Hindawi Publishing Corporation
International Journal of Aerospace Engineering
Volume 2011, Article ID 308245, 18 pages
doi:10.1155/2011/308245

*Research Article*

# Guidance Navigation and Control for Autonomous Multiple Spacecraft Assembly: Analysis and Experimentation

**Riccardo Bevilacqua,[1] Marcello Romano,[2] Fabio Curti,[3] Andrew P. Caprari,[4] and Veronica Pellegrini[5]**

[1] *Department of Mechanical, Aerospace & Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA*
[2] *Department of Mechanical and Aerospace Engineering and Space Systems Academic Group, Naval Postgraduate School, Monterey, CA 93943-5100, USA*
[3] *Dipartimento di Ingegneria Aerospaziale e Astronautica, Scuola di Ingegneria Aerospaziale, Universitá di Roma "La Sapienza", 00138 Roma, Italy*
[4] *Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA 93943-5100, USA*
[5] *Department of Applied Mathematics and Statistics, University of California, Santa Cruz, CA 95064, USA*

Correspondence should be addressed to Riccardo Bevilacqua, bevilr@rpi.edu

Received 31 August 2010; Revised 22 October 2010; Accepted 23 December 2010

Academic Editor: Giampiero Campa

This work introduces theoretical developments and experimental verification for Guidance, Navigation, and Control of autonomous multiple spacecraft assembly. We here address the in-plane orbital assembly case, where two translational and one rotational degrees of freedom are considered. Each spacecraft involved in the assembly is both chaser and target at the same time. The guidance and control strategies are LQR-based, designed to take into account the evolving shape and mass properties of the assembling spacecraft. Each spacecraft runs symmetric algorithms. The relative navigation is based on augmenting the target's state vector by introducing, as extra state components, the target's control inputs. By using the proposed navigation method, a chaser spacecraft can estimate the relative position, the attitude and the control inputs of a target spacecraft, flying in its proximity. The proposed approaches are successfully validated via hardware-in-the-loop experimentation, using four autonomous three-degree-of-freedom robotic spacecraft simulators, floating on a flat floor.

## 1. Introduction

The technical difficulties presented by the autonomous multiple spacecraft assembly problem relate to the development of robust and reliable guidance, navigation, and control techniques for on-orbit evolving systems. The main open challenges are: (1) propellant-efficient control of an assembling (also known as evolving system), the evolution occurring both in its mass and inertia properties, as well as in its sensors and actuators configuration and (2) accurate relative navigation among the spacecraft, especially in the event of low frequency measurements update and interruptions of measurements due, for example, to relative sensors' view's obstruction by other spacecraft. The works of [1–4] address specifically the problem of a system's evolution and its control. In [5], more emphasis is given to a potential solution for the wireless connectivity of different parts intended for the assembly of a bigger spacecraft, where a Wi-Fi bridge acts as the only real "assembly." Furthermore, wireless capability is becoming a more relevant option for exchanging data amongst close proximity spacecraft which eventually dock to each other (see [6]). Also, the high-risk situation of an assembly maneuver in space does not leave room for computationally intensive logics, such as optimal controllers (see [7]). Onboard CPUs must allocate most of their performance capabilities to platform safety issues.

The use of Commercial Off-The-Shelve (COTS) relative sensors, such as low-cost cameras, justify the need for robust relative navigation schemes. Many different filters for tracking a maneuvering target have been considered in the literature.

Approaches based on Kalman filter include the work of Singer [8], in which the target acceleration is modeled as a random process with known exponential autocorrelation. The input estimation approach for tracking a maneuvering target is proposed by Chan and Couture [9]. In this approach, the magnitude of the acceleration is identified by the least-squares estimation when a maneuver is detected. The estimated acceleration is then used in conjunction with a standard Kalman filter to compensate the state estimate of the target. The standard filter alone is used during periods when no maneuver takes place. The augmented filtering approach is proposed by Bar-Shalom et al. [10, 11]. In this approach, the state model for the target is changed by introducing extra state components, the target's accelerations. The maneuver, modeled as an acceleration, is estimated recursively along with other states associated with position and velocity while a target maneuvers. Bogler [12] used this method as an implementation on high maneuver target tracking with maneuver detection.

The input estimation filter and the augmented-dimension filter are commonly used in view of their computational efficiency and tracking performance. Among input estimation techniques, the Augmented State estimation approach yields reasonable performance without constant acceleration or small sampling time assumptions. Furthermore, it not only provides fast initial convergence rate, but it can also track a maneuvering target with fairly good accuracy as mentioned by Khaloozadeh and Karsaz [13]. Bahari et al. [14] and Bahari and Pariz [15] propose an intelligent error covariance matrix resetting, by a fuzzy logic approach, necessary for high maneuvering target tracking, to improve the estimation of the target state.

In space applications, particularly in the spacecraft relative navigation for the autonomous rendezvous and assembly, each vehicle is both the target and the chaser for the other spacecraft. Here, an additional challenge is considered: the frequent loss of communications for the data exchange when the application involves more than one spacecraft. Alternatives means to perform relative navigation may include a vision-based system. These types of sensors require the image processing and may result in low frequency measurement updates, especially for small spacecraft with limited computation capabilities. Such sensors suffer of problems such as limitations on the field of view and/or other spacecraft obstructing the view. Furthermore, each vehicle does not usually know the other vehicles inputs, that is, it does not possess the information about the maneuvers performed by its fellow spacecraft. This missing information needs to be reconstructed in the estimation scheme that would otherwise diverge quickly.

We here focus on the utilization of low frequency update and low-cost sensors, such as COTS devices. In particular, the spacecraft are envisioned to have range and line of sight measurements, and relative attitude measurements. The navigation algorithm here presented build upon our preliminary work of [16].

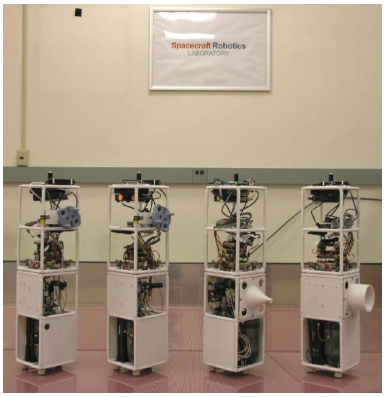


Figure 1: Multispacecraft testbed at the Spacecraft Robotics Laboratory of the Naval Postgraduate School.

In this work, we build upon known techniques in order to develop guidance, navigation, and control approaches to perform three-degree-of-freedom spacecraft assembly maneuvers. Furthermore, the suggested methodologies are validated via hardware-in-the-loop testing, using four robotics spacecraft simulators.

In particular, the guidance and control problems are tackled by continuously linearizing the dynamics about the current relative state vector between two spacecraft, and employing a Linear Quadratic Regulator to suboptimize propellant consumption. The LQR weighting matrices are computed in real time, depending on the relative state vector, acting as a feedback control. The LQR real-time solver developed for this research is an extension of what used during a real on-orbit spacecraft test inside the International Space Station [17], where a simplified problem-targeted LQR was executed (a version of the LQR Simulink solver for both RTAI Linux and xPC Target is available for download) (see [18]). While the system evolves, changing its mass properties and actuators' configuration, the LQR-based approach remains unaltered, controlling the growing structure by the simple online modification of a few parameters when a new spacecraft is docked.

As for the relative navigation, we here propose a design based on the augmented state estimation technique. Robustness to frequent signal loss and/or darkening of the sensors is achieved. Furthermore, the suggested approach reconstructs the information of the other vehicles' maneuvers. A spacecraft is envisioned to run a copy of the augmented state estimation technique algorithm for each other spacecraft in the bunch, every vehicle being chaser and target at the same time.

For the experimental part of this work, two dynamic models for the relative navigation filter are considered: (1) the classical Kalman filtering technique, [19], in which the unknown input (the maneuver command) is modeled as a random process and (2) the augmented state estimation technique, where the maneuver is estimated, using a Kalman filter scheme [19], in real time, as an additional variable in an augmented state vector.

Between the two approaches, the second one proves to be the most successful. It yields satisfactory performances without constant accelerations or small sampling time assumptions. Furthermore, it does not only provide fast initial convergence rate, but it can also track a maneuvering target with a good accuracy under unpredictable loss of the data link and slow data rate, allowing the spacecraft to perform critical maneuvers such as the docking and the multivehicle assembly.

The successful results of the study here presented pave the way for further research and implementation of the new GNC techniques for the full six degrees of freedom spacecraft relative motion.

Main contributions of this work to the state-of-the-art for multiple spacecraft assembly GNC are as follows.

(1) Development of a guidance and control approach flexible to mass and actuators' configuration changes during the assembly. The methodology is based on a suboptimal LQR for propellant-efficient rendezvous and docking maneuvers.

(2) Implementation of a spacecraft relative navigation scheme based on augmented state estimation, robust to low frequency measurements updates. In particular, the spacecraft are envisioned to have the availability of range and line of sight measurements, and relative attitude measurements. No relative velocities measurements are available. This is the first time, to our knowledge, that augmented state vector estimation is used for spacecraft relative navigation.

(3) The first (to the best of authors' knowledge) hardware-in-the-loop laboratory experiment involving four spacecraft simulators in a completely autonomous assembly maneuver.

The paper is organized as follows. Section 2 presents the robotic spacecraft simulators employed for the experiments. Section 3 presents the equations of the three-degree-of-freedom motion for the spacecraft relative maneuvering. Section 4 presents the augmented estimation approach and demonstrates the observability of the augmented state. Section 5 illustrates the guidance and control. Section 6 describes how navigation and control are performed once more spacecraft are assembled. Section 7 is dedicated to the experimental validation of the proposed methodologies. Section 8 concludes the paper.

## 2. Third Generation Spacecraft Simulators at the Spacecraft Robotics Laboratory

This section introduces the third generation of spacecraft simulators developed at the Spacecraft Robotics Laboratory of the US Naval Postgraduate School. Figure 1 shows the fleet of operational spacecraft simulators. The simulators float using air bearings over a very smooth epoxy floor, reproducing a nearly frictionless and weightless environment in two dimensions and three degrees of freedom, that is, two degrees of freedom for the translation and one for the rotation. This experimental testbed allows for the partial verification of guidance, navigation, and control algorithms in a simulated in-plane close proximity flight condition [20]. For more details on the different families of spacecraft simulators employed throughout the world, we address the reader to [6, 16, 20–23].

In order to perform docking experimentations, two separate custom designed docking interfaces have been developed and each is currently undergoing experimental testing (see Figure 2).

The type 1 docking interfaces are designed in order to passively connect the spacecraft through electromagnetic mechanisms, and their design will allow data/power/fluids exchange (see Figure 3). Conversely, the type 2 design lacks the afore mentioned characteristics but enhances the robustness on the docking concept by correcting residual translational and rotational errors developed during the final docking phase of the spacecraft assembly for experimentation. This second design hosts two small permanent magnets to provide a final docking force and to keep the robots physically connected.

