

SPHERES Reconfigurable Framework and Control System Design for Autonomous Assembly

Swati Mohan and David W Miller
Massachusetts Institute of Technology

Abstract:

Reconfigurable control system design is a key component for enabling autonomous on-orbit assembly. Current research on reconfigurable control systems focuses on adapting to failures. However, for assembly scenarios, the reconfiguration is necessitated by changing mass and stiffness properties. This paper provides a brief description of existing reconfigurable control system technology and develops a framework to incorporate reconfiguration into an existing baseline system to account for mass property variations. The reconfigurable control system framework has been developed and implemented using the SPHERES (Synchronized Position Hold Engage Reorient Experimental Satellites) testbed as the baseline system. The framework highlights the elements that need to be updated, introduces a variable p that captures the configuration, and details the updates necessary in the key algorithms to calculate the model on-line using p . Results are presented from the implementation on the SPHERES, focusing on the reconfigurable estimator. Plans are presented for an integrated assembly test that demonstrates the maintenance of stability, fuel efficiency, and accuracy throughout configuration changes that occur during assembly.

Introduction:

Reconfigurable control systems are used in a variety of applications, such as docking, servicing, high efficiency flight, and assembly. Autonomous assembly is a critical technology for future missions such as large space telescopes, on-orbit space stations, and a lunar base. One particular scenario for autonomous on-orbit assembly is to use an assembler tug to maneuver the different payload items. In this scenario, the dynamics of the assembler tug will vary greatly at each docking and undocking maneuver, based on the properties of the payload, such as mass and inertia. Depending on the relative sizes of the tug to the payload, this could be a very significant change. In order to maintain adequate control performance, it is important to account for the property changes. The implementation of a reconfigurable control system would be able to account for the frequent mass property changes, while also introducing adaptability into the system design. The sequence of assembly would not need to be pre-determined, nor would all configurations have to be pre-computed.

Most current research on reconfigurable control systems has been point designs, focused primarily on reconfiguration to adapt for failures. Little framework or example cases exist on how to guide engineers to design with reconfiguration in mind. Reconfigurable control systems can be classified based on the amount of prior information assumed by the designer. One end of the spectrum has all necessary information known *a priori* and can schedule the appropriate models. This method highly couples the assembly sequence into the control, which minimizes the system's ability to adapt in case of disturbances or failures. Examples of this type of reconfigurable control system is the control of the International Space Station (ISS) during docking with the Space Shuttle [1], and Maybeck and Stevens multiple model adaptive control [2] which stores a bank of Kalman filters. These methods could require large memory storage capability.

The other end of the spectrum requires on-line identification of the model using system identification techniques that can be costly in resources, such as time and fuel. A key example of this area of research is Wilson's use of a recursive least squares method to determine the center of mass and inertia of a spacecraft [3]. Some work has also been done to identify a methodology for reconfiguration, though it is specific to failure scenarios [4].

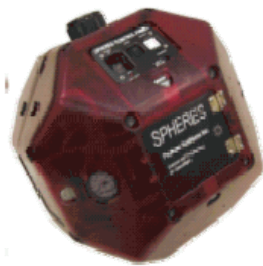
The development of the reconfiguration framework in this paper seeks to accommodate the following two scenarios:

- 1) *Docking to an active payload*: In some autonomous assembly scenarios, the tug is assembling a payload that has maneuvering capability. In these scenarios, it would be beneficial to make use of the actuation and sensing capability of the payload. This could lead to several benefits: fuel savings on the tug; greater mobility for the tug-payload system; and more accurate estimates since some of the sensors on the tug could be blocked by the payload.
- 2) *Docking to a passive payload*: In this scenario when the assembler tug docks to a passive payload, the center of mass of the system shifts. When the payload is larger than the tug, the center of mass shifts to be outside of the thruster envelope. No thrusters or sensors are available for use on the payload.

