View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by DSpace@MIT

On-Orbit Assembly of Flexible Space Structures with SWARM

Jacob G. Katz[1], Swati Mohan[2], and David W. Miller[3]
Massachusetts Institute of Technology, Cambridge, MA, 02139

On-orbit assembly is an enabling technology for many space applications. However, current methods of human assisted assembly are high in cost and risk to the crew, motivating a desire to automate the on-orbit assembly process using robotic technology. Construction of large space structures will likely involve the manipulation of flexible elements such as trusses or solar panels, and automation for assembly of flexible structures has significant challenges, particularly in control systems. This paper presents results of ground-based experiments on the assembly of a flexible space structures using the hardware developed under the Self-Assembling Wireless Autonomous Reconfigurable Modules (SWARM) program. Results are shown for a series of incremental tests that demonstrate control of a flexible structure, docking, and reconfiguration after docking. These results demonstrate the feasibility of the assembly of flexible structures using this methodology.

Nomenclature

$\mathbf{a}$	=	adaptive parameter vector
β	=	pixel size
$\mathbf{C}$	=	Coriolis matrix
$\mathbf{D}_{\mathrm{act}}$	=	force direction of thrusters
$\mathbf{D}_{\mathrm{T}}$	=	torque directions of thrusters
$\boldsymbol{\delta}$	=	vector of link deflections
δ_f	=	simplified beam deflection
f	=	focal length
k	=	spring constant of simplified beam
$\mathbf{K}_f$	=	diagonal matrix of spring constants
l_i	=	length of link i
$\mathbf{M}$	=	inertia matrix
$\mathbf{q}$	=	generalized coordinate vector
$\mathbf{r}_{\mathrm{act}}$	=	thruster locations
$\mathbf{r}_i$	=	distance to center of mass i
$\mathbf{s}$	=	weighted tracking error
$\boldsymbol{\tau}$	=	actuator input vector
x_0, y_0	=	camera center of projection
$\bar{x}, \bar{y}$	=	LED centroid position
$\boldsymbol{\Xi}$	=	control allocation matrix
$\mathbf{Y}$	=	dynamic regressor
y_t	=	beam tip deflection

I. Introduction

Spacecraft tend to be developed as point-designs, meant to optimize their particular tasks and missions. Launch costs are so high, and launch opportunities so rare, that it is only natural for program managers to try to include

[1] PhD Candidate, Dept. of Aeronautics and Astronautics, 77 Massachusetts Avenue, AIAA Student Member
[2] Post-Doctoral Student, Dept. of Aeronautics and Astronautics, 77 Massachusetts Avenue, AIAA Member.
[3] Professor, Dept. of Aeronautics and Astronautics, 77 Massachusetts Avenue, AIAA Senior Member

as much functionality as possible into each satellite. For this reason, except for satellites that are part of constellations (e.g., GPS, TDRSS, etc.), most spacecraft are one-of-a-kind, or few-of-a-kind creations. While components or subsystems may be re-used from one spacecraft to the next, there is usually a very high degree of customization, which in turn leads to inevitable increases in testing and verification costs, along with unavoidable decreases in reliability.

The Self-Assembling Wireless Autonomous Modules (SWARM) program uses modular design as a motivation to research techniques for assembly, reconfiguration and upgrade of systems that are unachievable using current technology. Instead of designing unique spacecraft for each different mission, SWARM instead consists of a finite set of modules. Each module could perform the role of a single mission subsystem or function. Modules would contain all components necessary for their particular function, as well as standardized mechanical and power interfaces required to properly interface with other modules. Depending upon the needs of a particular mission and payload, combinations of modules could be assembled to achieve a desired geometry and functionality.

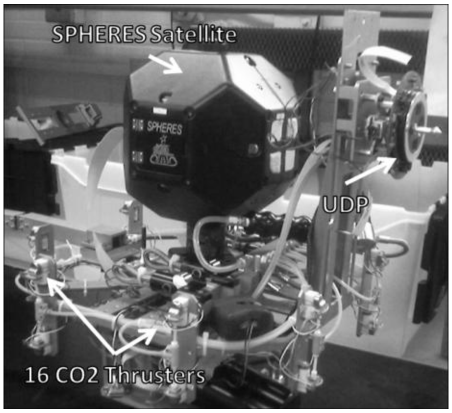


Figure 1. A SPHERES satellite mounted to the SWARM propulsion module.

On-orbit assembly is an exciting path to achieving the broad configuration possibilities offered by a modular assembly system and sidestep some of the volume and mass constraints imposed by launching a monolithic satellite on a single launch vehicle. For instance, for light-gathering satellites such as space telescopes or orbiting solar arrays, on-orbit assembly provides the opportunity of vastly increasing the surface area available from a single launch vehicle. Instead of relying on complicated deployment mechanisms, a large array could be assembled by a small helper robot that joins pieces of the structure together. The structures could be assembled over the period of several weeks or months and over multiple launches, supervised by a ground team instead of a costly in-flight human construction crew.

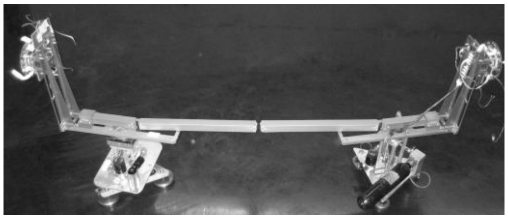


Figure 2. The SWARM beam mounted to two air carriages.