Other key features of the spacecraft simulators include the following.

(1) Ad-hoc wireless communication. Continuous data exchange amongst each simulator and the external environment over the wireless network provides for in situ communication. This greatly increases the robustness of data collection in the event of communication loss with one of the 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 different payload (Figure 4).

(3) Small footprint. The .19 m length × .19 m width of each simulator allows for the working area (~ 5 m × 5 m) on the epoxy floor to be optimally exploited.

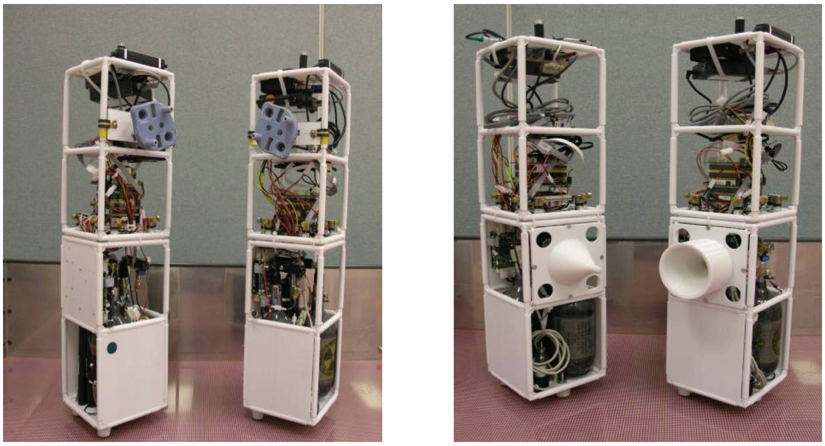
(4) Light weight. ~10 kg.

(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 our ongoing experimentation. The ad-hoc wireless communication system, currently employed onboard the simulators, was experimentally verified by a distributed computing test, which demonstrated the wireless communication real-time capability for the SRL (see [6]).

Table 1 illustrates the characteristics of the electronics used onboard each spacecraft simulator. The PC104 (onboard computer), the sensors, and the actuators are described below (see [6]).

Each robot performs absolute navigation in the laboratory environment employing indoor pseudo-GPS for position, and magnetometer and gyroscope for attitude (Table 1). The measurements from these sensors are processed by two separate Linear Digital Kalman Filters, estimating position and velocity of the center of mass with respect to the



(a) Type 1

(b) Type 2

Figure 2: (a) Patent pending docking interface design (electromagnet and fluid transfer capability). (b) Concept (male/female) docking interface used for the experiments in this paper.

Table 1: Electronics hardware description.

| Part’s name and manufacturer | Details | Description |
|---|---|---|
| PC104 (plus) Motherboard<br>(Advanced Digital Logic) | Processor<br>RAM | SmartCoreT3-400, 400 Mhz CPU<br>SDRAM256-PS |
| Compact Flash<br>(SanDisk Extreme IV) | — | 8 Gbyte capacity |
| 20 Relays Board (IR-104-PBF)<br>(Diamond Systems) | — | High-density optoisolated input + relay output |
| 8 Serial Ports Board (MSMX104+)<br>(Advanced Digital Logic) | — | — |
| Firewire PC104 board (Embedded Designs Plus) | — | IEEE1394 card with 16 Bit PC104 |
| Compact Wireless-G USB Adapter (Linksys) | — | 54 Mbps 802.11 b/g wireless USB network interface adapter |
| Wireless Pocket Router/AP DWL-G730AP<br>(D-Link)<br>Solenoid Valves (Predyne) | — | 2.4 Ghz 802.11 g, ethernet to wireless converter<br>2 way, 24VDC, 2 Watt |
| Fiber Optic Gyro DSP3000 (KVH) | — | Single axis rate, 100 Hz, Asynchronous, RS-232 |
| Magnetometer, MicroMag-3Axis (evaluation kit with RS232 board) (PNI) | — | Asynchronous, RS-232 (the evaluation kit is still a development version) |
| DC/DC converters: EK-05 Battery Controller and Regulator + DC1U-1VR 24V DC/DC Converter (Ocean Server) | — | 3.3, 5, 12, 24 Volts outputs. The main board is equipped with a batteries’ status controller. |
| Battery (Inspired Energy) | — | Lithium Ion Rechargeable battery (95 Whr) |
| Metris iGPS pseudo-GPS indoor system | — | — |

laboratory reference frame, and heading and heading rate of the robot with respect to the laboratory frame. The details on the robots’ absolute navigation are beyond the scope of this work, and they will not be discussed here; for additional information, the reader can refer to [20].

The maximum computational power of 400 Mhz listed in Table 1 is not required for real-time recomputation of the LQR solution. In the SPHERES satellites [17], Texas Instruments C6701 Digital Signal Processor is employed to solve a very comparable problem.

Figure 5 depicts the main concept of the testbed at the SRL. The main components and their interfaces are illustrated onboard the robot at the bottom of the sketch. Furthermore, 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 it is also necessary for receiving its own absolute position within the laboratory, as sensed by the pseudo-GPS indoor system.

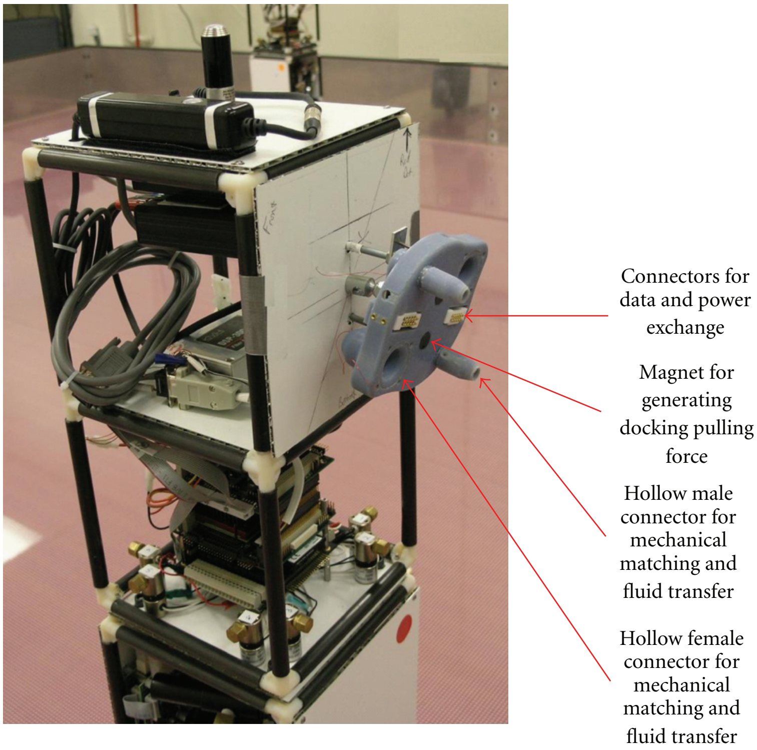

Connectors for data and power exchange
Magnet for generating docking pulling force
Hollow male connector for mechanical matching and fluid transfer
Hollow female connector for mechanical matching and fluid transfer

Figure 3: Main components of the patent pending docking interface.

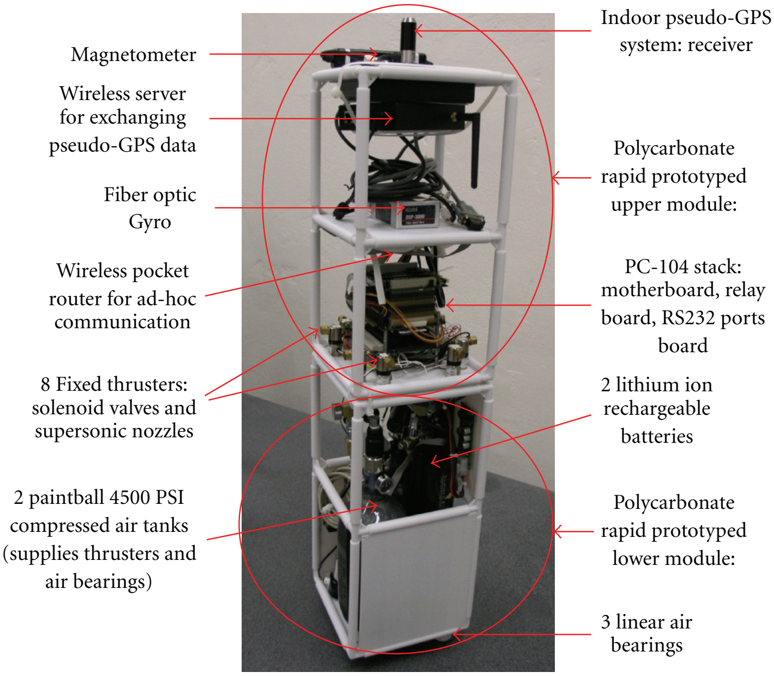

Magnetometer
Wireless server for exchanging pseudo-GPS data
Fiber optic Gyro
Wireless pocket router for ad-hoc communication
8 Fixed thrusters: solenoid valves and supersonic nozzles
2 paintball 4500 PSI compressed air tanks (supplies thrusters and air bearings)
Indoor pseudo-GPS system: receiver
Polycarbonate rapid prototyped upper module:
PC-104 stack: motherboard, relay board, RS232 ports board
2 lithium ion rechargeable batteries
Polycarbonate rapid prototyped lower module:
3 linear air bearings

Figure 4: Detailed collocation of the hardware on the spacecraft simulators.

The onboard real-time operating system is RTAI patched Linux (see [24]), in the Debian 2.6.19 flavor. The classical use of xPC Target by MathWorks as a real-time operating system (OS) is common in academic research (see [25]). A key advantage of xPC Target is its seamless integration between Simulink via Real-Time Workshop which allows for rapid prototyping of navigation and control algorithms for real-time requirements. Real-Time Workshop automatically generates C code from a Simulink model and the corresponding executable file for an xPC Target-based computer. On the

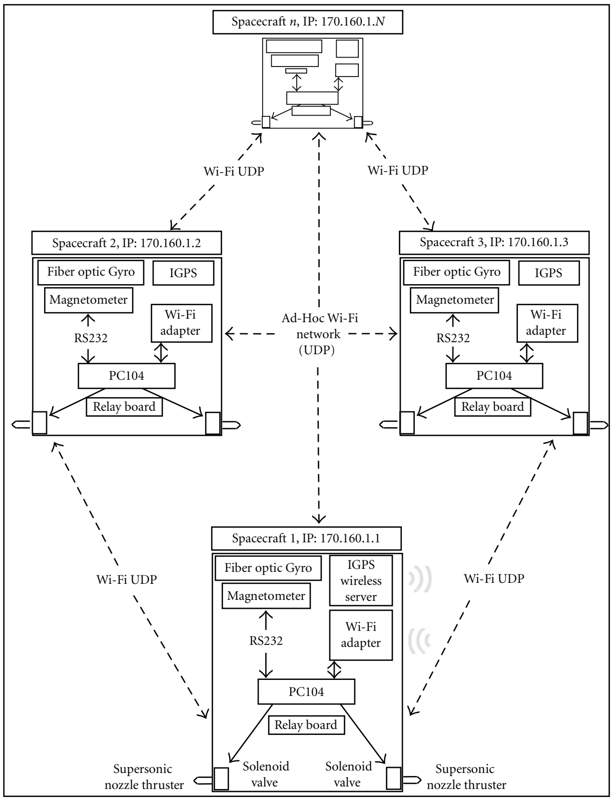


FIGURE 5: Ad-hoc wireless network at the SRL test-bed.

other hand, xPC Target has some disadvantages that include support for a limited number of hardware components and no support for USB or Firewire devices. Furthermore, 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 (see [26, 27]). The RTAI Linux solution is being widely exploited in several engineering areas (see [28–31]). 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 Workshop with the executable file for the onboard computers being created outside MATLAB by simple compilation of the C code.