The algorithms presented in this work are developed for the scenario 1, as scenario 2 is a subset of scenario 1 when there are no actuators or sensors on the payload. This paper upgrades the baseline algorithms on the SPHERES (Synchronized Position Hold Engage Reorient Experimental Satellites) testbed. The following sections give an overview of SPHERES (including description of the baseline algorithms), description of the reconfiguration framework, details on the updates to the algorithms to make them reconfigurable, and show the experimental results, focusing on the estimation.

SPHERES Overview:

SPHERES is a formation flight testbed aboard the ISS (Fig. 1). It consists of three free-flying satellites, about 20 cm in diameter. The satellites operate inside a 1.5m³ volume aboard the ISS, using a pseudo-GPS navigation system. Each SPHERES satellite has twelve thrusters, fueled by a single CO₂ tank, for full six degrees of freedom actuation [6].



(a) SPHERES satellite



(b) Satellites aboard the ISS

Figure 1: SPHERES hardware pictures

The satellites use an Extended Kalman Filter (EKF) for estimation, using primarily time of flight measurements from the ultrasound beacons and receivers, as well as gyroscopes [7]. The nominal control architecture is a mix of Proportional-Derivative (PD) and Proportional-Integral-Derivative (PID) controllers. The baseline control allocation algorithm on SPHERES, a thruster based pulse-width modulation scheme, uses the symmetry of the thruster placement to specify

thruster pairs. These forces/torque pairs are pre-determined using the knowledge of the thruster placement. Each thruster in a pair produces the exact opposite force and torque as its pair thruster. Thus, the conversion from control forces to thruster commands can be simplified to extrapolate all twelve thruster commands based on six pair forces. The SPHERES testbed is used as the baseline representative system, serving both as the tug and the payload.

Method:

In developing a reconfigurable control system framework, the first step is to identify how the change in mass properties influences the control system architecture. Figure 2 shows a representative control system block diagram, with the software elements that need to be updated outlined in a solid line. The physical elements that change due to docking are outlined in a dashed line. Within the estimation loop, the plant model in the estimator and the measurement model, which contains the sensor configuration, need to be updated. The actuator model (i.e. thruster configuration) is updated through the control allocation algorithm. The controllers must also be updated, whether it is the control law itself that is updated or simply the gains.

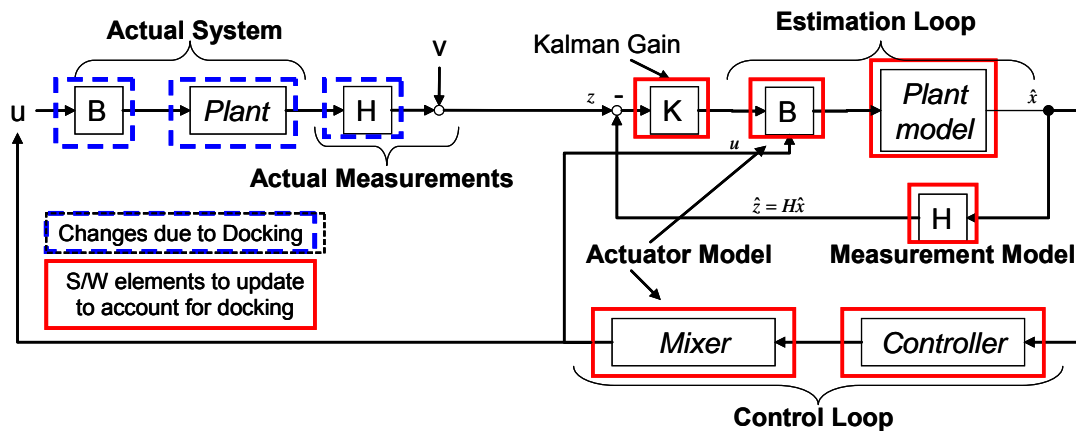


Figure 2: Control system block diagram with necessary elements to update highlighted

Each element can be analyzed to identify the physical properties that are used that require updating. The physical properties constitute three major model components: plant model, actuator model, and the measurement model. The **plant** model contains the physical mass property information of the system: mass, inertia, and center of mass. The **actuator** model contains the thruster configuration: location, direction, and force of each thruster; thruster health; and the location of the reference point from where the location of the thruster is measured. Similarly, the **measurement** model contains the sensor configuration: type of sensor; location and direction of sensor; sensor health; and reference point/axis for each sensor. The set of these variables is defined as the property vector p . One last variable included in the property vector is the docking status. This variable allows the system to identify when a configuration change occurs. The vector p is passed to each algorithm in order to calculate its respective model. The combination of all of these models (plant, actuator, and measurement) constitutes the model of the system, M . In state space terminology, the model M specifies the A (state), B (input), C (output), D (feedforward), H (measurement), and K (control gain) matrices.

