Rd., AIAA Member, <u>apcaprar@nps.edu</u>.

³ Ph. D. Candidate, US NAVY LCDR, Mechanical and Astronautical Department, code AA/JH, 699 Dyer Rd., AIAA Member, jshall@nps.edu.

⁴ Ph. D., Assistant Professor, Mechanical and Astronautical Department, code MAE/MR, 700 Dyer Rd., AIAA Senior Member, <u>mromano@nps.edu</u>.

I_k	=	Identity Matrix of generic Dimension k
J	=	Cost Function
$J_z \& J_{z_z}$	omb	= Inertia of a single & combined Spacecraft Simulator about the vertical axis
K _{LQR}	=	LQR resulting Gain Matrix
$m \& m_{comb}$	=	Single and combined Mass of the Spacecraft Simulator
n	=	Number of Thrusters
NPS	=	Naval Postgraduate School
PWM	=	Pulse Width Modulation
ϱ	=	LQR State Error Weighting Matrix
SRL	=	Spacecraft Robotics Laboratory
ω	=	Spacecraft Simulator Angular Velocity
R	=	LQR Control Effort Weighting Matrix
r →	=	Forque Arm: Thruster-Center of Mass Arm
r_{rsw}	=	Spacecraft-to-Spacecraft Vector
r_{goal}	=	Docking port-to-corresponding docking Spacecraft Vector
\vec{r}_{obs}	=	Obstacle-to-Spacecraft Vector
\vec{r}_{port}	=	Center of Mass to Docking Port Vector
<i>r</i> _{dock}	=	Spacecraft-to-Spacecraft Transition Distance between Far Away Phase and Docking Phase
r_{coll}	=	Spacecraft-to-Spacecraft Collision Avoidance Triggering Distance
9	=	Spacecraft Simulator Heading (third Euler Angle)
t	=	Time
Т	=	Maneuver Total Time
ū	=	Control Vector
\vec{u}_{LQR}	=	Optimal Control Vector
$u_{\rm max} > 0$	=	Single Engine Maximum Thrust/Acceleration
$u_{thr} > 0$	=	Threshold Value for Required Thrust/Acceleration Before using PWM
V	=	Scaling Factor in Q LQR Weighting Matrix
<i>x</i> , <i>y</i> , <i>z</i>	=	Spacecraft Simulator Body Axis
\vec{x}	=	State Vector
$\vec{x}_{err} = \vec{x} - $	\vec{x}_{des}	ired = State Vector Error
X,Y,Z	=	Inertial Frame Axis (they are also used as Spacecraft Relative Coordinates in the Inertial Frame)
\vec{v}	=	System Output (measured states)

I. Introduction

T HE technical challenges presented by multiple spacecraft assembly are extremely topical and are thus being studied from many different aspects. One key thread of concern remains in the way of controlling a system that is continuously evolving, both in its mass and inertia properties, as well as in its sensor and actuator configuration. The works of Ref. 1, 2, 3 and 4 specifically address the problem of the evolution and control these types of systems from a theoretical point of view. In Ref. 5 more emphasis is given to a potential solution for the wireless connectivity of different parts intended for the assembly of a larger satellite, where a Wi-Fi bridge acts as the only real "assembly." In fact, wireless capability is becoming a more relevant option for exchanging data amongst rendezvousing satellites.⁶ Furthermore, the extremely high-risk environment surrounding an on-orbit assembly maneuver precludes the employment of such high performance logics as optimal controllers.⁷ Portions of this problem remain with the fact that onboard CPUs must allocate much of their performance capabilities to platform safety issues and not be overloaded by heavy computational logics. Nevertheless, much research effort is being undertaken by NASA and other agencies to address new trends and operationally appealing advancements in aerospace technology. In particular, several of these research threads focus on the issue of increasing the computational capabilities of the onboard computers while others are concentrated on the effective use of COTS hardware on space systems.

Previous work by the authors on multiple spacecraft can be found in References 8, 9, and 10. In this work we present a new algorithm that functions as an autonomous controller for multiple spacecraft assembly. Three different phases of the assembly are described as follows:

- 1. A rendezvous phase representing LQR driven control of individual spacecrafts from arbitrary initial conditions.
- 2. A docking phase that minimizes plume impingement.

3. An assembled phase representing LQR driven control of the assembled structure to a new target point. Phase 1 and 2 are independently sub-optimized from the propellant consumption point of view. This is achieved by continuously linearizing the dynamics about the current state vector and employment of a LQR based control algorithm. In particular, a time-varying LQR function is implemented in Simulink® that can be automatically converted into an executable via MATLAB®'s Real-Time-WorkshopTM. In this way, the optimization acts as a feedback control by occurring at each time step during the experiment. Ultimately the LQR solver is implemented onboard a real-time operating system. This solver is a direct extension of what is used in Ref. 11 and Ref. 12 for simplified problem-targeted LQR execution. A version of the Simulink® based LQR solver for both RTAI Linux and xPC TargetTM is available for download.¹³

A research similar to this work has been presented in Ref. 14, where the authors employ an off-line LQR-generated, pre determined trajectory for a two spacecraft simulator docking, without optimization of the attitude motion.