The details on the ad-hoc wireless network and hardware-software interfaces developed for the Spacecraft Simulators are available in [6].

## 3. S/C Relative Motion Dynamics and Problem Statement

In this section, we provide the dynamics of spacecraft relative motion in the three degrees of freedom case. The dynamics encompasses both the relative translation (two degrees of freedom) and rotation (one degree of freedom). We will refer in the following to a Local Vertical Local Horizontal (LVLH) reference frame (Figure 6) that rotates with the orbital angular velocity $\omega_{\text{LVLH}}$. The origin of LVLH moves on a virtual orbit, conveniently chosen to remain in the vicinity of the maneuvering spacecraft. This point can also be chosen as coincident with one of the spacecraft. The $x$-axis points from the center of the Earth to the center of LVLH, while the $y$-axis is in the orbital plane in the direction of the motion along the orbit and perpendicular to the $x$-axis. The $z$-axis completes the right-handed LVLH frame.

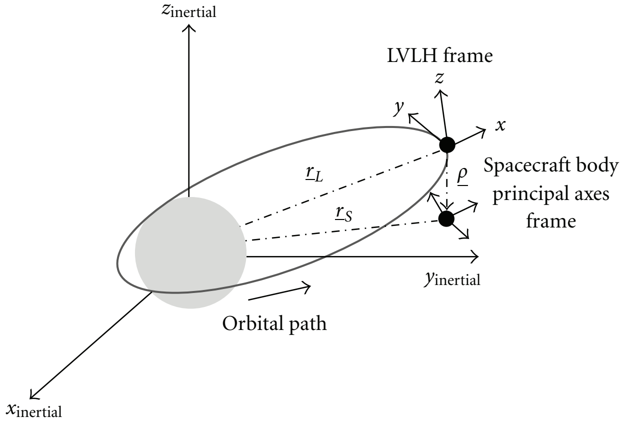


FIGURE 6: Local vertical local horizontal and inertial frames.

The dynamics of such motion can be represented in the compact form as

$$\dot{\mathbf{X}}_{\text{rel}} = f(\mathbf{X}_{\text{rel}}) + B(\mathbf{X}_{\text{rel}})\mathbf{u}. \tag{1}$$

From now on, we will consider the specific application of hardware-in-the-loop testing using the three-degree-of-freedom spacecraft simulators at the Naval Postgraduate School. For the experimental setup, the state vector becomes

$$\begin{aligned} &\mathbf{X}_{\text{rel}} \\ &= \begin{bmatrix} x & y & \theta & \dot{x} & \dot{y} & \omega \end{bmatrix}^T \\ &= \begin{bmatrix} x_T - x_C & y_T - y_C & \theta_T - \theta_C & \dot{x}_T - \dot{x}_C & \dot{y}_T - \dot{y}_C & \omega_{zT} - \omega_{zC} \end{bmatrix}^T \\ &\qquad \mathbf{u} = \begin{bmatrix} u_{x_T} - u_{x_C} & u_{y_T} - u_{y_C} & M_T - M_C \end{bmatrix}^T. \end{aligned} \tag{2}$$

Being $u_{i_j}$, $i = x, y$, $j = C, T$ the control forces components of chaser and target, and $M_j$, $j = C, T$ the control torque of chaser and target about the $z$ axis.

It is common use in the literature to linearize the relative motion dynamics and use the Clohessy and Wiltshire linear equation [32]

$$\begin{aligned} \ddot{x} - 2\omega_{\text{LVLH}}\dot{y} - 3\omega_{\text{LVLH}}^2 x &= \frac{1}{m}\left(u_{x_T} - u_{x_C}\right), \\ \ddot{y} - 2\omega_{\text{LVLH}}\dot{x} &= \frac{1}{m}\left(u_{y_T} - u_{y_C}\right), \end{aligned} \tag{3}$$

with the assumption that the spacecraft have the same mass $m$.

For maneuvers confined in the vicinity of the LVLH origin, elapsing a short time in comparison to the orbital period (3) can be further simplified into a double integrator for both $x$ and $y$. A double integrator dynamics also represents the dynamics of the spacecraft simulators in the laboratory inertial reference frame. For the above-mentioned reasons, (4) will be used for the remaining of the paper;

$$\begin{aligned}\ddot{x} &= \frac{1}{m}\left(u_{x_T} - u_{x_C}\right),\\ \ddot{y} &= \frac{1}{m}\left(u_{y_T} - u_{y_C}\right).\end{aligned} \tag{4}$$

Assuming the spacecraft having the same moment of inertia about the $z$ axis, the attitude dynamics is also represented by a double integrator

$$\ddot{\theta} = \ddot{\theta}_T - \ddot{\theta}_C = \dot{\omega}_{zT} - \dot{\omega}_{zC} = \frac{1}{J_z}(M_T - M_C). \tag{5}$$

The goal of this work is to develop a GNC approach for driving the state $\mathbf{X}_{\text{rel}}$ to perform assembly maneuvers. This requires accurate guidance, especially in the last phases of docking, optimized or suboptimized control, to minimize propellant consumption, and a robust relative navigation scheme. These requirements are addressed in the following sections.

## 4. Relative Navigation: The Augmented State Estimation Method

In this section, the theory for the three-degree-of-freedom augmented state relative navigation is presented. The controls of another vehicle (target) are treated as additional terms in the corresponding state equation, so that the model provides an augmented state vector. The measurements available on each spacecraft are relative positions (from range and line of sight) and relative attitude, and we assume the knowledge of the controls of the chaser, onboard the chaser itself. No relative velocities measurements are available. Observability proof of the vector $[x \;\; y \;\; \theta \;\; \dot{x} \;\; \dot{y} \;\; \omega \;\; u_{x_T} \;\; u_{y_T} \;\; M_{z_T}]^T$ from the measurements $[x \;\; y \;\; \theta]^T$ is provided for the proposed estimation technique, demonstrating how the augmented state technique can reconstruct relative velocities and controls of the target.

In the following developments, the estimated target's controls are considered constants within every sample time interval. It is worth underlying that the control variables $u_{x_T}$, $u_{y_T}$, and $M_{z_T}$ do not represent the actual actuators' control variables onboard the spacecraft simulator. The way $u_{x_T}$, $u_{y_T}$, and $M_{z_T}$ are obtained from the target does not matter from the augmented state filter's point of view. These control variables are estimated in order to add robustness to the filtering technique; they represent the target's maneuvers, but not the specific way they are performed by the target's actuation subsystem.

The same assumption will be used for the observability demonstration. The navigation algorithm is developed using the Kalman filter approach.

The augmented state estimation approach presents numerical efficiency comparable to the standard Kalman Filter applied on the state only. In fact, the augmented state approach introduces a few more variables in the Kalman Filter, without a significant increase on the numerical burden. Additional references with regards to the implementation of Extended Kalman Filters onboard real space missions can be found in [33, 34].

*4.1. Relative Motion Estimation.* The assumption is made of independent estimation and control for the attitude and the position, so that we can proceed as follows. For the relative position, the state vector can be written as (see (2))

$${}^{C_j}_{T_i}\mathbf{X} = {}^{C_j}_{T_i}\begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T. \tag{6}$$

The discrete dynamics for the problem is the following:

$$\begin{aligned}{}^{C_j}_{T_i}\mathbf{X}(k+1) &= {}^{C_j}_{T_i}\mathbf{\Psi}(k)\,{}^{C_j}_{T_i}\mathbf{X}(k)\\ &\quad + {}^{C_j}_{T_i}\mathbf{B}(k)(\mathbf{u}_T(k) - \mathbf{u}_C(k)) + \mathbf{G}\mathbf{W}(k),\\ \mathbf{Z}(k) &= \mathbf{H}{}^{C_j}_{T_i}\mathbf{X}(k) + \mathbf{V}(k).\end{aligned} \tag{7}$$

The expressions of the matrices: $G$, $H$, ${}^{C_j}_{T_i}B(k)$, and $\Psi$ as functions of the measurement update time $T_s$ for this planar case can be written as

$$\begin{aligned}\mathbf{G} &= \mathbf{I}_{4\times 4}, \qquad \mathbf{H} = \begin{pmatrix} \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 2} \end{pmatrix},\\ {}^{C_j}_{T_i}\mathbf{B}(k) &= \int_0^{T_s} \left({}^{C_j}_{T_i}\mathbf{\Phi}(t) \cdot {}^{C_j}_{T_i}\mathbf{B}\right) dt = \begin{pmatrix} \dfrac{T_s^2}{2} & 0 \\ 0 & \dfrac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{pmatrix} / m,\\ {}^{C_j}_{T_i}\mathbf{\Psi}(k) &= {}^{C_j}_{T_i}\mathbf{\Phi}(T_s) = e^{{}^{C_j}_{T_i}\mathbf{F} T_S}\\ &= \mathbf{I}_{4\times 4} + {}^{C_j}_{T_i}\mathbf{F} T_S = \begin{pmatrix} 1 & 0 & T_S & 0 \\ 0 & 1 & 0 & T_S \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},\end{aligned} \tag{8}$$

being

$${}^{C_j}_{T_i}\mathbf{F} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad {}^{C_j}_{T_i}\mathbf{B} = \begin{pmatrix} \mathbf{0}_{2\times 2} \\ \mathbf{I}_{2\times 2} \end{pmatrix} / m. \tag{9}$$