The incorporation of p into a control system framework is shown in Figure 3. The first step in the framework is to initialize the property vector p and the model M . The tug starts out with its original model, without being docked to any payload. This model is used in the control loop to

run the estimator, controller, and control allocation algorithms. During the control loop, the system is continually checking if p changes. Specifically, if the docking status variable changes, then a configuration change has occurred. The property vector p is obtained from the payload by communication. The new model is calculated by using the property vectors from the tug and the payload. The continuous observance of the property vector and maintenance of the model constitutes the reconfiguration loop.

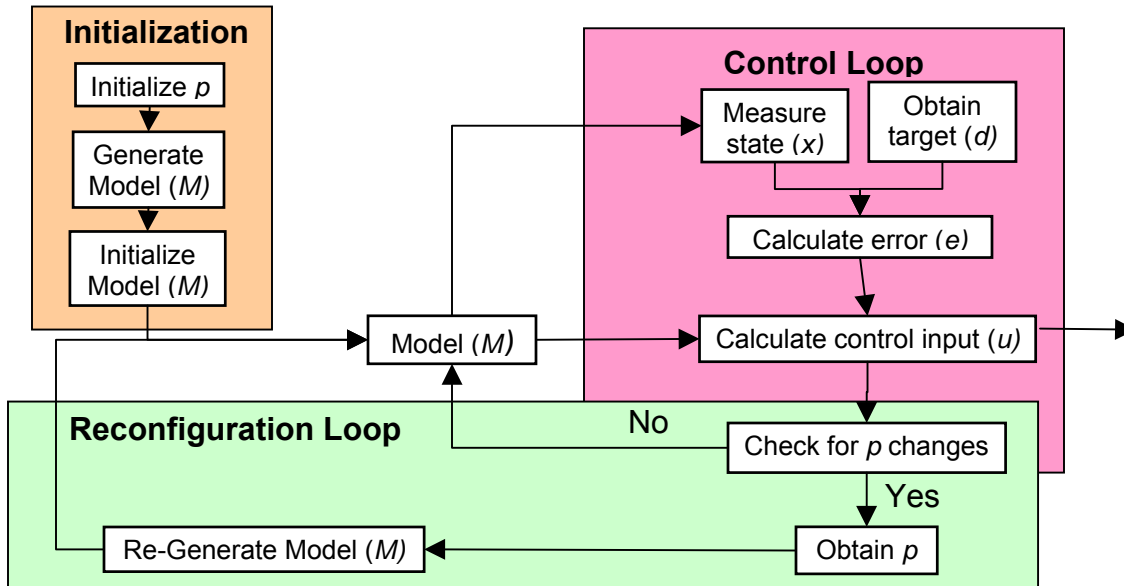


Figure 3: Block diagram of reconfiguration framework

There are four key algorithms in the reconfiguration framework: model calculation, estimator, controller, and control allocation. The three latter algorithms are coded from the baseline system to be a function of the property vector p . Each algorithm is detailed in the following section.

Algorithm Development:

Of the four algorithms to be developed, three already partially exist in the baseline system. The algorithms are described in the following order: control allocation, controller, estimator, and model generation.

The baseline control allocation algorithm is a pulse-width modulation thrust allocation algorithm that relies on a hardcoded matrix that converts the control input into thruster forces. The key update is to generate the matrix online based on the plant and actuator models. The variables of the p vector used are center of mass, location of thruster, direction of thruster, and force of thruster. Figure 4 compares the computation flow of the baseline algorithm and the reconfigurable algorithm, with the variables of p highlighted in red. Detailed implementation discussion and testing results can be found in Mohan [5].