The main contributions of this research to the state of the art for multiple spacecraft assembly are:¹⁵

- 1. Development of an autonomous LQR based logic for multiple spacecraft assembly that adapts to the changing shape, mass properties and sensors/ actuators configuration by simple modification of a minimal number of parameters within the control algorithm.
- Demonstration of control capability via an LQR in real-time given an arbitrary non-predetermined 2. trajectory, i.e. the optimization is performed "on the fly".
- To the authors' knowledge, the first time, on-the-ground experimentation of autonomous, non-3. preprogrammed, assembly and control of assembled structures.

The terms robot, spacecraft, and spacecraft simulator are used synonymously throughout this work. The paper is organized as follows. Section II is an overview of the relative motion and attitude dynamics. Section III describes the sub-optimal attitude and position control using a Linear Quadratic Regulator. Section IV introduces the improved third generation robots developed at the NPS SRL in Monterey, California. Section V reduces the LQR to the 3DOF environment utilized for the described experiment. Section VI describes the last phase of docking and how the plume impingement problem is faced. Section VII is dedicated to the navigation and control of the assembled robots. Section VIII provides the supporting experimental results. Lastly, Section IX is the Conclusion.

II. Relative Motion and Attitude Dynamics

The relative position dynamics between two spacecraft and the single spacecraft attitude dynamics can be represented in the form of Eq. (1). We assume the spacecraft to be controlled only by n body-fixed thrusters. We will also work in terms of accelerations, assuming the spacecraft to have constant mass throughout the maneuvers. From here-on the words thrust and accelerations will be exchanged without any loss of generality.

$$\dot{\vec{x}} = \vec{f}\left(\vec{x}\right) + \boldsymbol{B}\left(\vec{x}\right) \cdot \vec{u} \tag{1}$$

 $\vec{x} = \begin{bmatrix} X & Y & Z & \phi & \phi & \dot{x} & \dot{Y} & \dot{Z} & \dot{\phi} & \dot{\phi} & \dot{\beta} \end{bmatrix}^T$. The drift vector $\vec{f}(\vec{x})$ simplifies into a linear dynamics $A\vec{x}$ when the relative motion between the spacecraft is linearized as in the Clohessy-Wiltshire equations (Ref. 16)

or even more simplified as in a double integrator, which will be our assumption, and the attitude dynamics are linearized at each time step around the current attitude state $\begin{bmatrix} \varphi & \phi & \theta \end{bmatrix}^T$. With these assumptions, Eq. (1) reduces to

$$\dot{\vec{x}} = \boldsymbol{A} \cdot \vec{x} + \boldsymbol{B} \left(\vec{x} \right) \cdot \vec{u}, \qquad \boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_{6x6} & \boldsymbol{I}_{6x6} \\ \boldsymbol{0}_{6x6} & \boldsymbol{0}_{6x6} \end{bmatrix}$$
(2)

The $B(\vec{x})$ control distribution matrix maps the *n* acceleration values from the body frame into the inertial frame. In other words, it translates body-fixed forces (from the thrusters) in forces and torques into the inertial frame. In such a way, the nonlinearities in the dynamics equations due to the coupling between translation and

rotation are taken into account allowing one to work directly on the forces exerted by the thrusters without the need of thruster mapping. $B(\vec{x})$ depends on the particular geometry of the thruster configuration on the spacecraft and the spacecraft's attitude and thus it is not specified other than in the robotic testbed case of Section V.

III. Sub-Optimal Attitude-Position Control via Linear Quadratic Regulator

This section describes how the LQR optimal control is designed to maneuver both attitude and position of the assembling spacecrafts. Figure 1 shows the principal vectors used by the control algorithm for one possible configuration in which it is assumed that the docking ports do not have to be aligned with any particular body axis. Figure 2 depicts a body fixed docking port vector that is referred to later in the algorithm development.



Figure 1 Example of the relative vectors used in: alignment, docking and assembly algorithms.



Figure 2 Example of body fixed docking port vector.

The LQR problem seeks to determine the control sequence that minimizes the cost function in Eq. 3. The problem is solved at each time step with dynamically sized weighting matrices Q and R that adapt to the current

situation to avoid high control values even if the state error is relevant. As a result, the process occurs in such a way that there is no need to specify a reference trajectory, i.e. the LQR results define the way the spacecraft is maneuvering.

$$J = \int_{0}^{T} \left(\vec{x}_{err}^{T} Q \vec{x}_{err} + \vec{u}^{T} R \vec{u} \right) dt$$
(3)

In this particular work, we leave the trajectory for the center of mass unconstrained and free to be optimized unless in the vicinity of the docking phase. As for the attitude, we reproduce a realistic condition that commands each spacecraft to show the same face (normally the face with the docking port) towards the objective spacecraft. In other words, the face with the docking port is driven to be perpendicular with the \vec{r}_{rsw} or the \vec{r}_{goal} vector (Figure 1), depending on the phase of the mission. The following list describes phases 1 and 3 presented in the introduction, i.e. the LQR driven phases for a single spacecraft and an assembled spacecraft while they are not in the very final docking phase. Each spacecraft in Figure 1 can be considered either a single platform or an already assembled structure. The following description applies to both scenarios. In the following, the orthogonal vectors are always intended to be parallel to the XY plane because out of plane motion is canceled out when the spacecrafts are in the assembly phase. Nevertheless, the LQR approach suggested drives any residual Z to zero.