The augmented dynamics adds the estimation of $u_{x_T}$ and $u_{y_T}$, assuming knowledge of the chaser's controls $u_{x_C}$ and $u_{y_C}$. The related state equation matrices can be written as

$$
\begin{aligned}
{}^{C_j}_{T_i}\mathbf{X}_A &= \begin{bmatrix} {}^{C_j}_{T_i}\mathbf{X}(K+1) \\ \mathbf{u}_T(K+1) \end{bmatrix} \\
&= {}^{C_j}_{T_i}\mathbf{\Psi}_A(k)\begin{bmatrix} {}^{C_j}_{T_i}\mathbf{X}(K) \\ \mathbf{u}_T(K) \end{bmatrix} + \mathbf{G}_A\mathbf{W}(k) + {}^{C_j}_{T_i}\mathbf{B}_C(k)\mathbf{u}_C(k), \\
\mathbf{Z}_A(K) &= \mathbf{H}_A\begin{bmatrix} {}^{C_j}_{T_i}\mathbf{X}(K) \\ \mathbf{u}_T(K) \end{bmatrix} + \mathbf{V}(K), \\
\mathbf{G}_{\mathrm{A}} &= \begin{pmatrix} \mathbf{G} \\ \mathbf{0}_{2\times4} \end{pmatrix}, \qquad \mathbf{H}_A = \begin{pmatrix} \mathbf{I}_{2\times2} & \mathbf{0}_{2\times4} \end{pmatrix}, \qquad \mathbf{u}_C = \begin{bmatrix} u_{Cx} \\ u_{Cy} \end{bmatrix}, \\
{}^{C_j}_{T_i}\mathbf{\Psi}_A(k) &= \begin{pmatrix} {}^{C_j}_{T_i}\mathbf{\Psi}(k) & {}^{C_j}_{T_i}\mathbf{B}(k) \\ \mathbf{0}_{2\times4} & \mathbf{I}_{2\times2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & T_s & 0 & \dfrac{T_s^2}{2m} & 0 \\ 0 & 1 & 0 & T_s & 0 & \dfrac{T_s^2}{2m} \\ 0 & 0 & 1 & 0 & \dfrac{T_s}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dfrac{T_s}{m} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \\
{}^{C_j}_{T_i}\mathbf{B}_C(k) &= \begin{pmatrix} -{}^{C_j}_{T_i}\mathbf{B}(k) \\ \mathbf{0}_{2\times2} \end{pmatrix} = \begin{pmatrix} -\dfrac{T_s^2}{2} & 0 \\ 0 & -\dfrac{T_s^2}{2} \\ -T_s & 0 \\ 0 & -T_s \\ 0 & 0 \\ 0 & 0 \end{pmatrix} / m.
\end{aligned}
\tag{10}
$$

*4.2. Relative Attitude Estimation.* The same algorithm is implemented for target's attitude and control torque estimation. Assuming the target rotating only about the vertical axis ($z$-axis), the $i$th Target attitude state vector, with respect to the $j$th Chaser spacecraft, is chosen to be

$$
{}^{C_j}_{T_i}\mathbf{\Theta} = {}^{C_j}_{T_i}\begin{bmatrix} \theta & \dot{\theta} \end{bmatrix}^T. \tag{11}
$$

The discrete dynamics for the attitude problem is

$$
\begin{aligned}
{}^{C_j}_{T_i}\mathbf{\Theta}(k+1) &= {}^{C_j}_{T_i}\mathbf{\Psi}(k)\,{}^{C_j}_{T_i}\mathbf{\Theta}(k) \\
&\quad + {}^{C_j}_{T_i}\mathbf{B}(k)(M_T(k) - M_C(k)) + \mathbf{G}\mathbf{W}(k), \\
\mathbf{Z}(k) &= \mathbf{H}\,{}^{C_j}_{T_i}\mathbf{\Theta}(k) + \mathbf{V}(k),
\end{aligned}
\tag{12}
$$

and the principal dynamics matrices, as function of the time sampling $T_s$, are

$$
\begin{aligned}
&\mathbf{G} = \mathbf{I}_{2\times2}, \qquad \mathbf{H} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \\
&{}^{C_j}_{T_i}\mathbf{B}(k) = \int_0^{T_s} \left( {}^{C_j}_{T_i}\mathbf{\Phi}(t) \cdot {}^{C_j}_{T_i}\mathbf{B} \right) dt = \begin{pmatrix} \dfrac{T_s^2}{2J_z} \\ \dfrac{T_s}{J_z} \end{pmatrix}, \\
&{}^{C_j}_{T_i}\mathbf{\Psi}(k) = {}^{C_j}_{T_i}\mathbf{\Phi}(T_s) = e^{{}^{C_j}_{T_i}\mathbf{F}T_s} = \mathbf{I}_{2\times2} + {}^{C_j}_{T_i}\mathbf{F}T_s = \begin{pmatrix} 1 & T_s \\ 0 & 1 \end{pmatrix},
\end{aligned}
\tag{13}
$$

being

$$
{}^{C_j}_{T_i}\mathbf{F} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \qquad {}^{C_j}_{T_i}\mathbf{B} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} / J_z. \tag{14}
$$

The formulation of the augmentation of the state dynamics adds the estimation of $M_T$, assuming the knowledge of the chaser's control torque $M_C$. The related state equation matrices can be written as

$$
\begin{aligned}
{}^{C_j}_{T_i}\mathbf{\Theta}_A(k) &= \begin{bmatrix} {}^{C_j}_{T_i}\mathbf{\Theta}(k+1) \\ \mathbf{M}_T(k+1) \end{bmatrix} \\
&= {}^{C_j}_{T_i}\mathbf{\Psi}_A(k)\begin{bmatrix} {}^{C_j}_{T_i}\mathbf{\Theta}(k) \\ M_T(k) \end{bmatrix} + {}^{C_j}_{T_i}\mathbf{B}_C(k)M_c + G_A\mathbf{W}(k), \\
\mathbf{Z}_A(k) &= \mathbf{H}_A\begin{bmatrix} {}^{C_j}_{T_i}\mathbf{\Theta}(k) \\ M_T(k) \end{bmatrix} + \mathbf{V}(k), \\
\mathbf{G}_A &= \begin{pmatrix} \mathbf{G} \\ \mathbf{0}_{2\times2} \end{pmatrix}, \qquad \mathbf{H}_A = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \qquad M_c = [M_z], \\
{}^{C_j}_{T_i}\mathbf{B}_C(k) &= \begin{pmatrix} -{}^{C_j}_{T_i}\mathbf{B}(k) \\ 0 \end{pmatrix} = \begin{pmatrix} -\dfrac{T_s^2}{2} \\ -T_s \\ 0 \end{pmatrix} / J_z, \\
{}^{C_j}_{T_i}\mathbf{\Psi}_A(k) &= \begin{pmatrix} {}^{C_j}_{T_i}\mathbf{\Psi}(k) & {}^{C_j}_{T_i}\mathbf{B}(k) \\ \mathbf{0}_{1\times2} & 1 \end{pmatrix} = \begin{pmatrix} 1 & T_s & \dfrac{T_s^2}{2J_z} \\ 0 & 1 & \dfrac{T_s^2}{J_z} \\ 0 & 0 & 1 \end{pmatrix}.
\end{aligned}
\tag{15}
$$

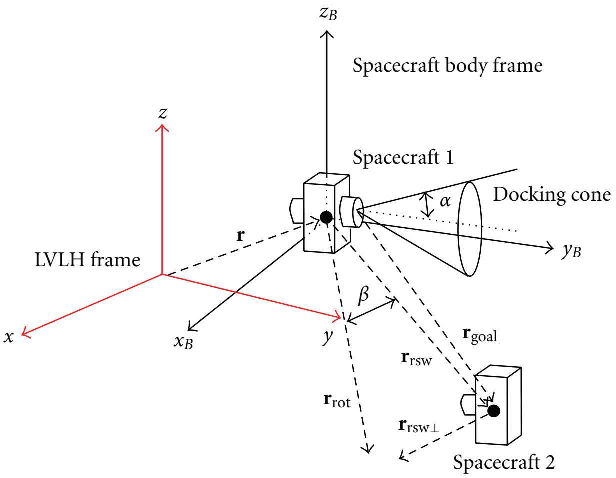


FIGURE 7: Relative vectors used in the alignment and assembly logic. All vectors are in the LVLH $xy$ plane.

*4.3. Observability of the Augmented Dynamics.* For sake of simplicity, considering that the controls are constant in each sample time, we provide, for the planar case, the proof of the observability for the continuous models of the relative dynamics. The observability property holds for the discrete models [35]. The augmented relative motion dynamics can be expressed as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \\ \dot{u}_{x_T} \\ \dot{u}_{y_T} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{m} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{(\mathbf{A})} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ u_{x_T} \\ u_{y_T} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{1}{m} & 0 \\ 0 & -\frac{1}{m} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{(\mathbf{B})} \begin{bmatrix} u_{x_C} \\ u_{x_C} \end{bmatrix}. \tag{16}$$

The measurements are related to the state as follows:

$$Z = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{(\mathbf{C})} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ u_{x_T} \\ u_{y_T} \end{bmatrix}. \tag{17}$$

It is of immediate demonstration that the following observability matrix has full rank:

$$\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \cdot \\ \cdot \\ \mathbf{CA}^5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{m} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{18}$$

Similar developments lead to observability for the relative attitude motion. The dynamics can be expressed as

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{M}_{zT} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{1}{J_z} \\ 0 & 0 & 0 \end{bmatrix}}_{(\mathbf{A})} \begin{bmatrix} \theta \\ \dot{\theta} \\ M_{zT} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ -\frac{1}{J_z} \\ 0 \end{bmatrix}}_{(\mathbf{B})} M_{zC}. \tag{19}$$

The measurements are related to the state as follows:

$$Z = \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_{(\mathbf{C})} \begin{bmatrix} \theta \\ \dot{\theta} \\ M_{zT} \end{bmatrix}. \tag{20}$$

The observability matrix is as following:

$$\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{J_z} \end{bmatrix}, \tag{21}$$

that has full rank.

## 5. Guidance and Control for the Assembly Maneuver

This section describes guidance and control for the autonomous assembly.

*5.1. Guidance.* Figure 7 shows the principal vectors used by the guidance algorithm. It is worth to underline that Figure 7

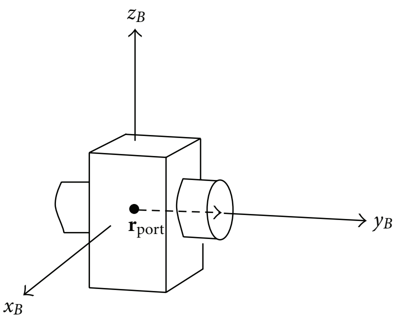


FIGURE 8: Body fixed docking port vector.

only represents one possible configuration and that the docking ports do not have to be aligned with any particular body axis. $\mathbf{r}_{\text{goal}}$ is the vector originating from the center of a docking interface, terminating at the center of mass of the other spacecraft going to dock to it. $\mathbf{r}_{\text{rsw}}$ is the vector originating from a spacecraft's center of mass, terminating at the other spacecraft's center of mass.

The guidance problem is here expressed in terms of desired state vector for each spacecraft, defined dynamically during the maneuver. The state vector error to minimize is

$$\mathbf{x}_{\text{err}} = \left[x - x_{\text{des}} \;\; y - y_{\text{des}} \;\; \theta - \theta_{\text{des}} \;\; \dot{x} - \dot{x}_{\text{des}} \;\; \dot{y} - \dot{y}_{\text{des}} \;\; \omega - \omega_{\text{des}}\right]^T. \tag{22}$$

The subscript "des" indicates a desired relative state vector component. The desired state is dynamically changed throughout the assembly maneuver according to the following two-phase guidance logic.