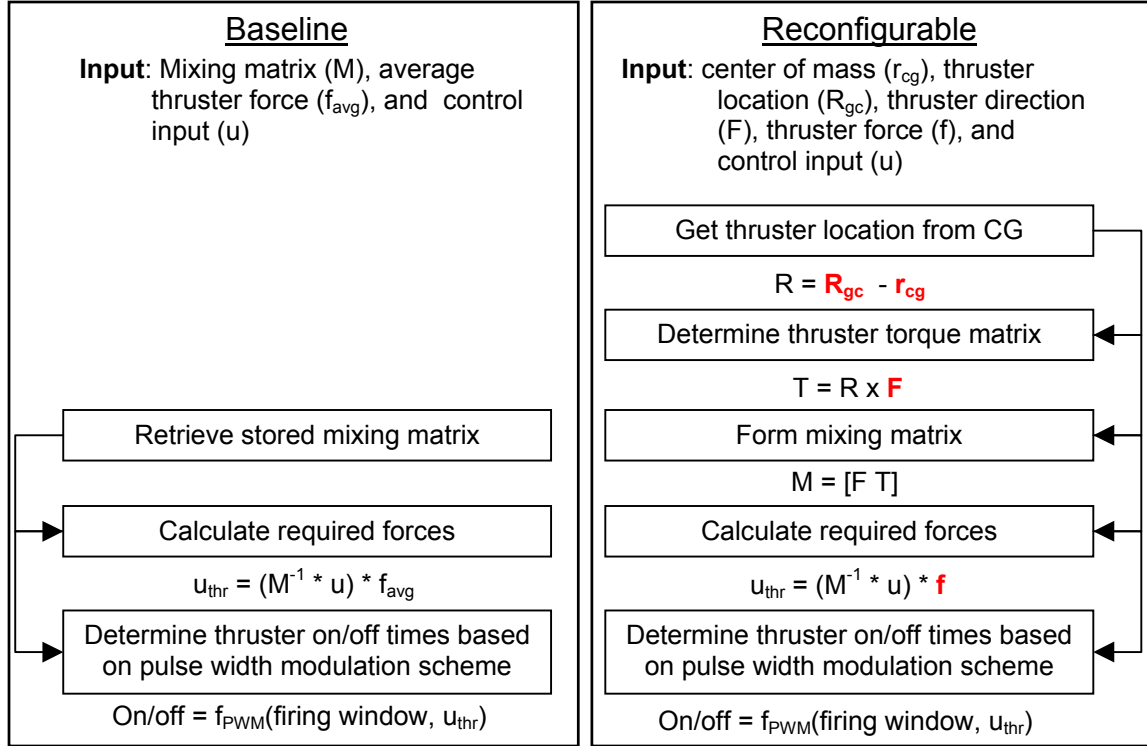


Figure 4: Control allocation flow diagram, baseline vs. reconfigurable

The baseline controller is a PID controller. For this controller, only the gains need to be updated to be reconfigurable. The gains are calculated using the Eq. (1) for attitude and Eq. (2) for position, where w_n is the control bandwidth, m is the mass, I is the inertia, ζ is the damping ratio, and T is the time constant associated with the integral term.

$$K_p = \left(w_n^2 + \frac{2\zeta w_n}{T} \right) \begin{bmatrix} I_x & & \\ & I_y & \\ & & I_z \end{bmatrix} \quad K_i = \left(\frac{w_n^2}{T} \right) \begin{bmatrix} I_x & & \\ & I_y & \\ & & I_z \end{bmatrix} \quad K_d = \left(2\zeta w_n + \frac{1}{T} \right) \begin{bmatrix} I_x & & \\ & I_y & \\ & & I_z \end{bmatrix} \quad (1)$$

$$K_p = \left(w_n^2 + \frac{2\zeta w_n}{T} \right) m \quad K_i = \left(\frac{w_n^2}{T} \right) m \quad K_d = \left(2\zeta w_n + \frac{1}{T} \right) m \quad (2)$$