Due to the use of lightweight materials, construction of large space structures will likely involve the manipulation of flexible elements such as trusses or solar panels. The objective of an assembly robot will be to position and join these components without inducing large vibrations that could disrupt assembly or damage the components. Therefore, it will be critical to consider flexibility in the development of a control and estimation system for autonomous on-orbit assembly.

Two phases of hardware development were performed in the SWARM program. Phase I investigated the construction of rigid modules to form a sparse-aperture space telescope assembly.[1,2] A SPHERES satellite mounted on an air carriage with two docking ports was used to maneuver and attach components of the structure. The Phase II research presented here made major hardware improvements to the Phase I components and considered the construction of a structure with flexible elements.

II. SWARM Hardware

The following sections summarize the Phase II hardware components used in this study.

A. Propulsion Module

In Phase I, SPHERES thrusters alone were inadequate for overcoming friction while moving an assembled structure on the flat floor. To alleviate this problem, a propulsion module was designed that augments the SPHERES thrusters with 16 additional thrusters located in pairs at each corner of a square mounting plate. Together, the propulsion module and SPHERES satellite shown in Fig. 1 simulate the propulsion system of a small assembly tug.

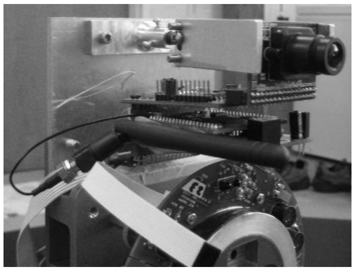


Figure 3. A Surveyor Corp. SRV-1 digital camera mounted above the UDP is used to observe the motion of the flexible beam.

B. Flexible Beam

A key objective of the SWARM project is the demonstration of on-orbit assembly in the presence of flexibility, but simulating large flexible structures in a gravitational field is challenging because they are prone to sagging or buckling. To address this issue, the beam is composed of several rigid links connected by flexible joints, shown in Fig. 2. The structure is a physical approximation to a homogeneous beam similar to a lumped approximation model in flexible dynamics simulation.

C. Universal Docking Port (UDP)

Both the SWARM propulsion module and the ends of the flexible beam are equipped with docking mechanisms by which the elements can be joined together. The UDP is genderless, which allows either end of the beam to be manipulated by or joined to other copies of the docking port.

D. Digital Video Camera

The SWARM propulsion module is equipped with a Surveyor Corporation SRV-1 digital video camera. The camera includes an Analog Devices Blackfin digital signal processor for pre-processing camera information before forwarding it via serial port to the satellite. As shown in Fig. 3, the camera is mounted just above the docking port where it is used to observe the dynamics of the flexible beam.

III. Dynamics

In addition to sharing properties with a homogenous flexible beam, the SWARM beam has similarities to a multi-segmented robot arm with unactuated, elastic joints. The set of base position and rotation coordinates along with the beam joint deflections form a convenient generalized coordinate system for building the dynamics using the Euler-Lagrange method. Given the Lagrangian, with the kinetic energy T and potential energy V expressed in generalized coordinates $\mathbf{q} \in \mathbb{R}^n$, the equations of motion can be developed from

$$L = T - V \tag{1}$$

$$\frac{d}{dt}\frac{\partial L}{\dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} = \boldsymbol{\tau} \tag{2}$$

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} \tag{3}$$

The generalized force $\boldsymbol{\tau} \in \mathbb{R}^n$ is the input to the system. The dynamics derived from the Euler-Lagrange method are frequently arranged into the form of Eq. (3) where $\mathbf{M} \in \mathbb{R}^{nxn}$ is a generalized inertia matrix that represents the masses and inertias of the links along with their dynamic couplings, $\mathbf{C} \in \mathbb{R}^{nxn}$ represents velocity-dependent, or Coriolis, terms, and $\mathbf{G} \in \mathbb{R}^n$ contains position-dependent torques such as from gravity or bending.

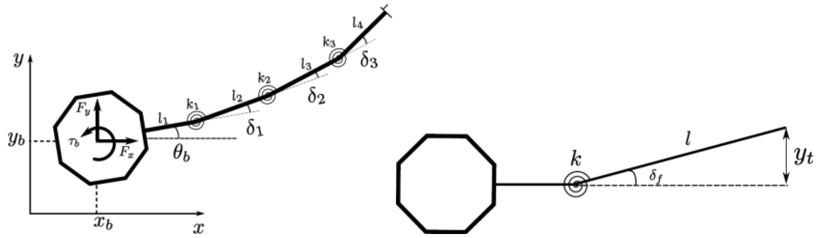


Figure 4. Parameters and states in the full 4-link dynamics (left). A simplified model with lumped flexibility (right).

Fig. 4 displays the physical states and parameters involved in the dynamics of the 4-link SWARM robot. The parameters include link lengths, moments of inertia, and stiffness values. These parameters may vary significantly across the variety of structural elements used for the assembly of a space structure, and even identical components may have different dynamic behavior if they are manipulated from different directions. This parameter uncertainty motivates the exploration of adaptive control techniques and creating a single model with a small set of parameters that represents the important flexible dynamics for multiple systems.