The center of mass trajectory is unconstrained, free to be optimized, unless in the vicinity of the docking phase. As for the attitude, we reproduce a realistic condition in which the spacecraft has to show one particular side (usually the one with the docking port) towards the current target spacecraft. In other words, the docking port side is commanded to be perpendicular to either the $\mathbf{r}_{\text{rsw}}$ or the $\mathbf{r}_{\text{goal}}$ vector (Figure 7), depending on the phase. Each spacecraft in Figure 7 can be considered either a single agent or an already assembled structure; the following, description applies to both scenarios. In the following the vectors are always intended to be parallel to the $xy$ plane. $r_{\text{dock}}$ is a user defined distance threshold, specifying when the docking phase begins.

(1) $|\mathbf{r}_{\text{rsw}}| > r_{\text{dock}}$, RENDEZVOUS: the spacecraft is at a far away distance from its target docking port. The state vector error is $\mathbf{x}_{\text{err}} = [\mathbf{r}_{\text{goal}_x} \;\; \mathbf{r}_{\text{goal}_y} \;\; \theta - \theta_{\text{des}} \;\; \dot{x} \;\; \dot{y} \;\; \omega]^T$. The desired attitude $\theta_{\text{des}}$ is such to align $\mathbf{r}_{\text{port}}$ to $\mathbf{r}_{\text{rsw}}$ (Figures 7 and 8).

(2) $|\mathbf{r}_{\text{rsw}}| \leq r_{\text{dock}}$, DOCKING APPROACH: the spacecraft is close to its target docking port. The desired state vector to minimize is:

(a) If $\cos^{-1}((\mathbf{r}_{\text{goal}} \cdot \mathbf{r}_{\text{port}})/|\mathbf{r}_{\text{goal}}||\mathbf{r}_{\text{port}}|) < \alpha$, that is, the spacecraft is within the security docking cone, there are two subcases.

SUBCASE 1. The distance between the spacecraft is greater than the chosen impingement stand-off range, then $\mathbf{x}_{\text{err}} = [\mathbf{r}_{\text{goal}_x} \;\; \mathbf{r}_{\text{goal}_y} \;\; \theta - \theta_{\text{des}} \;\; \dot{x} \;\; \dot{y} \;\; \omega]^T$. The desired attitude $\vartheta_{\text{des}}$ is such to align $\mathbf{r}_{\text{port}}$ to $\mathbf{r}_{\text{rsw}}$ (Figures 7 and 8).

SUBCASE 2. The distance between the spacecraft is less than the chosen impingement stand-off range, then any thrusters causing plume impingement on the other spacecraft are shut off, and only used if an emergency brake is needed, in the event of docking occurring at high velocity (above a chosen threshold). For the NPS spacecraft simulators, this will mean shutting off two thrusters, as it will be clear later on. The remaining actuators will compensate for attitude alignment in the last phase of docking and will provide required forces to push the spacecraft together.

(b) If $\cos^{-1}((\mathbf{r}_{\text{goal}} \cdot \mathbf{r}_{\text{port}})/|\mathbf{r}_{\text{goal}}||\mathbf{r}_{\text{port}}|) \geq \alpha$, that is, the spacecraft is outside the security docking cone. In this case, referring to spacecraft 2 of Figure 7, the vehicle maneuvers orbiting around the one hosting its target docking port, moving along the direction perpendicular to the $\mathbf{r}_{\text{rsw}}$ vector, towards the way that is the shortest in order to reach the safety corridor. The amount of commanded rotation at each time step, around the target docking port, is a chosen parameter $\beta = \text{const}$. In terms of state vector error to minimize, defining a reference frame which has as a basis the unit vectors $\hat{\mathbf{r}}_{\text{rsw}_\perp}$, $\hat{\mathbf{r}}_{\text{rsw}}$, the $\mathbf{r}_{\text{rsw}}$ can be rotated of an angle $\beta$ into $\mathbf{r}_{\text{rot}}$ and easily expressed as function of the basis $\mathbf{r}_{\text{rot}} = |\mathbf{r}_{\text{rsw}}|(\cos\beta\hat{\mathbf{r}}_{\text{rsw}} + \sin\beta\hat{\mathbf{r}}_{\text{rsw}\perp})$. The state error to minimize is $\mathbf{x}_{\text{err}} = [\mathbf{r}_{\text{rot}_x} - \mathbf{r}_{\text{goal}_x} \;\; \mathbf{r}_{\text{rot}_y} - \mathbf{r}_{\text{goal}_x} \;\; \theta - \theta_{\text{des}} \;\; \dot{x} \;\; \dot{y} \;\; \omega]^T$. The desired attitude $\theta_{\text{des}}$ is such to show the chosen side to the target docking spacecraft, that is, $\perp \mathbf{r}_{\text{goal}}$ (Figure 7). In simple terms, the satellites circle around each other, in the direction of shortest angular displacement, to allow the docking interfaces to be in the mutual fields of view. Each spacecraft needs to be in the safety corridor one of the other, with the respective docking interfaces' $\mathbf{r}_{\text{port}}$ vectors and $\mathbf{r}_{\text{goal}}$ vectors aligned. The respective $\mathbf{r}_{\text{port}}$ vectors of two satellites will need to be at 180 degrees (plus-minus the tolerance); the same applies, as a consequence, to the $\mathbf{r}_{\text{goal}}$ vectors.

*5.2. LQR Control.* The LQR problem (23) is solved at each time step, with dynamically sized weighting matrices $\mathbf{Q}$ and $\mathbf{R}$, adapting to the current situation, avoiding high control values when the state vector error is relevant, and vice versa. This choice results in a smoother behavior, in terms of requested control actions, with respect to classical fixed gain matrices LQR;

$$J = \int_0^\infty \left(\mathbf{x}_{\text{err}}^T \mathbf{Q} \mathbf{x}_{\text{err}} + \mathbf{u}^T \mathbf{R} \mathbf{u}\right) dt. \tag{23}$$

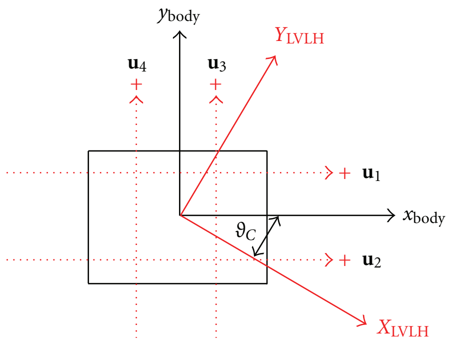


Figure 9: Locations of controls for the planar assembly.

Table 2: Main simulation parameters.

| | | |
|---|---|---|
| Mass of each simulator | 10,5 | kg |
| Inertia of simulator | 0.063 | kg m$^2$ |
| Inertia of the two simulators assembled | 0.18 | kg m$^2$ |
| Single thruster estimated force [36] | 0.16 | $N$ |
| Docking cone semiaperture | 0.75 | degrees |
| Force arms (for torque generation) | 5, 10, 21 | cm |
| Limit distance for switching off the thrusters | 0.7 | m |
| iGPS accuracy | 1 | mm |
| Gyroscope accuracy | 0.003 | deg/sec |
| Thrusters minimum actuation time | $1.5 \cdot 10^{-3}$ | sec |

The cost function in (23) aims to minimize control effort, while reducing the relative state vector between two satellites. The mutual relevance between state vector error and control effort is dictated by the relative values of the weighting matrices **Q** and **R**.

The control vector **u** is chosen as a four-component vector of forces, expressed in the spacecraft body frame (Figure 9). The choice of **u** in the spacecraft body frame, removes the need for thruster mapping [22].

For the phases described in the previous section the weighting matrices for the LQR are chosen as

$$\mathbf{Q} = \begin{bmatrix} \dfrac{1}{\left|\mathbf{r}_{\text{goal}}\right|} \cdot \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \left|\mathbf{r}_{\text{goal}}\right|^3 \cdot V \cdot \mathbf{I}_3 \end{bmatrix}, \tag{24}$$

$$\mathbf{R} = \frac{\left|\mathbf{r}_{\text{goal}}\right|}{a^2} \mathbf{I}_4.$$

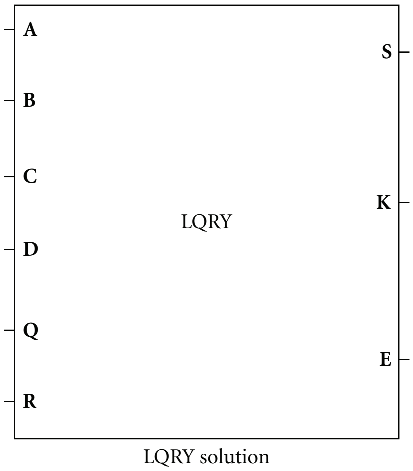


Figure 10: LQR Solver Simulink Block [18]. This routine solves the complete algebraic Riccati equation accepting the input matrices: **A** (dynamics matrix), **B** (control matrix), **C** (state-output mapping matrix), **D** (control-output mapping matrix), and **Q** and **R** weighting matrices. The outputs are the LQR gain matrix, **K**, which is the solution to the associated algebraic Riccati equation, the matrix **S**, and a two-dimensional vector, **E**, whose first element indicates an error when it is greater than zero or a somewhat unreliable result when it is negative. The second element of **E** is the condition number of the **R** matrix.

Each time step solution of the LQR generates a gain matrix $\mathbf{K}_{\text{LQR}}$, used to implement the required suboptimal control vector

$$\mathbf{u}_{\text{LQR}} = -\mathbf{K}_{\text{LQR}} \cdot \mathbf{x}_{\text{err}}. \tag{25}$$

The values of the constants $a = 3.05 \cdot 10^{-2}$ and $V = 0.06$ are chosen as in [23]. In particular, their values are chosen looking at variables with physical meaning, but we do not assign dimensions to them here, being their dimensions the appropriate ones for consistency in the cost function (23). The $a$ constant weighs the terms in the matrix **R** with respect to the maximum translational acceleration achievable on the spacecraft simulator. This value is computed considering two thrusters simultaneously activated on the same side of the vehicle. The thrust values and mass of the simulator can be found in Table 2, and the interested reader can find more details on the thrusters in [36]. The originating idea for scaling the **R** matrix as in (24) is the desire to maintain the controls required by the LQR solution below the maximum hardware-achievable controls. With regard to the constant $V$ in the **Q** matrix, it is set to be the maximum translation speed allowed for the simulator. $V$ weighs the part of the state vector which is related to linear velocities. The choice to introduce the above mentioned parameters does not a priori guarantee controls below the maximum onboard control authority and a limited translational velocity, but the scaling in (24) has been proven very effective in mitigating high requests on the control and undesired fast maneuvers on the testbed. This result was previously found in numerical simulations, and then experimentally verified [23].