The variables of the property vector used for gain calculation are inertia and mass. The bandwidth is also a function of the mass. In a rigid body case, the bandwidth can be scaled proportional to the increase in mass. For example, if mass doubles, bandwidth is decreased by half to maintain the same fuel efficiency and accuracy. These gains can be used in the baseline PID control law, as given by Eq. (3). The variables q and p are the integration terms for attitude and mass respectively. The variable e is the state error and t is the integration period.

$$\begin{aligned} q &= q + e * t & p &= p + e * t \\ u &= u + 2K_p e + 2K_i q + K_d \dot{e} & u &= u + K_p e + K_i p + K_d \dot{e} \end{aligned} \quad (3)$$

The updates to the control allocation and controller algorithms complete the updates to make the control loop reconfigurable. However, the estimation loop must also be updated. This involves updating the baseline EKF. There are three aspects in the EKF that need to be updated: state propagator, thruster propagation, and measurement model. This work currently assumes all rigid body motion, so the dynamics equations in the state propagator remain

fundamentally the same. The dynamics propagation equations are given in Equation 4, x^+ refers to the estimate after the update, while x^- refers to the value prior to update.

$$\begin{aligned} x^+ &= x^- + \dot{x}dt \\ \dot{x}^+ &= \dot{x}^- + \frac{fdt}{m} \end{aligned} \quad (4)$$

The mass (m) and thruster force (f) need to be updated to make the state propagator reconfigurable. The mass is obtained from the property vector p , while f is obtained through the thruster propagation. Figure 5 shows the algorithm updates for the estimator thruster propagation.