To create a compact model, we will consider a dynamic model of the flexible beam system with all flexible dynamics lumped into a single flexible joint as shown in Fig. 4. Developing the dynamics of the simplified model, requires a guess for the spring constant, k, that approximates the behavior of the true model. The simplification presented here is similar to the method shown in Ref. 3 in which the spring constant is determined by matching the potential energy stored in a static deflection of the true flexible beam. To apply this method to the SWARM beam, we start with an expression for the tip deflection, y_t, as a function of the joint deflection angles $\boldsymbol{\delta} = (\delta_1 \quad \delta_2 \quad \delta_3)^T$ and the link lengths shown in Fig. 4.

$$y_t = l_2 \sin(\delta_1) + l_2 \sin(\delta_1 + \delta_2) + l_4 \sin(\delta_1 + \delta_2 + \delta_3) \tag{4}$$

$$\mathbf{L} = [(l_2 + l_3 + l_4) \quad (l_3 + l_4) \quad l_4] \tag{5}$$

$$y_t = \mathbf{L}\boldsymbol{\delta} \tag{6}$$

$$\delta_f \approx \frac{y_t}{l} \tag{7}$$

For small angles, the deflection is linear in $\boldsymbol{\delta}$ with a coefficient vector of $\mathbf{L}$ and the simplified joint angle deflection, δ_f, is approximately the ratio of the tip deflection to the link length.

The potential energy in the true flexible beam can now be related to the potential energy stored in the single virtual torsional spring with spring constant k by

$$\frac{1}{2}\boldsymbol{\delta}^T \mathrm{K}_f \boldsymbol{\delta} = \frac{1}{2} k \left(\frac{y_t}{l}\right)^2, \tag{8}$$

with $\mathbf{K}_f$ as a diagonal matrix of the beam spring constants. To solve for the virtual spring constant, a valid configuration for $\boldsymbol{\delta}$ must be selected. The potential energy equation is underdetermined, but it can be solved by assuming the beam is temporarily in a static condition and therefore $\boldsymbol{\delta}$ minimizes the stored potential energy. After imposing the constraint that the deflection matches the intended tip position, $\boldsymbol{\delta}$ is calculated by solving the following optimization.

$$\min_{\delta} \left[\frac{1}{2}\boldsymbol{\delta}^T \mathrm{K}_f \boldsymbol{\delta} + \lambda(\mathbf{L}\boldsymbol{\delta}\text{-}y_t)\right] \tag{9}$$

$$\boldsymbol{\delta} = \frac{\mathrm{K}_f^{-1}\mathbf{L}^T}{\mathbf{L}\mathrm{K}_f^{-1}\mathbf{L}^T} y_t \tag{10}$$

Substituting into the left hand side of Eq. (8)

$$\frac{1}{2}\boldsymbol{\delta}^T \mathrm{K}_f \boldsymbol{\delta} = \frac{1}{2} y_t^2 \left(\frac{1}{\mathbf{L}\mathrm{K}_f^{-1}\mathbf{L}^T}\right) \tag{11}$$

and comparing to the right hand side, the spring constant is

$$k = \frac{l^2}{\mathbf{L}\mathrm{K}_f^{-1}\mathbf{L}^T}. \tag{12}$$

It is important to note this approach is only a rough estimate for a representative spring constant. Additional techniques such as system identification and online adaptation can be used to refine the estimate.

To build the Lagrangian dynamics, we can use the generalized coordinate system $\mathbf{q} = (\mathrm{x_b} \quad \mathrm{y_b} \quad \theta_\mathrm{b} \quad \delta_\mathrm{f})^T$ consisting of the base position and orientation along with the simplified beam deflection. All potential energy is stored in the virtual torsional spring due to the deflection of the beam.

$$V = \frac{1}{2} k \delta_f^2 \tag{13}$$

$$\mathbf{r}_0 = \begin{pmatrix} x_b \\ y_b \end{pmatrix} \tag{14}$$

$$\mathbf{r}_1 = \mathbf{r}_0 + \begin{pmatrix} r_a \cos\theta_b + \dfrac{1}{2} l \cos\delta_f \\ r_a \sin\theta_b + \dfrac{1}{2} l \sin\delta_f \end{pmatrix} \tag{15}$$

$$T = \frac{1}{2}\big(m_0 \dot{\mathbf{r}}_0^T \dot{\mathbf{r}}_0 + \mathbf{I}_0 \dot{\theta}_b^2 + m_1 \dot{\mathbf{r}}_1^T \dot{\mathbf{r}}_1 + \mathbf{I}_1(\dot{\theta}_b + \dot{\delta}_f\big)^2 \tag{16}$$

For the kinetic energy, the system is viewed as two rigid bodies connected by a single joint, where $\mathbf{r}_0$ and $\boldsymbol{r}_1$ represent the center of mass positions of the base and the beam respectively, and $\mathbf{I}_0$ and $\mathbf{I}_1$ are the moments of inertia of the base and link about their respective centers of mass. The resulting dynamics are partitioned into two parts: an actuated section with inputs from the robot base, $\boldsymbol{\tau}_x$, and an unactuated section for the beam dynamics

$$\begin{bmatrix} \mathbf{M}_{xx} & \mathbf{M}_{x\delta} \\ \mathbf{M}_{\delta x} & m_{\delta\delta} \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} \mathbf{C}_{xx} & \mathbf{C}_{x\delta} \\ \mathbf{C}_{\delta x} & c_{\delta\delta} \end{bmatrix} \dot{\mathbf{q}} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & k \end{bmatrix} \mathbf{q} = \begin{pmatrix} \boldsymbol{\tau}_x \\ 0 \end{pmatrix}. \tag{17}$$

IV. Estimation

A. Vision-Based Beam Measurements

The SPHERES satellites are equipped with a global estimation system that provides the location and orientation of the robot base in the satellite work area, but additional measurements are required for estimation and control of the flexible beam dynamics. To allow for docking and undocking, all sensors involved in estimating the state of the flexible beam must be mounted to the propulsion tug. Cameras provide a cheap, lightweight method for passively extracting information about motion in a scene. For vibration estimation, the camera can provide position or angular measurements for uniquely identified features attached to the oscillating body. The SWARM testbed uses a camera to observe the motion of a small infrared LED at the tip of the flexible beam. A tip-mounted LED is useful for supplying information about vibrational motion as well as positioning for docking. Practical implementations could use fiducial markers or reflectors, which have been used in vision applications on-orbit, or even an LED powered by a small solar panel.