1. $|\vec{r}_{rsw}| > r_{dock}$, RENDEZVOUS: the spacecraft is at a far away distance from its target docking port. The state vector to minimize is $\vec{x}_{err} = \begin{bmatrix} \vec{r}_{rsw}^T & \varphi - \varphi_{des} & \varphi - \varphi_{des} \end{bmatrix}^T$ with its time derivatives. The desired Euler angles are such to show the chosen side to the target docking spacecraft, i.e. $\perp \vec{r}_{rsw}$.

$$\boldsymbol{Q} = \begin{bmatrix} \frac{1}{\left|\vec{r}_{goal}\right|} \cdot \boldsymbol{I}_{6} & \boldsymbol{0}_{6} \\ \boldsymbol{0}_{6} & \left|\vec{r}_{goal}\right|^{3} \cdot \boldsymbol{V} \cdot \boldsymbol{I}_{6} \end{bmatrix}, \qquad \boldsymbol{R} = \frac{\left|\vec{r}_{goal}\right|}{a^{2}} \boldsymbol{I}_{n}$$
(4)

- 2. $|\vec{r}_{rsw}| \le r_{dock}$, DOCKING APPROACH: the spacecraft is close to its target docking port. The state vector to minimize is either a or b below:
 - a. If $\cos^{-1}\left(\frac{\vec{r}_{goal} \cdot \vec{r}_{port}}{|\vec{r}_{goal}||\vec{r}_{port}|}\right) < \alpha$, the spacecraft is within the security docking cone and there are two sub cases:

CASE 1: if the distance between the spacecraft is greater than the chosen impingement stand-off range, then $\vec{x}_{err} = \begin{bmatrix} \vec{r}_{goal}^T & \varphi - \varphi_{des} & \varphi - \varphi_{des} \end{bmatrix}^T$ with its time derivatives. The desired Euler angles are such to show the chosen side to the target docking spacecraft, i.e. $\perp \vec{r}_{rw}$.

$$\boldsymbol{Q} = \begin{bmatrix} \frac{1}{\left|\vec{r}_{goal}\right|} \cdot \boldsymbol{I}_{6} & \boldsymbol{0}_{6} \\ \boldsymbol{0}_{6} & \left|\vec{r}_{goal}\right|^{3} \cdot \boldsymbol{V} \cdot \boldsymbol{I}_{6} \end{bmatrix}, \qquad \boldsymbol{R} = \frac{\left|\vec{r}_{goal}\right|}{a^{2}} \boldsymbol{I}_{n}$$
(5)

CASE 2: the distance between the spacecraft is less than the chosen impingement stand-off range, then the LQR control is shut off and the docking logic takes over, as described in Section VI.

b. If $\cos^{-1}\left(\frac{\vec{r}_{goal} \cdot \vec{r}_{port}}{|\vec{r}_{goal}||\vec{r}_{port}|}\right) \ge \alpha$, the spacecraft is outside the security docking cone. In this case, referring

to spacecraft 2 of Figure 1, the vehicle performs an orbiting maneuver around the one hosting its target docking port and continues moving in a direction perpendicular to the \vec{r}_{rsw} vector. This trajectory is by definition the shortest path to the safety corridor. The amount of rotation around the target docking port is a chosen constant parameter β . With respect to the LQR's goal of minimizing state vector error, a reference frame is defined with the basis unit vectors $\hat{r}_{rsw\perp}, \hat{r}_{rsw}$. In this manner, \vec{r}_{rsw} can be rotated by an angle β into \vec{r}_{rot} and easily expressed as function of the basis $\vec{r}_{rot} = |\vec{r}_{rsw}| (\cos \beta \hat{r}_{rsw} + \sin \beta \hat{r}_{rsw\perp})$. Thus, the state error to minimize is $\vec{x}_{err} = [(\vec{r}_{rot} - \vec{r}_{rsw})^T \quad \varphi - \varphi_{des} \quad \varphi - \varphi_{des} \quad \vartheta - \vartheta_{des}]^T$ along with its time derivatives. The desired Euler angles are such to show the chosen side to the target docking port, i.e. $\perp \vec{r}_{sod}$.

$$\boldsymbol{Q} = \begin{bmatrix} \frac{1}{\left|\vec{r}_{goal}\right|} \cdot \boldsymbol{I}_{6} & \boldsymbol{0}_{6} \\ \boldsymbol{0}_{6} & \left|\vec{r}_{goal}\right|^{3} \cdot \boldsymbol{V} \cdot \boldsymbol{I}_{6} \end{bmatrix}, \qquad \boldsymbol{R} = \frac{\left|\vec{r}_{goal}\right|}{a^{2}} \boldsymbol{I}_{n}$$
(6)

Each time step solution of the LQR results in a gain matrix K_{LQR} that is used to generate the required optimal control vector of Eq. 7. Figure 3 shows the required inputs to the LQR solver.

 $\vec{u}_{LOR} = \boldsymbol{K}_{LOR} \cdot \vec{x}_{err}$



(7)

Figure 3 LQR Solver Simulink® Block

The LQR solver employed for developing the proposed approach was initially downloaded from Ref. 13 and modified from its original Windows® only support to support full Linux/Unix support and automatic code generation through Real-Time-Workshop[™] for RTAI Linux. This enhanced support solver now resides for public accessibility at Ref. 13.