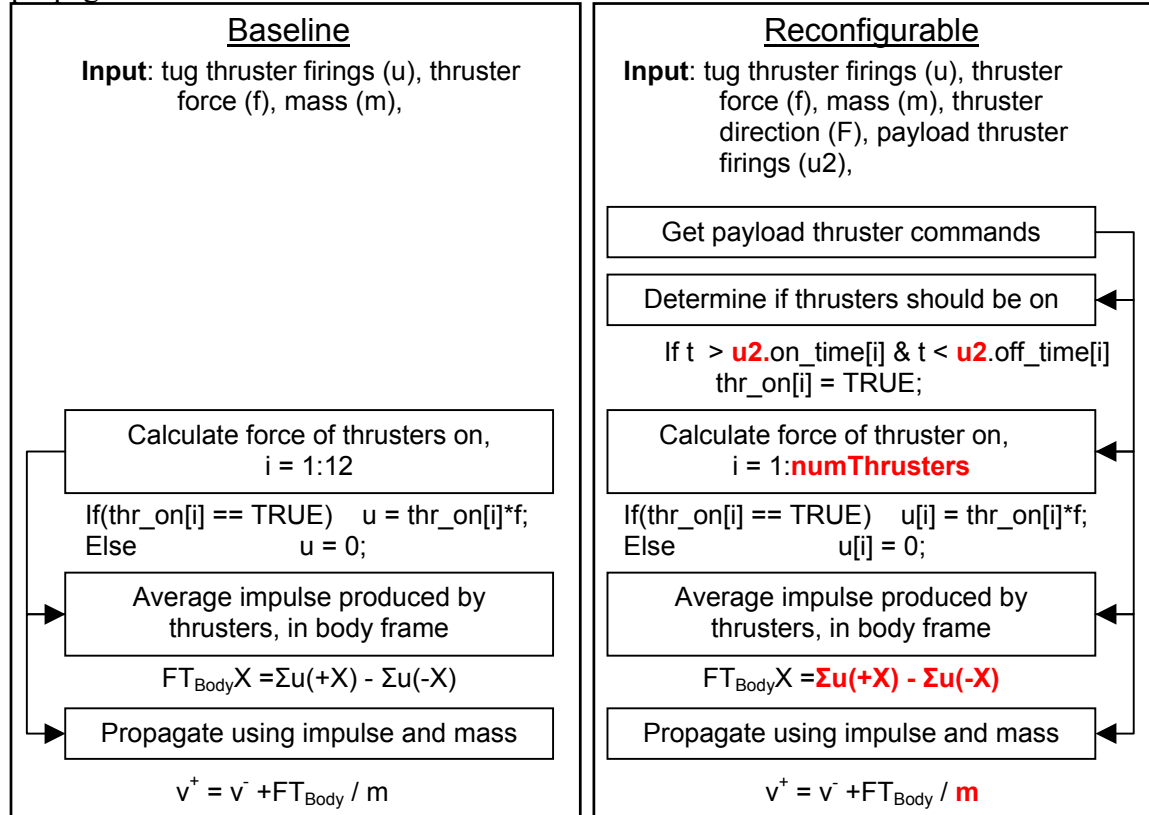


Figure 5: Thruster propagation flow diagram, baseline vs. reconfigurable

When the sensor configuration changes due to the addition or removal of sensors, it is important to update the measurement model. In the baseline case of SPHERES, there are three types of sensors: gyroscopes, accelerometers, and ultrasound receivers. Thus, the docking of another satellite would increase the number of the sensors, but not change the type. Therefore, the parameters that need to be updated are number of receivers, receiver positions, number of gyroscopes, number of accelerometers, and the bias for each gyroscope. For this initial study, only the additional receivers were considered. In order for the additional sensors to be properly incorporated, the raw measurements from the additional sensors must be communicated such that all measurements can be filtered together. The subsequent accuracy performance of the EKF is dependent on many factors, such as the communication time to receive the raw measurements, knowledge of the new sensor locations, distance of the sensors to the geometric center, as well as the method of incorporation. For this testing, raw measurements were downloaded from both satellites and post-processed to identify the estimator performance without communication delay.

Accounting for the communication delay would required changing the structure of how the measurements are processed, from as obtained to batch processed, since the SPHERES communication delay is 40ms or greater. Measurements were processed together in the pre-filter code, which was updated to account for the increased number of receivers. Since the receivers are identical, the heuristics remain the same. Details of the heuristics of the baseline EKF pre-filter can be found in Nolet [7].

The fourth component is the model generation algorithm. This is a key component of the reconfigurable control system because it allows for the maintenance and generation of the current model. The inputs to this algorithm are the property vector of the tug (p_{tug}), the property vector of the payload (p_{pl}), and the interface where the payload has docked to. The property vector p is specified in Equation 5.

$$p = \begin{bmatrix} m & \vec{r}_{cg} & I_x & I_y & I_z \\ \vec{r}_{thr} & \vec{F}_{thr} & \vec{D}_{thr} & \vec{R}_{ref-thr} & \\ \vec{r}_{sensor} & \vec{F}_{sensor} & \vec{D}_{sensor} & \vec{R}_{ref-sensor} & \end{bmatrix} \quad (5)$$

The components of the vector are mass (m), center of mass (r_{cg}), principal moments of inertia (I_x, I_y, I_z), thruster locations (r_{thr}), thruster force (F_{thr}), thruster directions (D_{thr}), thruster reference point ($R_{ref-thr}$), sensor locations (r_{sensor}), sensor type (F_{sensor}), sensor direction/axis (D_{sensor}), and sensor reference point ($R_{ref-sensor}$). Components in Eq. (5) with an arrow are vectors also, that are combined to form the larger property vector, p . The two property vector inputs must be combined to yield a new property vector that is representative of the combined system. Figure 6 is a block diagram of the calculation of some of the different elements of the property vector.