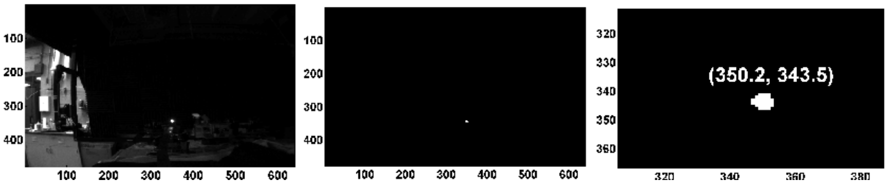


Figure 6. An example of the centroidng process (left to right): (1) An initial image is acquired with a low exposure, then (2) all pixels below a threshold are set to 0, and (3) the centroid is determined from the remaining pixels (shown zoomed in on the LED)

Digital imaging CCDs are very sensitive to the infrared spectrum, and the light from the LED easily saturates the pixels near its position in the image frame. By adjusting the camera exposure and admitting only pixel intensities above a certain threshold, the LED can be made to appear as a dot in the image plane. The centroid of the dot $(\bar{x}, \bar{y})$ can be calculated with an accuracy less than or equal to a pixel. Fig. 5 illustrates this process.

Next, the perspective projection (pinhole camera) equation[4] are applied to the image to convert from image frame pixel coordinates to camera frame distances.

$$\frac{X'}{Z'} = \beta \frac{\bar{x} - x_0}{f} \tag{18}$$

$$\frac{Y'}{Z'} = \beta \frac{\bar{y} - y_0}{f} \tag{19}$$

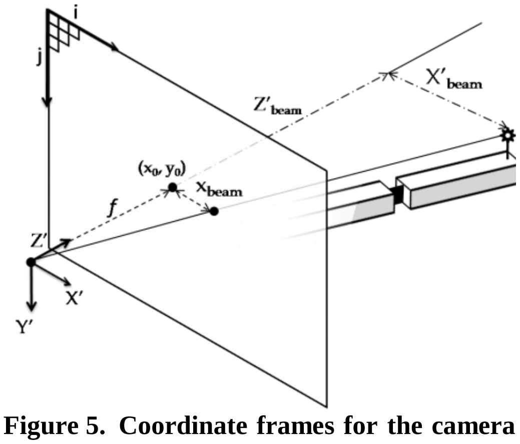


Figure 5. Coordinate frames for the camera measurement.

Subtracting the center of projection (x_0, y_0) shifts the centroid coordinates from the top-left reference frame to an image-center reference frame where they are scaled to a distance by the pixel size β. The remaining parts of Eq. (18) and (19) come from the similar triangles that share a common vertex at the focal point. Examining Fig. 6, we can see that the ratio of the image plane coordinate to the focal length f is the same as the ratio of the camera frame coordinate to the depth Z'.

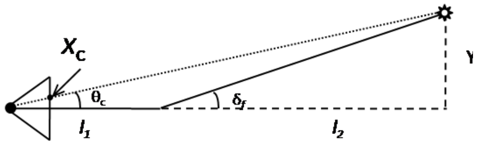


Figure 7. Measurements for the simplified beam deflection model.

As with the dynamic model, considering only the beam deflection angle introduces a significant simplification because the tip deflection angle, δ_f, can be measured directly by the camera. Figure 7 shows the measurement model for estimating the tip deflection. If the camera is not collocated with the virtual flexible joint, a small transformation must be made to estimate the deflection angle. Starting with the perspective projection equations,

$$\tan\theta_c = \frac{Y}{l_1 + l_2} = -\beta\frac{x_c}{f} \tag{20}$$

Noting that the two triangles in Figure share the beam deflection, Y , as a common side and substituting the equation above:

$$\begin{aligned} \tan\delta_f &= \frac{Y}{l_2} = \frac{l_1 + l_2}{l_2}\tan\theta_c \\ &= -\left(\frac{\beta}{f}\right)\left(\frac{l_1 + l_2}{l_2}\right)x_c \end{aligned} \tag{21}$$

$$\delta_f \approx -\left(\frac{\beta}{f}\right)\left(\frac{l_1 + l_2}{l_2}\right)x_c \tag{22}$$

Length l_1 represents the distance from the focal point of the camera to the flexible joint. l_2 is set to the length of the beam and assumed to be constant for small angles. Eq. (22) assumes small deflections.

Equation (22) shows that the primary limit on angular resolution of the camera is the ratio of the pixel size, β, to the focal length, f. This result is intuitively correct because a camera with a larger focal length observes a narrower field of view for the same number of pixels, and smaller pixel size implies that there are more pixels available to represent a given field of view. Although the deflection angle accuracy is not sensitive to the length of the beam, the deflection magnitude scales, to first order, with the length of the beam. Longer beams, therefore, require either a more tightly focused camera lens or a higher resolution camera to maintain the accuracy required for docking.