IV. Improved Third Generation (3G-i) Spacecraft Simulators at the Spacecraft Robotics Laboratory

This section introduces the improved third generation of spacecraft simulators developed at the NPS SRL. Figure 4 shows the fleet of operational spacecraft simulators. The simulators float using air bearings over a very smooth, flat

epoxy floor, reproducing a nearly frictionless environment in three dimensions, i.e. two degrees of freedom for the translation and one for the rotation. Although the experiments utilize three degrees of freedom (3DOF) which differs from the real world (6DOF) environment, they allow for the verification of the integration between the algorithms and the sensors/actuators, thus accurately reproducing the dynamics of multi-satellite close proximity flight. Ultimately, the main goals of the SRL team are the development and experimental testing of navigation and control logics for maneuvering multiple spacecraft systems during proximity operations.



Figure 4 Multi-Spacecraft Test-bed at the SRL, US NPS.

In order to perform docking experimentations, two separate custom-designed, rapid prototyped docking interfaces have been developed at the SRL, and each is currently undergoing experimental testing (see Figure 5)



Figure 5 (a) Patent pending docking interface design (electro-magnet and fluid transfer capability) (Ref. 17) (b) Concept (male/female) docking interface used for the experiment validating this work

The type 1 docking interfaces are designed in order to passively connect the spacecraft through electro-magnetic mechanisms, and their design will allow data/power/fluids exchange (see Figure 6 and Ref. 17). Conversely, the type 2 design lacks the before mentioned characteristics but enhances the robustness on the docking concept by correcting residual translation and rotational errors developed during the final docking phase of the spacecraft assembly. This second design also implements two small permanent magnets to provide a final docking force, and keep the robots physically connected during assembled maneuvers.



Figure 6 Main components of the patent pending docking interface as shown on the 3G spacecraft (Ref. 17)

Other key features of the (3G-i) spacecraft simulators include:

- 1. Ad-hoc wireless communication: Continuous data exchange amongst simulators and the external wireless network environment provide for insitu communication. This greatly increases the robustness of data collection in the event there is a loss of one communication to one of the spacecraft simulators.
- 2. **Modularity:** The simulators are divided into two modules where the payload can be disconnected from the consumables, thus allowing for a wide range of applications with virtually any kind of payload (Figure 7).
- 3. **Small footprint:** The 0.19m length x.019m width of each simulator allows for the working area (~5m x5m) on the epoxy floor to be optimally exploited.
- 4. Light weight: (~10 kg) and getting lighter by the generation.
- **5. Rapid Prototyping:** The capability to rapidly reproduce further generations of simulators and improve existing designs via computer aided design (CAD) with the in-house STRATASYS 3D printing machine.

Most notably, point 1 of the previous list has provided an invaluable contribution to the success of the ongoing experimentation. The ad-hoc wireless communication system, currently employed onboard the (3G-i) simulators, was experimentally verified by a distributed computing test, which demonstrated the wireless communication real time capability for the SRL.⁶

Figure 7 shows the robot's operating configuration with the docking interface removed with key hardware components annotated.



Figure 7 Spacecraft Simulator Main Components

Table 1 illustrates the characteristics of the electronics used on board each spacecraft simulator. The PC104 (onboard computer), the sensors, and the actuators are described below (see figures 7, 9 and Ref. 6).

PART'S NAME and		-
MANUFACTURER	DETAILS	DESCRIPTION
PC104 (plus) Motherboard		
(Advanced Digital Logic)	processor	SmartCoreT3-400, 400Mhz CPU
	RAM	SDRAM256-PS
Compact Flash		
(SanDisk Extreme IV)	-	8 Gbyte capacity
20 Relays Board (IR-104-PBF)		High Density Opto-isolated Input +
(Diamond Systems)	-	Relay Output
8 Serial Ports Board (MSMX104+)		
(Advanced Digital Logic)	-	-
Firewire PC104 board (Embedded		
Designs Plus)	-	IEEE1394 Card with 16 Bit PC104
Compact Wireless-G USB Adapter		54Mbps 802.11b/g Wireless USB
(Linksys)	-	Network Interface Adapter
Wireless Pocket Router/AP DWL-		2.4 Ghz 802.11g, ethernet to
G730AP (D-Link)		wireless converter
Solenoid Valves (Predyne)	-	2 way, 24VDC, 2Watt
Fiber Optic Gyro DSP3000 (KVH)		Single axis rate, 100Hz,
	-	Asynchronous, RS-232
Magnetometer, MicroMag-3Axis		Asynchronous, RS-232 (the
(evaluation kit with RS232 board)		evaluation kit is still a development
(PNI)	-	version)
DC/DC converters: EK-05 Battery		
Controller and Regulator + DC1U-		3.3, 5, 12, 24 Volts outputs. The
1VR 24V DC/DC Converter		main board is equipped with a
(Ocean Server)	-	batteries' status controller.
Battery (Inspired Energy)		Lithium Ion Rechargeable Battery
	-	(95 Whr)
Metris iGPS pseudo-GPS indoor		
system	-	-

Table 1 Electronics Hardware Description

Figure 8 depicts the main concept of the robotic testbed at the SRL. The main components and their interfaces are illustrated onboard the schematic of a representative simulator at the bottom of the sketch. The figure emphasizes the fact that the configuration is scalable to an arbitrary number of robots depending on the application or mission.