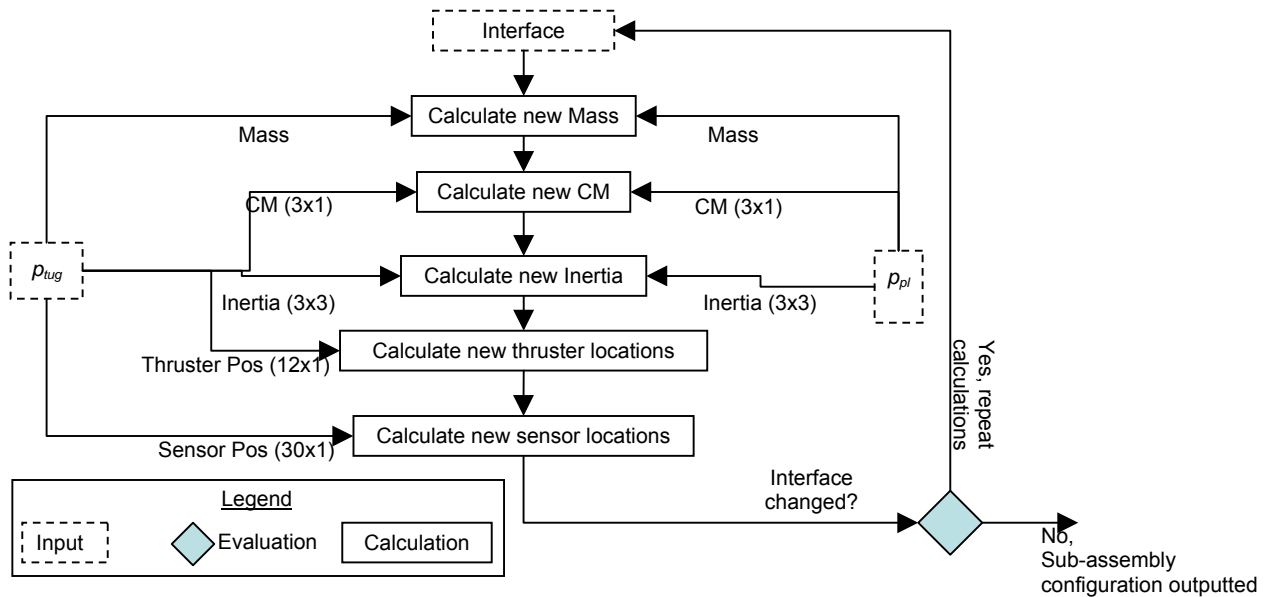


Figure 6: Model generation flow diagram example for a passive payload

For the mass, it is simply an addition of the two masses. The center of mass is recomputed, maintaining the tug's reference point. The center of mass of the payload is determined in the tug's reference point by using the information of which docking face it is attached to. Eq. (7) gives the new center of mass, where r_{dpl} is the location of the tug's docking port in the tug's

reference and r_{dp2} is the location of the payload's docking port face in the payload's reference frame. The minus sign in front of r_{dp2} switches it to be the location from the docking port to the center of the payload reference frame.

$$\vec{r}_{cg} = \frac{m_{ug} \vec{r}_{ug} + m_{pl} (\vec{r}_{dp1} - \vec{r}_{dp2} + \vec{r}_{pl})}{m_{ug} + m_{pl}} \quad (7)$$

The inertia is similarly obtained by using the parallel axis theorem. First, the inertia about the center of mass is converted to be about the interface point for both the tug and payload. Then, the coordinate system of the payload is rotated to match the coordinate system of the tug. Finally, the inertias are added and recalculated to be about the combined system's center of mass. The thruster positions and sensor positions are simply updated to be with reference to the new center of mass. More detailed information about the model generation algorithm, such as update of thruster force directions, can be found in Mohan [8].

This section provided a detailed description of the framework and reconfiguration methodology, as well as brief descriptions of each algorithmic component of the framework. The development of the property vector, identification of the elements that need to be updated, and the framework of the reconfiguration loop can be generalized to any system. A simple rigid body baseline system is chosen for ease of implementation. The algorithm details described, particularly for the generation of p and model calculation, are specific to the baseline system and will need to be modified for each system. The following section focuses on estimator testing results.