B. Beam Deflection Estimation

Although the deflection angle can be measured directly, for a full state estimate, it is still necessary to determine the angular velocity $\dot{\delta}_f$. A steady-state filter such as the Linear Quadratic Estimator (LQE) is sufficient for tracking the angle. With state $\mathbf{x} = \begin{pmatrix}\delta_f & \dot{\delta}_f\end{pmatrix}^T$, process noise $\mathbf{w}_k$, and measurement noise v_k, the filter equations are summarized in Eq. (23)-(26).

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{L}(y_k - \mathbf{H}\hat{\mathbf{x}}_k)\mathbf{w}_k \tag{23}$$

$$\mathbf{P}_{\mathrm{ss}} = \mathbf{Q} + \mathbf{A}(\,\mathbf{P}_{ss} - \mathbf{P}_{ss}\mathbf{H}^T[\mathbf{H}\mathbf{P}_{ss}\mathbf{H}^T + \mathbf{R}]^{-1}\mathbf{H}\mathbf{P}_{ss}) \tag{24}$$

$$\mathbf{L} = \mathbf{P}_{ss}\mathbf{H}^T\mathbf{R}^{-1} \tag{25}$$

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} -\left(\frac{f}{\beta}\right)\left(\frac{l_2}{l_1 + l_2}\right) & 0 \end{bmatrix} \tag{26}$$

The filter gain $\mathbf{L}$ is the optimal steady state gain for noise covariance characteristics specified by $\mathbf{Q} = E[\mathbf{w}_k\mathbf{w}_k^T]$ and $\mathbf{R} = E[\mathrm{v}_k\mathrm{v}_k^T]$. The model assumes a constant velocity with the second order dynamics in $\mathbf{A}$ and lumps the beam deflection dynamics into process noise. In the SWARM system, this is a reasonable assumption because the camera samples at approximately 20 Hz and large deflection beam dynamics are on the order of 0.1 Hz.

V. Control

A. Summary of Adaptive Control for Robot Manipulators

The following is a summary of the adaptive trajectory control for robotic manipulators presented in Ref. [5]and applies to many systems with dynamics of the form of Eq. (3) . The control law consists of an actuator command an adaptation law that adjusts parameters of the system to achieve tracking performance as if the parameters were known.

The dynamics for systems derived from the Euler-Lagrange method depend linearly on a set of constant physical parameters, $\mathbf{a} \in \mathbb{R}^p$, that represent combinations of masses, inertias and link lengths. Using this property, Equation (3) can be decomposed as

$$\mathbf{Ya} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \tag{27}$$

The matrix $\mathbf{Y} \in \mathbb{R}^{nxp}$ is known as a dynamic regressor and contains the nonlinear terms of the dynamics.

The control law makes use of a sliding variable, $\mathbf{s} \in \mathbb{R}^n$, which is the weighted sum of velocity and position errors $\dot{\tilde{\mathbf{q}}}$ and $\tilde{\mathbf{q}}$ for the trajectory specified by $(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)$.

$$\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d \tag{28}$$

$$\mathbf{s} = \dot{\tilde{\mathbf{q}}} + \mathbf{\Lambda}\tilde{\mathbf{q}} \tag{29}$$

$$\mathbf{s} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r \tag{30}$$

$$\dot{\mathbf{q}}_r = \dot{\tilde{\mathbf{q}}}_d - \mathbf{\Lambda}\tilde{\mathbf{q}} \tag{31}$$

$$\tag{32}$$

The final adaptive control law is of the form

$$\mathbf{Y}\hat{\mathbf{a}} = \widehat{\mathbf{M}}\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}\dot{\mathbf{q}}_r + \widehat{\mathbf{G}} \tag{33}$$

$$\boldsymbol{\tau} = \mathbf{Y}\hat{\mathbf{a}} - \mathbf{K}_{\mathrm{D}}\mathbf{s} \tag{34}$$

$$\dot{\hat{\mathbf{a}}} = -\mathbf{\Gamma}\mathbf{Y}^T\mathbf{s} \tag{35}$$

Actuator commands are specified by $\boldsymbol{\tau}$ and the estimated parameters, $\hat{\mathbf{a}}$, are updated by the adaptation law in Eq. (35). The first term of the actuator commands represents an adaptive feedforward command based on the estimated parameters and the reference trajectory, and the second term is a PD tracking gain. It can be shown that applying this control law results in $\mathbf{s} \to \mathbf{0}$ as $t \to \infty$, which implies, through the structure of the sliding variable, that the tracking error goes to 0 as well.

An important aspect of the stability argument is the availability of a fully actuated control input, $\boldsymbol{\tau}$. Without full actuation, the control input can only implement part of the feedforward acceleration provided by $\mathbf{Y}\hat{\mathbf{a}}$, and therefore, in its current form, the control law is not suited for underactuated systems. The next section presents an alternative form that helps to address this problem.

B. Adaptive Control for Flexible Beam Maneuvering

The idea of normal form augmentation proposed by Gu et al. in Ref. 6 was extended in Ref. 7 to derive an adaptive controller for the simplified beam dynamics. The main results are summarized here.