The Wi-Fi capability of each robot is not only used to communicate with other robots, but is also necessary for receiving its own absolute position within the laboratory, as sensed by the pseudo-GPS indoor system.

The onboard real-time operating system is RTAI patched Linux of the Debian 2.6.19 type.¹⁸ Classical use of xPC TargetTM by MathWorksTM as a real-time operating system (OS) is common in academic research.¹⁹ A key advantage of xPC TargetTM is its seamless integration between Simulink[®] via Real-Time WorkshopTM, which allows for rapid prototyping of navigation and control algorithms for real-time requirements. Real-Time WorkshopTM automatically generates C code from a Simulink[®] model and the corresponding executable file for a xPC TargetTM based computer. On the other hand, xPC TargetTM has some disadvantages that include support for a limited number of hardware components and no support for USB or Firewire devices. In addition, the inaccessibility of its source code, due to its proprietary commercial nature, makes it challenging to add or modify drivers for unsupported hardware.

RTAI Linux has been successfully used as an onboard real-time OS. RTAI is a patch to the Linux kernel that allows for the execution of real-time tasks in Linux.^{20,21} The RTAI Linux solution is being widely exploited in several engineering areas.²²⁻²⁵ In this work, we use RTAI Linux with a wide variety of hardware interfaces to include wireless ad-hoc radio communication using UDP, RS232 interface with the sensor suite and power system and a PC/104 relay board for actuating compressed air nozzles. RTAI Linux also allows for automatic generation of C code from Simulink[®] models through Real-Time WorkshopTM with the executable file for the onboard computers being created outside MATLAB[®] by simple compilation of the C code. Further details on the ad-hoc wireless network and hardware-software interfaces developed for the spacecraft simulators are available in Ref. .6

V. Specializing the LQR Problem to the 3 DOF Spacecraft Simulators

In reducing the problem to the 3DOF environment of the SRL spacecraft simulators, the eight body-fixed thrusters can be treated in couples so that symmetric thrusters are reduced to one control variable, i.e. u_{max} , $-u_{\text{max}}$ or 0. Figure 9 shows the thruster couplings: 1-4, 2-7, 3-6, and 5-8. Through thruster coupling, the LQR is used to solve a reduced problem with a four vice eight dimensional control vector defined by $\vec{u} = [u_1, u_2, u_3, u_4]^T$. The red arrows along the couples in Figure 9 depict the positive directions assumed for the controls.



Figure 9 Thruster Coupling on the Spacecraft Simulators

The equations of motion for a spacecraft simulator reduce to the form $\dot{\vec{x}} = A\vec{x} + B\vec{u}$, with A, B given in Eq. (8), the control vector described as $\vec{x} = \begin{bmatrix} X & Y & \mathcal{B} & \dot{X} & \dot{Y} & \dot{\mathcal{B}} \end{bmatrix}^T$. The system is assumed to be completely observable $(\vec{y} = \vec{x})$.

Linearization of the B matrix leads to

$$B_{LN} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\cos(g_{des}) - \sin(g_{des})(g - g_{des})}{m} & \frac{\cos(g_{des}) - \sin(g_{des})(g - g_{des})}{m} & \frac{-\sin(g_{des}) + (\cos(g_{des}))(g - g_{des})}{m} & \frac{-\sin(g_{des}) + (\cos(g_{des}))(g - g_{des})}{m} & \frac{-\sin(g_{des}) - \sin(g_{des}) + (\cos(g_{des}))(g - g_{des})}{m} & \frac{\cos(g_{des}) - \sin(g_{des}) + (g - g_{des})}{m} & \frac{\cos(g_{des}) - \cos(g_{des}) + (g - g_{des})}{m} & \frac{\cos(g_{des}) - \cos(g_{de$$

By inputting the matrices defined in Eq. (8) and (9) and the weighting matrices described in Section III into the LQR solver of Figure 3, the optimal four dimensional control vector of accelerations is obtained at each time step during the maneuver as given by Eq. (7). Given the continuous nature of the obtained control vector and the discrete nature of the on/off thrusters, a Pulse Width Modulation (PWM) scheme is used. Furthermore, a threshold acceleration value u_{thr} is used before the PWM to filter out low commanded controls and reduce the amount of chattering.

VI. Control during Docking

This section describes the spacecraft maneuvering logic when point (2, a., CASE 2) of Section III occurs. During the docking phase, when the docking ports are aligned, the thrusters on the spacecraft's docking faces (Figure 5) must be minimally used to avoid plume impingement. Residual misalignment of the docking spacecraft is allowed by the interface's design, thanks to the cone-hollow cone shape.