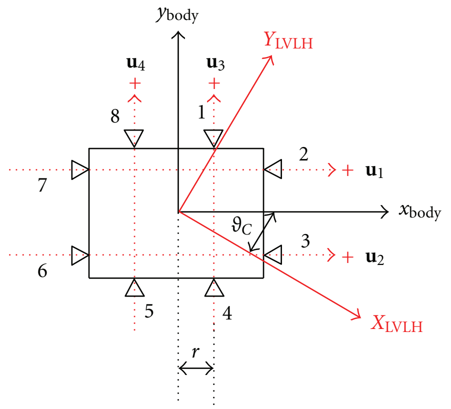


Figure 11: Thruster coupling on the spacecraft simulators.

Figure 10 shows the required inputs to the LQR solver, implemented in Simulink.

The LQR solver employed for developing the proposed approach was downloaded from [18], adapted for automatic generation of code through Real-Time-Workshop for RTAI Linux (it was originally only compatible with Windows Operating Systems), and uploaded again on the MathWorks file exchange website [18].

In specializing the design of Figure 9 to the SRL spacecraft simulators, we treat the eight body fixed thrusters in couples, so that symmetric thrusters are reduced to one control variable, which can be either $u_{\max}$, $-u_{\max}$, 0. Figure 11 shows the thruster couplings: 1–4, 2–7, 3–6, and 5–8. The control vector is $\mathbf{u} = [u_1\ \ u_2\ \ u_3\ \ u_4]^T$. The red arrows along the couples in Figure 11 show the positive directions assumed for the controls. Ultimately, thrusters coupling allows the LQR to solve a reduced problem in which the control vector has four components instead of eight.

Given the choice for the control vector, the control distribution matrix becomes nonlinear, as in (25). Equation (25) also shows the system dynamics matrices, when the expression $\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{BU}$, $\mathbf{Y} = \mathbf{CX} + \mathbf{DU}$ is used. The spacecraft orientation $\theta_C$ is replaced with $\theta$ for simplicity

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \dfrac{\cos(\theta)}{m} & \dfrac{\cos(\theta)}{m} & \dfrac{-\sin(\theta)}{m} & \dfrac{-\sin(\theta)}{m} \\ \dfrac{\sin(\theta)}{m} & \dfrac{\sin(\theta)}{m} & \dfrac{\cos(\theta)}{m} & \dfrac{\cos(\theta)}{m} \\ \dfrac{-r}{J_z} & \dfrac{r}{J_z} & \dfrac{r}{J_z} & \dfrac{-r}{J_z} \end{pmatrix}, \tag{26}$$

$$\mathbf{C} = \mathbf{I}_{6\times 6}, \qquad \mathbf{D} = \mathbf{0}_{6\times 4}.$$

In order to employ the LQR approach, the control distribution matrix is linearized at each time step, in the vicinity of the desired attitude.

$$\mathbf{B}_{\mathbf{LIN}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \dfrac{\cos(\theta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m} & \dfrac{\cos(\vartheta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m} & \dfrac{-\sin(\vartheta_{\text{des}}) + (\cos(\theta_{\text{des}}))(\theta - \theta_{\text{des}})}{m} & \dfrac{-\sin(\theta_{\text{des}}) + (\cos(\theta_{\text{des}}))(\theta - \theta_{\text{des}})}{m} \\ \dfrac{\sin(\theta_{\text{des}}) + (\cos(\theta_{\text{des}}))(\theta - \theta_{\text{des}})}{m} & \dfrac{\sin(\theta_{\text{des}}) + (\cos(\theta_{\text{des}}))(\theta - \theta_{\text{des}})}{m} & \dfrac{\cos(\vartheta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m} & \dfrac{\cos(\theta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m} \\ \dfrac{-r}{J_z} & \dfrac{r}{J_z} & \dfrac{r}{J_z} & \dfrac{-r}{J_z} \end{pmatrix} \tag{27}$$

By inserting the matrices defined in (26) and (27) and the weighting matrices described in (24) into the LQR solver of Figure 10, the optimal four components vector of forces is obtained, at each time step during the maneuver. The obtained control vector will be a continuous signal. In order to drive the on/off thrusters from the continuous signal Pulse Width Modulation is used; the PWM collects commanded controls over 10 sample times before actuating. Furthermore, a Schmitt Trigger is implemented, to filter out low commanded controls and reduce the amount of chattering.

## 6. Navigation and Control of the Assembled Structure

Once the S/Cs are assembled, the mass and inertia properties along with the thrusters configuration change. Figure 12 shows an example, applied to the SRL testbed, in which thrusters six and seven of both spacecraft cannot be used anymore. The assembled new spacecraft has doubled mass, different moment of inertia and four more thrusters, differently allocated with respect to the single spacecraft. In assembled configuration, one of the robots acts as master

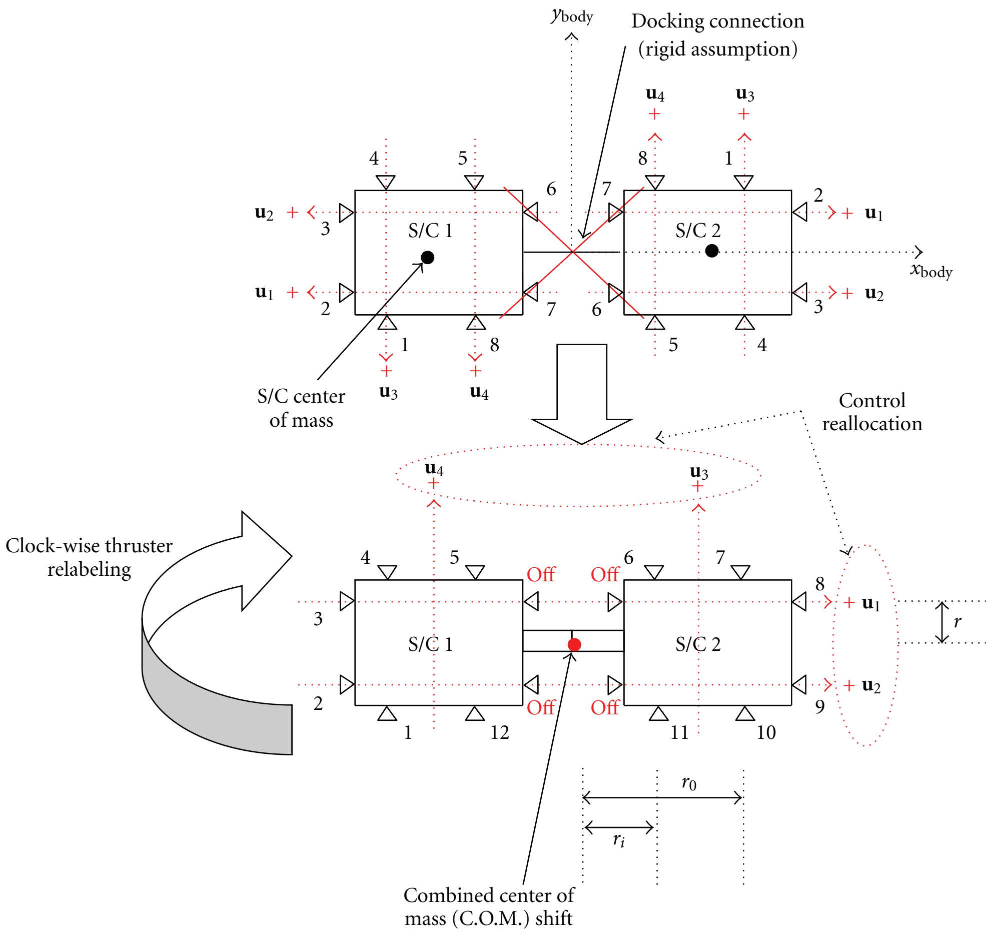


Figure 12: Assembled configurations with reallocated thruster coupling and COM shift.

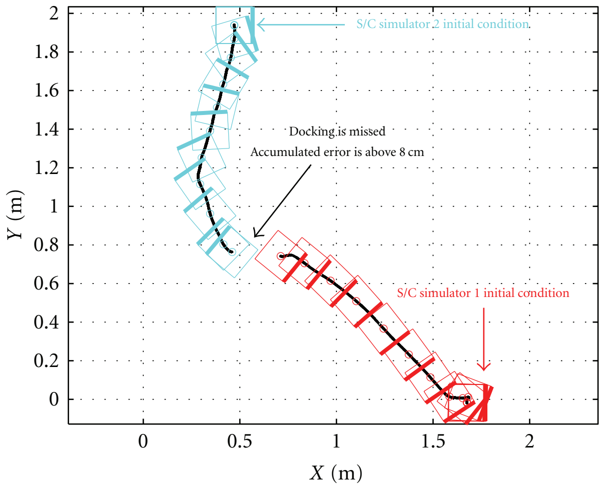


Figure 13: Experimental Result: bird's eye view for two spacecraft simulator failed assembly maneuver. The relative navigation is performed via classical Kalman Filtering, no Augmented State Estimation. The bolded lines are employed to help visualize the simulator's orientation.

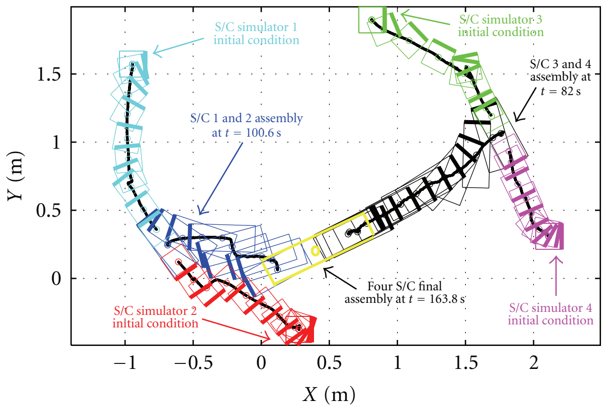


Figure 14: Experimental Result: bird's eye view for four spacecraft simulator assembly maneuver. The relative navigation is performed via augmented state estimation. The bolded lines are employed to help visualize the simulator's orientation.

and performs both navigation and control of the new bigger robot. In order to keep using the same logic employed for controlling a single simulator, the twelve thrusters of the new assembled spacecraft are associated according to the following sets:

(1) $u_1$ is generated by firing either thruster 8 ($u_1 < 0$) or 3 ($u_1 > 0$),

(2) $u_2$ is generated by firing either thruster 9 ($u_2 < 0$) or 2 ($u_2 > 0$),

(3) $u_3$ is generated by firing either thrusters 6 and 7 synchronously ($u_3 < 0$) or 11 and 10 synchronously ($u_3 > 0$),

(4) $u_4$ is generated by firing either thrusters 4 and 5 synchronously ($u_4 < 0$) or 1 and 12 synchronously ($u_{\,4} > 0$).