Starting with a desired output equation $\mathbf{y} = \mathbf{h}(\mathbf{q}) \in \mathbb{R}^m$, where m is the dimension of the control input, we augment the output with the unactuated coordinates to have the same dimension as $\mathbf{q}$.

$$\mathbf{y}_{\boldsymbol{a}} = \begin{pmatrix} \boldsymbol{y} \\ \delta_f \end{pmatrix} \tag{36}$$

$$\dot{\mathbf{y}}_{\boldsymbol{a}} = \mathbf{J}_{sq}\begin{pmatrix} \dot{\mathbf{q}}_x \\ \dot{\delta}_f \end{pmatrix} = \mathbf{J}_{sq}\dot{\mathbf{q}} \tag{37}$$

$$\ddot{\mathbf{y}}_{\boldsymbol{a}} = \mathbf{J}_{sq}\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{sq}\ddot{\mathbf{q}} \tag{38}$$

Differentiating Eq. (36) we obtain an expressions for the velocity and acceleration of the augmented output in terms of the robot coordinates as well as a Jacobian, $\mathbf{J}_{sq}$, relating the robot coordinate velocities to the output velocities.

With the requirement that the Jacobian be full rank, the relationship can be inverted to define the robot coordinates in terms of the augmented output. This leads to a modification of the original dynamic equations and the final control law.

$$\dot{\mathbf{q}} = \mathbf{J}_{sq}^{-1}\,\dot{\mathbf{y}}_{\mathrm{a}} \text{ and } \ddot{\mathrm{q}} = \mathbf{J}_{sq}^{-1}\,\ddot{\mathbf{y}}_{\mathrm{a}} \tag{39}$$

$$\bar{\mathbf{M}} = \mathbf{J}_{sq}^{-T}\mathbf{M}\mathbf{J}_{sq}^{-1} \text{ and } \bar{\mathbf{C}} = \mathbf{J}_{sq}^{-T}\mathbf{C}\mathbf{J}_{sq}^{-1} \tag{40}$$

$$\bar{\mathbf{M}}\ddot{\mathbf{y}}_{\mathrm{a}} + \bar{\mathbf{C}}\dot{\mathbf{y}}_{\boldsymbol{a}} + \mathbf{K}\mathbf{y}_{a} = \mathbf{J}_{sq}^{-1}\boldsymbol{\tau} \tag{41}$$

$$\mathbf{s} = \dot{\tilde{\mathbf{y}}}_{a} + \boldsymbol{\Lambda}\tilde{\mathbf{y}}_{a} \tag{42}$$

$$\begin{pmatrix}\boldsymbol{\tau}\\ 0\end{pmatrix} = \mathbf{J}_{sq}^{-1}\left[\mathbf{Y}\hat{\mathbf{a}} - \begin{pmatrix}\mathbf{K}_{\mathrm{D}}(\dot{\tilde{\mathbf{y}}} - \boldsymbol{\Lambda}\tilde{\mathbf{y}})\\ 0\end{pmatrix}\right] \tag{43}$$

As shown in Ref. 7, with an augmented output vector for the simplified beam model selected as

$$\boldsymbol{y}_{a} = [x \quad y \quad \theta_{b} + \alpha\delta_{f} \quad \delta_{f}]^{T} \tag{44}$$

the system has similar convergence properties as the original adaptive control law, though the choice of α can affect stability.

An important benefit of using the simplified model as a template comes from the way rigid bodies are represented in the adaptive control law. In 2D, a single rigid body has four physical parameters: mass, inertia, and the two center of mass coordinates relative to the origin of the body. The simplified beam model consists of two rigid bodies connected by a spring with an unknown spring constant, for a total of 9 parameters, each of which are adapted in the control law to achieve a tracking goal. For the first body (the base of the robot), we must know a priori the point where the second body (the beam) connects to determine the effect of forces caused by the beam's motion. This is not a significant restriction because we assume that the physical layout of the assembly vehicle is known beforehand and that the beam attaches to a pre-defined location. More importantly, we do not need any additional information about the second body because the parameters responsible for its dynamic interaction with the first body are all part of the adaptive control law. In industrial robotics the same property is used to adapt for unknown payloads carried by manipulators, which are essentially extensions of the last link of the robot arm.

Provided the simplified dynamic model continues to be a good approximation to the selected structural element and a trajectory is chosen that satisfies actuator limitations, the adaptive control system should be capable of stably manipulating beams of a variety of lengths without modification. In the same way, the controller is also capable of maneuvering an arbitrary rigid structure attached to the robot base.

C. Control Allocation

The control allocation scheme used in this work is a pulse-width modulation allocator. Thruster pulses are calculated by attempting to match the impulse that would have been delivered by a force or torque command over a specified control period. The control allocator converts a vector of commanded body forces and torques into thruster on and off times primarily using a mixing matrix. In typical satellite missions, the mixing matrix is hardcoded due to the static configuration of thrusters, but in on-orbit assembly, the mass and actuator configuration may change with each docking. For this situation a parameterized mixing scheme may be useful for updating the control allocation.