This limitation in controls availability is tackled by designing a control law as follows (refer to Figure 9, when thrusters 6 and 7 are the ones not to be used):

- 1. This is the only portion of the overall algorithm when a trajectory is tracked. In particular, the \vec{r}_{rsw} vector, at the moment when point (2, a., CASE 2) of Section III occurs, is "frozen" in the inertial frame and represents the common line of docking that the two spacecraft must follow.
- 2. A Proportional/Derivative (PD) controller, using only side thrusters 1, 4, 5 and 8 of Figure 9, corrects alignment errors between the spacecraft's \vec{r}_{dock} and \vec{r}_{rsw} .
- 3. If the error angle between \vec{r}_{dock} and \vec{r}_{rsw} is below a chosen threshold, the same side thrusters act according to the PD law in order to maintain the spacecraft's center of mass on the docking line.
- 4. If the error angle between \vec{r}_{dock} and \vec{r}_{rsw} is below a chosen threshold and the approach velocity along the docking line is below a chosen value, thrusters 2 and 3 activate in order to provide a docking force.
- 5. If the approach velocity along the docking line is too high, the deactivated thrusters 6 and 7 are turned on for an emergency brake to avoid high speed collision.

VII. Navigation and Control of the Assembled Structure

Once the robots are assembled, the mass and inertia properties along with the thruster configuration change. Figure 10 shows an example applied to the SRL testbed, in which thrusters six and seven of both spacecrafts cannot be used anymore. The new assembled spacecraft has a doubled mass, different moment of inertia, and four additional thrusters that are reallocated with respect to the single spacecraft. In the assembled configuration, one of the robots acts as a master and performs both navigation and control of the newly evolved system. Nevertheless, in order to keep using the same logic employed for controlling a single robot, the twelve thrusters of the new assembled spacecraft are reallocated according to the following sets:

- 1. u_1 is generated by firing either 8 ($u_1 < 0$) or 3 ($u_1 > 0$);
- 2. u_2 is generated by firing either 9 ($u_2 < 0$) or 2 ($u_2 > 0$);
- 3. u_3 is generated by firing either 6 & 7 synchronously ($u_3 < 0$) or 11 & 10 synchronously ($u_3 > 0$);
- 4. u_4 is generated by firing either 4 & 5 synchronously ($u_4 < 0$) or 1 & 12 synchronously ($u_4 > 0$);



Combined Center of Mass (C.O.M.) shift

Figure 10 Assembled Configurations with Reallocated Thruster coupling and COM shift

The input matrices to the LQR solver are changed once an additional portion of the structure is connected. Also, the new control vector of accelerations has its maximum and minimum values reduced due to the increase of mass. For instance, the case represented in Figure 10 leads to the new matrices

Linearization of the new \boldsymbol{B} matrix leads to

$$B_{LN} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\cos(g_{ds}) - \sin(g_{ds})(g - g_{ds})}{m_{carb}} & \frac{\cos(g_{ds}) - \sin(g_{ds})(g - g_{ds})}{m_{carb}} & \frac{-\sin(g_{ds}) + (\cos(g_{ds}))(g - g_{ds})}{m_{carb}} & \frac{-\sin(g_{ds}) + (\cos(g_{ds}))(g - g_{ds})}{m_{carb}} & \frac{-\sin(g_{ds}) + (\cos(g_{ds}))(g - g_{ds})}{m_{carb}} \\ \frac{\sin(g_{ds}) + (\cos(g_{ds}))(g - g_{ds})}{m_{carb}} & \frac{\sin(g_{ds}) + (\cos(g_{ds}))(g - g_{ds})}{m_{carb}} & \frac{\cos(g_{ds}) - \sin(g_{ds})(g - g_{ds})}{m_{carb}} & \frac{\cos(g_{ds}) - \sin(g_{ds})(g - g_{ds})}{m_{carb}} \\ \frac{-r}{J_{z_carb}} & \frac{r}{J_{z_carb}} & \frac{r$$

Where $J_{z_{-comb}}$ is the inertia of the assembled system about the vertical axis. The thrusters that remain available after docking will be commanded by either spacecraft one or two through the real-time wireless communication link.⁶ Furthermore, navigation for the assembled structure is performed following the rigid body assumption onboard the acting master robot (Eq. 12, see also Figure 2).

$$\vec{x}_{assembled} = \begin{bmatrix} \vec{r}_{assembled} & \mathcal{G}_{assembled} & \vec{v}_{assembled} & \dot{\mathcal{G}}_{assembled} \end{bmatrix}^{T} = \begin{bmatrix} \vec{r}_{master} + \vec{r}_{port} & \mathcal{G}_{master} & \vec{v}_{master} + \vec{\omega}_{master} \times \vec{r}_{port} & \dot{\mathcal{G}}_{master} \end{bmatrix}^{T},$$

$$\vec{\omega}_{master} = \begin{bmatrix} 0 & 0 & \dot{\mathcal{G}}_{master} \end{bmatrix}$$

$$\vec{r}_{master} = \begin{bmatrix} X & Y & 0 \end{bmatrix}$$
(12)

VIII. Experimental Results

This section is dedicated to a two spacecraft experimental test. The goal of the experiment is to dock two simulators and then control the assembled structure to a new target point. Table 2 summarizes the principal constants and chosen parameters for the presented experiment.