The input matrices to the LQR solver will be changed once an additional portion of the structure is connected. Also, the new control vector will have maximum and minimum values reduced, due to the increase of mass. For instance, the case represented in Figure 12 leads to the new matrices

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \dfrac{\cos(\theta)}{m_{\text{comb}}} & \dfrac{\cos(\theta)}{m_{\text{comb}}} & \dfrac{-\sin(\theta)}{m_{\text{comb}}} & \dfrac{-\sin(\theta)}{m_{\text{comb}}} \\ \dfrac{\sin(\theta)}{m_{\text{comb}}} & \dfrac{\sin(\theta)}{m_{\text{comb}}} & \dfrac{\cos(\theta)}{m_{\text{comb}}} & \dfrac{\cos(\theta)}{m_{\text{comb}}} \\ \dfrac{-r}{J_{z\_\text{comb}}} & \dfrac{r}{J_{z\_\text{comb}}} & \dfrac{r_o + r_i}{J_{z\_\text{comb}}} & \dfrac{-(r_o + r_i)}{J_{z\_\text{comb}}} \end{pmatrix}, \qquad (28)$$

$$\mathbf{C} = \mathbf{I}_{6\times 6}, \qquad \mathbf{D} = \mathbf{0}_{6\times 4},$$

where $J_{z\_\text{comb}}$ is the inertia of the assembled system about the vertical axis and $m_{\text{comb}}$ is the new mass. Linearization of the new control distribution matrix leads to

$$\mathbf{B}_{\mathbf{LIN}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \dfrac{\cos(\theta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \dfrac{\cos(\theta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \dfrac{-\sin(\theta_{\text{des}}) + \cos(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \dfrac{-\sin(\theta_{\text{des}}) + \cos(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} \\ \dfrac{\sin(\theta_{\text{des}}) + \cos(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \dfrac{\sin(\theta_{\text{des}}) + \cos(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \dfrac{\cos(\theta_{\text{des}}) - \sin(\vartheta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \dfrac{\cos(\theta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} \\ \dfrac{-r}{J_{z\_\text{comb}}} & \dfrac{r}{J_{z\_\text{comb}}} & \dfrac{r_o + r_i}{J_{z\_\text{comb}}} & \dfrac{-(r_o + r_i)}{J_{z\_\text{comb}}} \end{pmatrix} \qquad (29)$$

The thrusters remained available after docking will be commanded by either spacecraft one or two, thanks to the real-time wireless link (see [6]). Navigation for the assembled structure is performed onboard the robot acting as the

Table 3: Position, augmented state filter parameters.

| | | |
|---|---|---|
| $\overline{\mathbf{Q}}$ | $(1 \times 10^{-6}) \cdot \mathbf{I}_{6\times6}$ | Process covariance matrix, for the adapted augmented state filter |
| $\overline{\mathbf{R}}$ | $(1 \times 10^{-4}) \cdot \mathbf{I}_{2\times2}$ | Measurements covariance matrix, for the adapted augmented state filter |
| $\mathbf{P}_o$ | $R_k \cdot \mathbf{I}_{6\times6}$ | Initial covariance matrix, for the adapted augmented state filter |
| $T_s$ | 0.02 sec | Simulation sampling time |

Table 4: Attitude, augmented state filter parameters.

| | | |
|---|---|---|
| $\overline{\mathbf{Q}}$ | $(1 \times 10^{-12}) \cdot \mathbf{I}_{3\times3}$ | Process covariance matrix, for the adapted augmented state filter |
| $\overline{\mathbf{R}}$ | $(1 \times 10^{-2})$ | Measurements covariance matrix, for the adapted augmented state filter |
| $\mathbf{P_o}$ | $R_k \cdot \mathbf{I}_{3\times3}$ | Initial covariance matrix, for the adapted augmented state filter |
| $T_s$ | 0.02 sec | Simulation sampling time |

master. For the two-robot configuration in Figure 12, it follows the rigid body equations, being the attitude and attitude rate of the new spacecraft the same of the master, and its center of mass position and velocity deducted by those of the master (see also Figure 12).

## 7. Experimental Results: Four Spacecraft Simulators Assembly

In this section, assembly maneuvers are employed to experimentally test the suggested guidance, navigation, and control schemes. For our experiment, we do not implement any collision avoidance algorithm, which has been, however, previously successfully tested [23]. At the time of writing this paper, the simulators do not have hardware dedicated to relative measurements. Relative measurements are assumed to be the range, line of sight, and relative attitude. This information is obtained via software, by having the robots exchanging data over the ad-hoc wireless channel [6]. This feature has the benefit of flexibility in imposing the desired frequency of measurement update, by simple modification of the software. Furthermore, for the following experiments, we do not assume any particular noise or bias characteristics for the measurements, that is, the filter does not have that information. Noise is present, and it comes from the wireless communication. This assumption does not conflict with the previously stated contribution of our work in designing an estimation technique more robust than standard Kalman Filtering in the presence of low frequency updates, as demonstrated in the following.

Two experimental runs are presented. The first one demonstrates the unsuccessful relative navigation when classical Kalman Filtering is employed, considering the other S/C's maneuvers as a random process. Only two simulators are involved. The second experiment involves the four vehicles, showing how augmented state estimation can handle low measurement updates and unpredictable interruptions of updates, and still perform correct relative navigation, driving the mission to success. In particular, we are here imposing, via the wireless network, an update of 2 seconds. Once the two couples of robots are docked, each of the assembled structure is considered to be a new vehicle with new mass and geometry. For this reason, the augmented state estimator is reinitialized for the new structure with different mass and inertia as in Table 2. For this part of the experimentation, the software is running only onboard the master S/C, that is, one pre-chosen unit for each couple.

The time step, or simulation sampling time, was chosen to be: (1) higher than the thrusters minimum actuation time (Tables 3 and 4), (2) in compatibility with CPU computational power, and (3) so to maintain the dynamics within the linearity range. The choice was also justified by previous experience with the employed hardware and by prior computer numerical simulations. In fact, the experimental activities at the Spacecraft Robotics Laboratory are always anticipated by high fidelity numerical simulations of the test-bed dynamics in Simulink, by visualizing on the computer how the experiment will develop. This prototyping approach reduces the time-to-market and trouble-shooting costs of newly developed GNC methodologies, by significantly cutting down the need for intermediate hardware prototypes and the number of experimental tests.

*7.1. The Classical Kalman Filter Technique.* Figure 13 is the bird's eye view of the experiment, demonstrating the unfeasibility of classical Kalman filter for spacecraft relative navigation, when relative measurements updates occur at low frequency. Two spacecraft simulators start maneuvering, with the goal of docking, from a short distance. The sides opposite to the bolded lines are the designated docking sides. After approximately 1 minute of maneuver, the accumulated error in relative state vector (position) exceeds the tolerance of the docking interfaces (Figure 2), driving the vehicles into a failed docking maneuver. A video of the experiment can be found online at [37].

*7.2. The Augmented State Estimation Technique.* Figure 14 is the bird's eye view of the experiment, demonstrating the feasibility of the augmented state estimation for spacecraft relative navigation. The main data for the filters are presented in Tables 3 and 4. Four spacecraft simulators start maneuvering, with the goal of assembling into a line-shaped structure, from short distances. The sides opposite to the bolded lines are the designated docking sides. After less than 3 minutes of maneuver, the four vehicles successfully complete the given mission. The rectangular black and blue vehicles represent two spacecraft simulators docked and maneuvering as a single bigger unit. A video of the experiment can be found online at [38].

Once the simulators are assembled in couples, they maneuver as a single bigger unit. In particular, the augmented state estimation is reinitialized in order to switch to a new target vehicle in terms of relative navigation. In Figure 14, for the left couple, the cyan-represented unit acts as master of the new assembled cyan-red spacecraft. Likewise, for the right couple, the green vehicle is the master in the green-magenta assembly.

## 8. Conclusion

In this work, we are suggesting a complete solution for guidance, navigation, and control of planar multiple spacecraft assembly maneuvers. Guidance is performed by dynamically defining a desired state vector, so that the spacecraft can prepare for docking and correctly connect. The control is based on a real time LQR approach. As for the relative navigation, augmented state estimation is proposed, allowing for correct awareness of the other spacecraft configuration, even in the event of low frequency measurements update. The framework adapts itself to the evolving spacecraft, by switching among different values of mass properties and sensors and actuators configuration, when a new unit assembles to the aggregate.

Theoretical developments are presented for the three-degree-of-freedom case, considering a planar motion for the relative position and a single axis of rotation.

The experimental validation of the proposed methodology is presented, via floating spacecraft simulators, using an assembly maneuver as baseline. Experiments show how the augmented state estimation can cope with low frequency measurement updates, correctly performing the relative navigation, driving the mission to success. On the other hand, Classical Kalman Estimation, is not accurate for close distances with low frequency measurement updates as demonstrated in the three-degree-of-freedom experimental section. The dynamic guidance and control demonstrate real-time feasibility and the capability of performing autonomous assembly.

## Nomenclature

*Acronyms*

| | |
|---|---|
| COTS: | Commercial off the shelve |
| DoF: | Degree of freedom |
| GNC: | Guidance, navigation, and control |
| IE: | Input estimation |
| LQR: | Linear quadratic regulator |
| LVLH: | Local vertical local horizontal reference frame centered on the chaser spacecraft |
| NPS: | Naval postgraduate school |
| PWM: | Pulse width modulation |
| SRL: | Spacecraft robotics laboratory. |