The mixing matrix is calculated online using the following inputs: center of mass (r_{cg}), thruster location with respect to the geometric center ($\mathrm{r}_{\mathrm{act}}$), and the direction vector of the thruster force ($\mathrm{D}_{\mathrm{act}}$). The mixing matrix is a matrix of the force and torque generated by each thruster. The calculation of the mixing matrix (Ξ) is given in Eq. 47. The force directions are included as is, but the torque directions are based on the current center of mass of the satellite. To calculate the torque directions, the thruster locations are converted to be with respect to the center of mass. The locations are then crossed with the force direction to determine the torque generated by that thruster. This allows for a reconfigurable control allocation algorithm, which can properly set the mixing matrix online during the assembly process to accommodate for changing physical configurations.

$$\mathbf{r} = \mathbf{r}_{act} - \mathbf{r}_{cg} \tag{45}$$

$$\mathbf{D} = \mathbf{r} \times \mathbf{D}_{act} \tag{46}$$

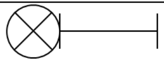
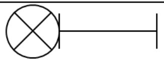
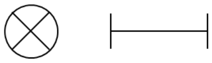
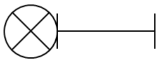
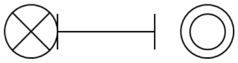
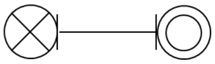
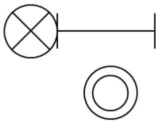
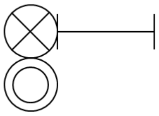
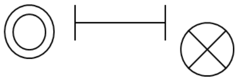
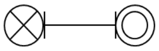
$$\boldsymbol{\Xi} = [\mathbf{D}_{act} \quad \mathbf{D}_{T}] \tag{47}$$

VI. Flat Floor Assembly Testing

A. Assembly Test Matrix

Testing was performed in an incremental manner. The sequence of tests was developed to incrementally build up in complexity, while also accounting for several hardware availability constraints. These constraints were: maximum availability of 3 SPHERES satellites (SPH), maximum availability of 7 Universal Docking Ports (UDP), maximum availability of 1 Flexible Beam (FB) Module, and maximum availability of 3 air carriages.

The test matrix was designed to demonstrate assembly of a complex space structure with flexible dynamics with four objectives: (1) Control of a flexible structure; (2) Docking control at both ends of the flexible structure; (3) Reconfiguration and control after docking; and (4) Reconfiguration and control after undocking. Table 1 is the minimum number of tests that demonstrate all the science objectives. Each phase of testing optimizes the science gain from each test to enable the next.

T	Initial Config	Initial Config	Assembly Steps	Science Obj	Final Config
1	- SPH+FB			Demonstrate control	
2	- SPH floating - FB fixed		Dock to SPH to FB	Dock to a flexible structure	
3	- SPH1+FB floating - SPH2 fixed		Dock flexible end to fixed sphere (parallel to flexible beam)	Docking control with non-collocated actuation	
4	- SPH1+FB floating - SPH 2 fixed		Dock SPH1 end of SPH1+FB to fixed SPH2	Demonstrate control docking perpendicular to SPH (collocated docking)	
6	- SPH1 floating - FB floating - SPH2 fixed		-Dock SPH1 to FB -Dock SPH1+FB to SPH2	Full assembly test. Incorporates all aspects individually tested above	

◎ SPH1 (Prop Module) |—| FB (Flexible Beam)   ⊗ SPH2 (Target)

Table 1. SWARM test configuration matrix.

B. Results

In each phase of the assembly sequence an online path planner generated a trajectory for the base of the satellite tug to track between its starting location and a target location in the work area. The trajectories followed a bang-bang profile to minimize assembly time and command the maximum amount of thrust to overcome friction on the flat floor.

Selected test results are shown to demonstrate successful completion of the test sequence. All results shown

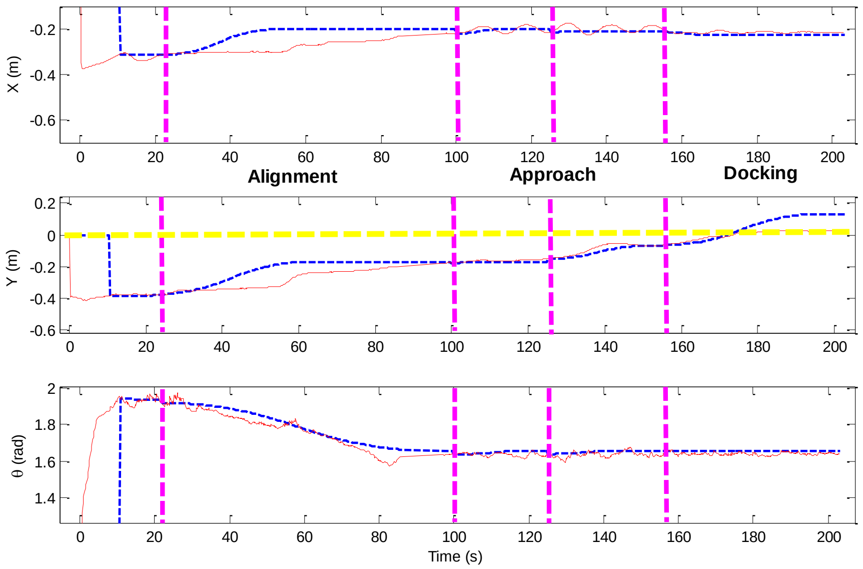


Figure 8. Test 3: trajectory tracking and docking while attached to a flexible beam.

were performed on hardware at the Flat Floor testing facility at MIT using an early version of the adaptive controller that used primarily PD control. Figure 8 shows the trajectory tracking performance for Test 3. Test 3 includes maneuvering while attached to a flexible beam, approaching and docking to a fixed target. This test demonstrates control of a flexible structure, reconfiguration and maneuvering after docking, and docking of the flexible end of a beam to a target. The horizontal line represents the docking target. Once it reaches the horizontal line, the docking ports have aligned and docked. The dashed vertical lines indicate maneuver changes to enter the next phase of the docking approach. The name of the phase is written between the vertical lines. For example, attitude orientation is the first phase of the docking sequence. The first plots show the X and Y position of the satellite in the workspace along with its rotation in radians. There is some lag between the commanded trajectories and the satellite response due to the air carriages friction on the flat floor surface. Successful docking occurs at t=170s (Figure 8).