Table 2 Experimental Parameters

Simulator Mass m, m_{comb}	10, 20	Kg
Inertia of Simulator J_z, J_{z_comb}	0.063, 0.18	Kg m ²
Single Thruster Estimated Force		C
$u_{\rm max}$	0.16	Ν
Docking Cone Semi-aperture α	0.75	degrees
Force arms (for torque generation)		
r, r_i, r_o	5, 10, 21	cm
Limit distance for docking phase		
r _{dock}	1	m
Limit distance for switching off the		
thrusters	0.7	m
LQR parameter <i>a</i>	0.03	m/s ²
LQR parameter V	0.03	m/s
Orbiting angle in docking orbiting	25	degrees
Chattering avoidance thrust threshold	30% of	
	$u_{\rm max}$	Ν
Angle error threshold for following		
docking line in docking phase	0.5	degrees
Minimum approach velocity for	0.04	
docking	0.01	m/s
Maximum approach velocity for	0.02	,
docking	0.03	m/s

Figure 11 shows the bird's eye view of the performed test. Spacecraft 1 (black) is chosen as master for the assembled navigation and control. Two robots start at arbitrary initial conditions on the floor, dock at approximately half way, and the new structure (blue) is controlled to a stand-off range on the docking side of a virtual third robot (green, at coordinates (2,2)). The docking occurs at approximately 80 seconds, and the alignment of the assembled spacecraft in front of the third target is completed in about 300 seconds.

The comment about the camcorder position on the floor in Figure 11 refers to a video of the experiment, available upon request to the authors.

The compressed air consumption is estimated as a ΔV by adding the thrust value of 0.16N multiplied by the

opening times of the thrusters, along the whole experiment. The final value for this test was 9.8 m_s .



Figure 11 Experimental result: Bird's Eye View of two-spacecraft assembly

Figure 12 presents the heading of the master robot and the assembled structure after docking. Figure 13 shows the angular rate of the master and the assembled spacecraft about the vertical axis. Figure 14 and Figure 15 are the center of mass and velocity on the floor for the master robot and the assembled system. Figure 16 represents the on/off thrust commands for spacecraft 1 while Figure 17 is the thrust history for spacecraft 2 during the while maneuver to docking. Figure 18 is the assembled thrust history with half of the thrusters commanded on-board spacecraft 1, which acts as master, and half of the thrusters commanded on spacecraft 2 via wireless communication link. Of note, the use of thrusters 6 and 7 is dramatically reduced during the very last phase of docking (see Figure 16 and Figure 17) due to the introduced logic in Section VI.



Figure 12 Experimental result: Spacecraft Simulator heading during two-spacecraft assembly



Figure 13 Experimental result: Spacecraft Simulator Angular Velocity during two-spacecraft assembly



Figure 14 Experimental result: Spacecraft Simulator Velocity on x axis during two-spacecraft assembly



Figure 15 Experimental result: Spacecraft Simulator Velocity on y axis during two-spacecraft assembly



Figure 16 Experimental result: Spacecraft Simulator 1 Thrust History. At t=80 seconds the new structure's thrusters activate, the original single-robot commands go to zero.



thrusters activate, the original single-robot commands go to zero.



 $\underset{092407}{\underbrace{22}}$ American Institute of Aeronautics and Astronautics



Figure 18 Experimental result: Assembled Spacecraft Simulator Thrust History. At t=80 seconds the new structure's thrusters activate, the original single-robot commands go to zero.

IX. Conclusion

This work presents a novel technique for sub-optimal fuel consumption for multiple spacecraft assembly and its experimental verification. It also introduces the new robotic testbed developed at the NPS SRL that is used to validate the proposed methodology. The approach relies on linearized attitude dynamics, a Linear Quadratic Regulator implemented as a Simulink® block that can be converted into a real-time executable, and an algorithm that executes the following: collision-free docking maneuvers, seamless accounting of the evolving systems changing mass and inertial properties, and reallocation of the thruster commands and controls for an autonomously assembled spacecraft configuration to mission completion. It further relies on the ability to communicate via the adhoc wireless network from experiment initialization throughout mission critical docking evolutions utilizing custom, CAD, rapid prototyped docking interfaces. A successful two spacecraft experiment is performed where two spacecraft simulators dock and the new assembled spacecraft is maneuvered to a third target point using the same logic employed on a single agent. Ultimately, this work aims to leverage the research and developments in the areas of GNC and propellant-optimal control for autonomous multiple spacecraft assemblies.

Acknowledgments

This research was partially supported by DARPA. This research was performed while Dr. Bevilacqua was holding a National Research Council Research Associateship Award at the Spacecraft Robotics Laboratory of the US Naval Postgraduate School.