*Variables and Symbols*

| | |
|---|---|
| $\alpha$: | Docking safety cone semiaperture |
| $\beta$: | Commanded orbiting angle around target docking port in docking phase |
| $\Delta t$: | Control system sample time |
| $\omega_{\text{LVLH}}$: | Chaser orbital angular velocity |
| $\theta$: | Target attitude angle in chaser S/C body frame |
| $\dot{\theta}$: | Target angular velocity in chaser S/C body frame |
| ${}^{C_j}_{T_i}\mathbf{\Phi}$: | Transition Matrix of the dynamics of the $i$th target S/C with respect to $j$th chaser S/C |
| ${}^{C_j}_{T_i}\mathbf{\Phi}_A$: | Augmented transition matrix of the dynamics of the $i$th Target S/C with respect to $j$th chaser S/C |
| ${}^{C_j}_{T_i}\mathbf{\Theta}$: | Attitude state vector of $i$th target S/C with respect to $j$th chaser S/C |
| ${}^{C_j}_{T_i}\mathbf{\Theta}_A$: | Augmented attitude state vector of $i$th Target S/C with respect to $j$th chaser S/C |
| $a$: | Scaling factor in $\mathbf{R}$ LQR weighting matrix |
| $k$: | Discrete time index |
| $m, m_{\text{comb}}$: | Single & combined mass of the spacecraft simulator |
| $r$: | Torque Arm: Thruster-center of mass arm |
| $\mathbf{r}_{\text{rsw}}$: | Spacecraft-to-spacecraft vector |
| $\mathbf{r}_{\text{goal}}$: | Docking port-to-corresponding docking spacecraft vector |
| $\mathbf{r}_{\text{port}}$: | Center of mass to docking port vector |
| $r_{\text{dock}}$: | Spacecraft-to-spacecraft transition distance between far away phase and docking phase |
| $t$: | Time |
| $\mathbf{u}_C$: | Chaser's control vector |
| $\mathbf{u}_T$: | Target's control vector |
| $\mathbf{u}$: | Relative control vector |
| $\mathbf{u}_{\text{LQR}}$: | Optimal control vector |
| $u_{\max} > 0$: | Single engine maximum thrust |
| $u_{\text{thr}} > 0$: | Threshold value for required thrust Before using PWM |
| $\mathbf{A}$: | State matrix |
| $\mathbf{B}$: | Control distribution matrix |
| $\mathbf{C}$: | State-output mapping matrix |
| $\mathbf{D}$: | Control-output mapping matrix |
| ${}^{C_j}_{T_i}B$: | Control matrix referred of $i$th target S/C with respect to $j$th chaser S/C |

| | |
|---|---|
| ${}^{C_j}_{T_i}\mathbf{B}(k)$: | Discretized control matrix referred of $i$th target S/C with respect to $j$th chaser S/C |
| ${}^{C_j}_{T_i}\mathbf{B}_A(k)$: | Discretized augmented control matrix referred of $i$th target S/C with respect to $j$th chaser S/C |
| ${}^{C_j}_{T_i}\mathbf{F}$: | State matrix of $i$th target S/C with respect to $j$th chaser S/C |
| ${}^{C_j}_{T_i}\mathbf{F}_A$: | Augmented state matrix of $i$th target S/C with respect to $j$th chaser S/C |
| $\mathbf{G}$: | Input noise matrix |
| $\mathbf{H}$: | Measurement matrix |
| $\mathbf{G}_A$: | Augmented input noise matrix |
| $\mathbf{H}_A$: | Augmented measurement matrix |
| $J$: | Cost function |
| $J_z, J_{z\_\text{comb}}$: | Inertia of a single and combined Spacecraft simulator about the vertical axis |
| $\mathbf{K}_{\text{LQR}}$: | LQR resulting gain matrix |
| $M_C$: | Chaser spacecraft torque |
| $M_T$: | Target spacecraft torque |
| $\mathbf{P_0}$: | Initial state error covariance matrix |
| $\mathbf{Q}$: | LQR state error weighting matrix |
| $\overline{\mathbf{Q}}_\mathbf{k}$: | Process noise covariance matrix |
| $\mathbf{R}$: | LQR control effort weighting matrix |
| $R_k$: | Measurement noise covariance |
| $T$: | Maneuver total time |
| $T_s$: | Sampling time |
| $\mathbf{V}$: | Measurement noise vector, assumed to be gaussian white zero mean with covariance $R_k$ |
| $V$: | Scaling factor in $\mathbf{Q}$ LQR weighting matrix |
| $\mathbf{W}$: | Input noise vector, assumed to be Gaussian white zero mean with covariance $Q_k$ |
| ${}^{C_j}_{T_i}\mathbf{X}$: | State vector of $i$th target S/C with respect to $j$th chaser S/C, |
| ${}^{C_j}_{T_i}\mathbf{X}_A$: | Augmented state vector of $i$th target S/C with respect to $j$th chaser S/C |
| $\mathbf{X}_{\text{rel}}$: | Relative state vector between two S/C |
| ${}^{C_j}_{T_i}\mathbf{S}$: | Complete $i$th target state vector ref to $j$th chaser S/C |
| $\mathbf{Z}$: | Measurement vector |
| $\mathbf{Z}_A$: | Augmented measurement vector |
| $\mathbf{I}_{p\times p}$: | Identity matrix, |
| $\mathbf{0}_{q\times s}$: | Zeros matrix |
| $[x \quad y \quad z]^T$: | Target Cartesian coordinates in chaser S/C body frame |
| $[\dot{x} \quad \dot{y} \quad \dot{z}]^T$: | Target linear velocities in chaser S/C body frame. |

## Acknowledgments

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] S. A. Frost and M. J. Balas, "Adaptive key component controllers for evolving systems," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, USA, August 2008.

[2] M. J. Balas and S. A. Frost, "Evolving systems: a theoretical foundation," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colo, USA, August 2006.

[3] T. S. VanZwieten and M. J. Balas, "Evolving systems: inheriting stability with decentralized adaptive control," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, SC, USA, August 2007.

[4] S. A. Frost and M. J. Balas, "Stabilizing controllers for evolving systems with application to flexible space structures," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, SC, USA, August 2007.

[5] S. Dong, K. Allen, P. Bauer et al., "Self-assembling wireless autonomously reconfigurable module design concept," *Acta Astronautica*, vol. 62, no. 2-3, pp. 246–256, 2008.

[6] R. Bevilacqua, J. S. Hall, J. Homing, and M. Romano, "Ad hoc wireless networking and shared computation for autonomous multirobot systems," *Journal of Aerospace Computing, Information and Communication*, vol. 6, no. 5, pp. 328–353, 2009.

[7] R. Bevilacqua and M. Romano, "Fuel-optimal spacecraft rendezvous with hybrid on-off continuous and impulsive thrust," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 1175–1178, 2007.

[8] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 6, no. 4, pp. 473–483, 1970.

[9] Y. T. Chan and F. Couture, "Manoeuvre detection and track correction by input estimation," *IEE Proceedings F*, vol. 140, no. 1, pp. 21–28, 1993.

[10] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, New York, NY, USA, 2001.

[11] Y. Bar-shalom and K. Birmiwal, "Variable dimension filtering for maneuvering target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 18, no. 5, pp. 621–629, 1982.

[12] P. L. Bogler, "Tracking a maneuvering target using input estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 23, no. 3, pp. 298–310, 1987.

[13] H. Khaloozadeh and A. Karsaz, "A new state augmentation for maneuvering targets detection," in *Proceedings of International Conference on Signal Processing and Communications (SPCOM '04)*, pp. 65–69, December 2004.

[14] M. H. Bahari, A. Karsaz, and M. B. Naghibi-S, "Intelligent error covariance matrix resetting for maneuver target tracking," *Journal of Applied Sciences*, vol. 8, no. 12, pp. 2279–2285, 2008.

[15] M. H. Bahari and N. Pariz, "High maneuvering target tracking using an input estimation technique associated with fuzzy

forgetting factor," *Scientific Research and Essays*, vol. 4, no. 10, pp. 936–944, 2009.

[16] V. Pellegrini, R. Bevilacqua, M. Romano, and F. Curti, "Spacecraft proximity navigation and autonomous assembly based on augmented state estimation: analysis and experiments," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, August 2010.

[17] S. B. McCamish, M. Romano, S. Nolet, C. M. Edwards, and D. W. Miller, "Flight testing of multiple-spacecraft control on SPHERES during close-proximity operations," *Journal of Spacecraft and Rockets*, vol. 46, no. 6, pp. 1202–1213, 2009.

[18] http://mathworks.com/matlabcentral/fileexchange/2651.

[19] P. Zarchan and H. Musoff, *Fundamental of Kalman Filtering: A Practical Approach*, AIAA, Reston, Va, USA, 2009.

[20] M. Romano, D. A. Friedman, and T. J. Shay, "Laboratory experimentation of autonomous spacecraft approach and docking to a collaborative target," *Journal of Spacecraft and Rockets*, vol. 44, no. 1, pp. 164–173, 2007.

[21] J. L. Schwartz, M. A. Peck, and C. D. Hall, "Historical review of air-bearing spacecraft simulators," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 4, pp. 513–522, 2003.

[22] F. Curti, M. Romano, and R. Bevilacqua, "Lyapunov-based thrusters' selection for spacecraft control: analysis and experimentation," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1143–1160, 2010.

[23] R. Bevilacqua, T. Lehmann, and M. Romano, "Development and experimentation of LQR/APF guidance and control for autonomous proximity maneuvers of multiple spacecraft," *Acta Astronautica*, vol. 68, no. 7-8, pp. 1260–1275, 2011.

[24] https://www.rtai.org/RTAILAB/RTAI-Lab-tutorial.pdf.

[25] http://www.mathworks.com/products/xpctarget/.

[26] L. Dozio and P. Mantegazza, "Real time distributed control system using RTAI," in *Proceedings of the 6th IEE International Symposium on Object-Oriented Real-Time Distributed Computing*, May 2003.

[27] G. Quaranta and P. Mantegazza, "Using MATLAB-simulink RTW to build real time control applications in user space with RTAI-LXRT," in *Realtime Linux Workshop*, Milan, Italy, 2001.

[28] S. Ricci and A. Scotti, "Aeroelastic testing on a three surface airplane," in *Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, pp. 7131–7147, Newport, RI, USA, May 2006.

[29] L. Dozio, A. Toso, W. Corbetta, E. Vigoni, and G. Ghiringhelli, "Comparison of feedback and feedforward strategies for active structural—acoustic control of broadband sound transmission into a cavity," in *Proceedings of the 13th AIAA/CEAS Aeroacoustics Conference, 28th AIAA Aeroacoustics Conference*, 2007.

[30] M. G. Perhinschi, M. R. Napolitano, G. Campa, B. Seanor, S. Gururajan, and G. Yu, "Design and flight testing of intelligent flight control laws for the WVU YF-22 model aircraft," in *Proceedings of AIAA Guidance, Navigation, and Control Conference*, pp. 5925–5936, San Francisco, Calif, USA, August 2005.

[31] Y. Zheng, B. J. Martin, and N. Villaume, "VERSE—virtual equivalent real-time simulation environment," in *Proceedings of AIAA Modeling and Simulation Technologies Conference*, pp. 368–376, San Francisco, Calif, USA, August 2005.

[32] W. H. Clohessy and R. S. Wiltshire, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.

[33] S. D'Amico, E. Gill, M. Garcia, O. Montenbruck, and E. Gill, "GPS-based real-time navigation for the PRISMA formation flying mission," in *Proceedings of the 3rd ESA Workshop on Satellite Navigation User Equipment Technologies (NAVITEC '06)*, Noordwijk, The Netherlands, December 2006.

[34] S. D'Amico, J. S. Ardaens, and O. Montenbruck, "Navigation of formation flying spacecraft using GPS: the PRISMA technology demonstration," in *Proceedings of the 22nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION-GNSS '09)*, pp. 2250–2264, Savannah, Ga, USA, September 2009.

[35] K. Ogata, *Discrete-Time Control Systems*, Prentice-Hall, New York, NY, USA, 2nd edition, 1995.

[36] C. Lugini and M. Romano, "A ballistic-pendulum test stand to characterize small cold-gas thruster nozzles," *Acta Astronautica*, vol. 64, no. 5-6, pp. 615–625, 2009.

[37] http://aa.nps.edu/~mromano/Movies4Web/GNC10_Video1.flv.

[38] http://aa.nps.edu/~mromano/Movies4Web/GNC10_Video2.flv.