Test 6 attempted to demonstrate the entire assembly process. The test consisted of a free floating satellite docking to a fixed beam, then maneuvering to dock the flexible end of the beam into another fixed docking port.

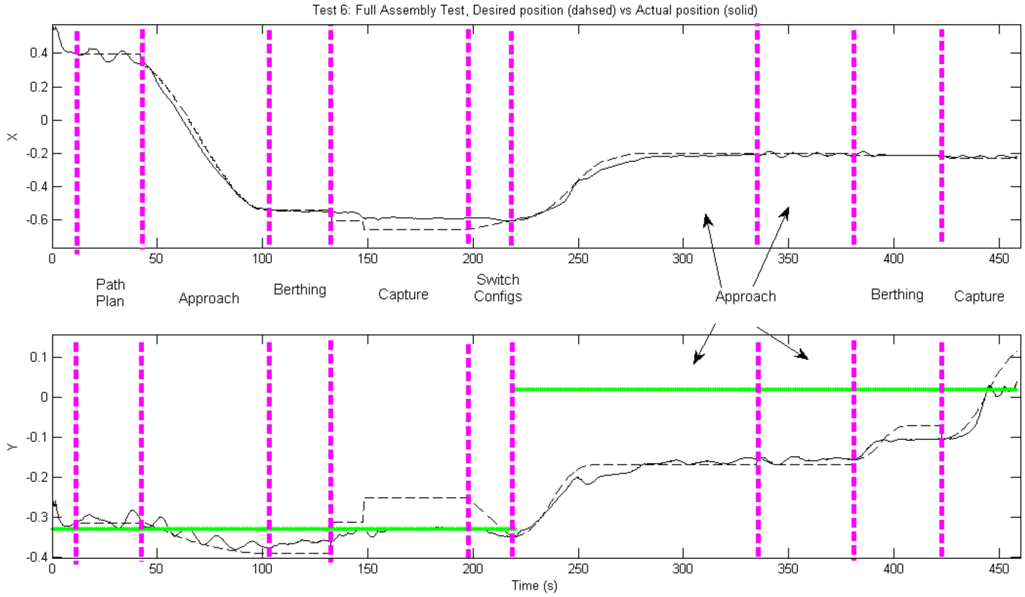


Figure 9. Test 6 Trajectory Tracking results, Full assembly test.

Operational difficulties with friction on the flat floor prevented a complete demonstration of the assembly maneuver, but several tests were completed to the final docking phase without a successful capture. Figure 9 shows an example test. The two green lines are the two docking targets, first, the position target for docking to the beam, and second, at the target ford docking the beam end to a fixed docking target. The trajectory plots show the X and Y position of the satellite as it maneuvers the beam, and the assembly phases are described in between the plots. Figure 9 shows quite good tracking throughout the entire test, leading to a successful satellite-beam docking, and a near miss of the beam-target docking.

VII. Conclusion

To develop an understanding of the behavior of flexible dynamic systems and provide a template for the control system design, a lumped flexibility model was developed into dynamic equations with an Euler-Lagrange analysis. The simplified model proved to introduce several simplifications in the development of a control and estimation approach for the SWARM system. Though early versions of this system were used in the hardware tests presented in this study, the full implementation in hardware remains as future work.

The SWARM program has successfully demonstrated the maneuvering and docking of a flexible beam in a 2D flat floor environment. Future work will be required to perfect a complete assembly of several elements into a complete structure. Since a major hurdle is overcoming the limitations of the 2D air-bearing testbed, future assembly tests may be performed on the SPHERES facility aboard the ISS, pending hardware upgrades to support assembly.

Acknowledgments

This work was performed under NASA SBIR Contract No. NNM07AA22C Self-Assembling Wireless Autonomous Reconfigurable Modules.

References

[1]Hoff III, N. R., Mohan, S., Nolet, S., and Miller, D. W., "Docking and reconfiguration of modular spacecraft: preliminary SWARM testing at MSFC," *Proc SPIE*, Vol. 6555, Orlando, FL, 2007, pp. 65550W-8
[2] Mohan, S., N. Hoff, S. Nolet, and D.W. Miller. "Autonomous Satellite Reconfiguration using SPHERES Micro-Satellite Testbed." *Dynamics and Control of Systems and Structures in Space Conference*. London, England, July 16-20, 2006.
[3] Ge SS, Lee TH, Zhu G. "Tip tracking control of a flexible manipulator using PD type controller," *Control Applications, 1996., Proceedings of the 1996 IEEE International Conference on*. 1996:309-313.
[4] Horn, BK., *Robot Vision*, MIT Press, Cambridge, MA 1986.
[5] Slotine, J. E., & Li, W., *Applied Nonlinear Control*. Upper Saddle River, New Jersey 1991.
[6] Gu, Y., & Xu, Y., "A normal form augmentation approach to adaptive control of space robot systems," *Dynamics and Control*, *5*(3), 275-294.
[7] Katz, J. G. "Estimation and control of flexible space structures for autonomous on-orbit assembly," Masters Thesis, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA 2009.