References

- 1 Frost, S. A., Balas, M. J., "Adaptive Key Component Controllers for Evolving Systems," AIAA Guidance, Navigation and Control Conference and Exhibit, 18 - 21 August 2008, Honolulu, Hawaii, AIAA 2008-6279.
- 2 Balas, M. J., Frost, S. A., "Evolving Systems: A Theoretical Foundation," AIAA Guidance, Navigation, and Control Conference and Exhibit, 21 - 24 August 2006, Keystone, Colorado, AIAA 2006-6333.
- 3 VanZwieten, T. S., Balas, M. J., "Evolving Systems: Inheriting Stability with Decentralized Adaptive Control," AIAA Guidance, Navigation and Control Conference and Exhibit, 20 - 23 August 2007, Hilton Head, South Carolina, AIAA 2007-6504.
- 4 Frost, S. A., Balas, M. J., "Stabilizing Controllers for Evolving Systems with Application to Flexible Space Structures," AIAA Guidance, Navigation and Control Conference and Exhibit, 20 - 23 August 2007, Hilton Head, South Carolina, AIAA 2007-6663.
- 5 Dong, S., Allen, K., et al., "Self-assembling wireless autonomously reconfigurable module design concept," ACTA Astronautica, vol. 62, pages 246-256, Jan 2008.
- Bevilacqua, R., Hall, J. S., Horning, J., Romano, M., "Ad-hoc Wireless Networking and Shared Computation based upon Linux for Autonomous Multi-Robot Systems," AIAA Journal of Aerospace Computing, Information, and Communication. Vol. 6, No 5, pp. 328-353, May 2009. Doi: 10.2514/1.40734.
- 7 Bevilacqua, R., Romano, M., "Fuel-Optimal Spacecraft Rendezvous with Hybrid On-Off Continuous and Impulsive Thrust," Journal of Guidance, Control, and Dynamics 2007, 0731-5090 vol.30 no.4 (1175-1178).
- 8 Lugini, C., Romano, M., "A Ballistic-Pendulum Test Stand to Characterize Small Cold-gas Thruster Nozzles," Acta Astronautica," Vol. 64, No. 5-6, pp. 615-625, March-April 2009.
- 9 Romano, M., Friedman, D.A., Shay, T.J., "Laboratory Experimentation of Autonomous Spacecraft Approach and Docking to a Collaborative Target," AIAA Journal of Spacecraft and Rockets, Vol. 44, No. 1, pp. 164-173, January-February 2007.

- ¹⁰ Hall, J.S., Romano, M., "Robotic Testbed for the Experimentation of Guidance and Control of Spacecraft During Proximity Maneuvers," Mechatronic Systems, ISBN 978-953-7619-46-6, In-Tech, Vienna, Austria, August 2009.
- ¹¹ McCamish, S., Romano, M., Nolet, S., Edwards, C., Miller, D., "Ground and Space Testing of Multiple Spacecraft Control During Close-Proximity Operations," AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, Aug. 18-21, 2008, AIAA-2008-6664.
- ¹² McCamish, S.B., Romano, M., Nolet, S., Edwards, C.M., Miller, D. W., "Flight Testing of Multiple Spacecraft Control on SPHERES During Close Proximity Operations," AIAA Journal of Spacecraft and Rockets. Accepted for publication. To appear.
- ¹³ http://mathworks.com/matlabcentral/fileexchange/2651.
- ¹⁴ Toglia, C., Kettler, D., Kennedy, F., Dubowsky, S., "A Study of Cooperative Control of Self-Assembling Robots in Space with Experimental Validation," 2009 IEEE International Conference on Robotics and Automation, May 12 - 17, 2009, Kobe, Japan.
- ¹⁵ Caprari, A.P., "Evolving Robotics Systems Navigation and Control: Simulation and Hardware-In-The- Loop Testing for Multiple Spacecraft Autonomous Assembly," M.S. Thesis, Department of Astronautical and Mechanical Engineering, Naval Postgraduate School, Monterey, CA, 2009.
- ¹⁶ Clohessy, W. H., Wiltshire, R. S., "Terminal Guidance System for Satellite Rendezvous," Journal of the Aerospace Sciences, Vol. 27, No. 9, 1960, pp. 653-658.
- ¹⁷ Romano, M., Oppenheimer, P., "Spacecraft Docking Interface Scalable to Very Small Spacecraft" (patent pending).
- ¹⁸ https://www.rtai.org/RTAILAB/RTAI-Lab-tutorial.pdf
- ¹⁹ http://www.mathworks.com/products/xpctarget/
- ²⁰ Dozio, L., Mantegazza, P., "Real time distributed control system using RTAI," Sixth IEE International Symposium on Object-Oriented Real-Time Distributed Computing, May 2003.
- ²¹ Quaranta, G., Mantegazza, P., "Using MATLAB-Simulink RTW to Build Real Time Control Applications in User Space with RTAI-LXRT," Realtime Linux Workshop, Milan, Italy, 2001.
- ²² Ricci, S., Scotti, A., "Aeroelastic Testing on a Three Surface Airplane," 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 1 - 4 May 2006, Newport, Rhode Island AIAA, 2006-2189.
- ²³ Dozio, L., Toso, A., Corbetta, W., Vigoni, E., Ghiringhelli, G., "Comparison of Feedback and Feedforward Strategies for Active Structural–Acoustic Control of Broadband Sound Transmission into a Cavity," 13th AIAA/CEAS Aeroacoustics Conference (28th AIAA Aeroacoustics Conference), AIAA 2007-3575.
- ²⁴ Perhinschi, M. G., Napolitano, M. R., Campa, G. Seanor, B., Gururajan, S., Yu, G., "Design and Flight Testing of Intelligent Flight Control Laws for the WVU YF-22 Model Aircraft," AIAA Guidance, Navigation, and Control Conference and Exhibit, 15 - 18 August 2005, San Francisco, California, AIAA 2005-6445.
- ²⁵ Zheng, Y., Martin, B. J., Villaume, N., "VERSE Virtual Equivalent Real-time Simulation Environment," AIAA Modeling and Simulation Technologies Conference and Exhibit, 15 - 18 August 2005, San Francisco, California, AIAA 2005-6021.