Results

This paper focuses on the hardware results for the reconfigurable estimator. Hardware results for the controller and the control allocation algorithms can be found in Mohan [5]. Implementation results for the model generation algorithm can be found in Mohan [8]. The estimator was tested in two configurations: static and dynamic. The static tests confirm the conversion of measurements into a state, while the dynamic tests confirm the state and thruster propagation. The satellite is stationary during the static tests and placed in a known location. Multiple locations were tested to ensure convergence with different measurement conditions. Figure 7 shows the convergence and maintenance of the position state. A spike is seen near $t=0$ as the estimator updates the position from the initial guess. The estimator converges to the appropriate value within 2 s and achieves the sub-centimeter accuracy, similar to the baseline estimator.

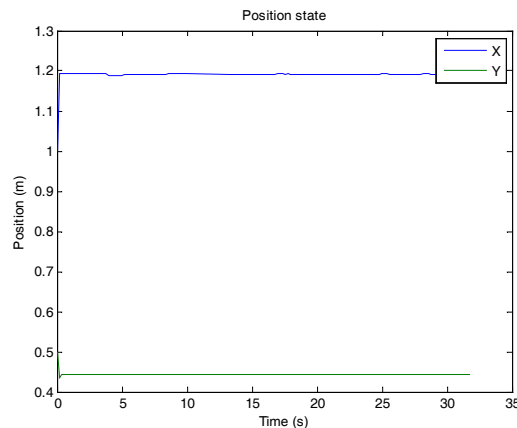


Figure 7: Hardware results of reconfigurable estimator tests, stationary at (1.2m, 0.45m)

In the dynamic tests, two identical satellites are rigidly attached via Velcro at one face. The satellites perform an open loop firing sequence along the axis of docking, as shown in Figure 8. An initial 10 s is allowed for estimator convergence (Phase 1 in Fig. 8). Then, satellite one (SPH1) fires in the $-X$ direction, shown as Phase 2 of Fig. 8. Next, satellite two (SPH2) fires in the $+X$ direction, during Phase 3 of Fig. 8. When firing, the satellites fire for 600ms every 1s, for a total duration of 10 s.

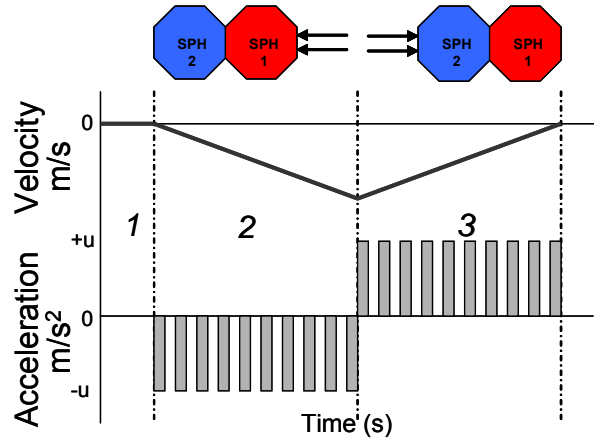


Figure 8: Test set-up for dynamics reconfigurable estimator tests

Figures 9 and 10 show the estimator results from the dynamic tests. Figure 9 shows the velocity performance in simulation, while Figure 10 shows the performance in hardware. The dashed line represents the estimated velocity, while the solid line is the measured velocity obtained by integrating the accelerometer measurements. When using the “without reconfigurable thruster propagation”, the estimator has the single satellite model. The estimator does not know a second satellite (i.e. SPH2) is attached, so it does not account for its added mass or thruster firings. From $t=20s$ to $t=30s$, SPH1 is firing thrusters. The estimator accounts for SPH1 firing, but accounts only for SPH1’s mass, which is half of the total mass. Thus, the velocity estimate overshoots the true velocity. From $t=30s$ to $t=40s$, SPH2 is firing thrusters. The velocity estimate has a delay from following the true state in this phase because it does not incorporate SPH2’s firings. It must rely on the ultrasound measurements to update the velocity state. These two issues are removed when using the dual satellite model. Therefore, the estimated velocity tracks significantly better when using the dual satellite model than the single satellite model. The jump in the “true” y velocity in Figure 10 reflects a slope on the table which appears after stiction is removed when the satellite starts firing. In general, updating the thruster propagation in the estimator improves the velocity estimate, which improves fuel usage.

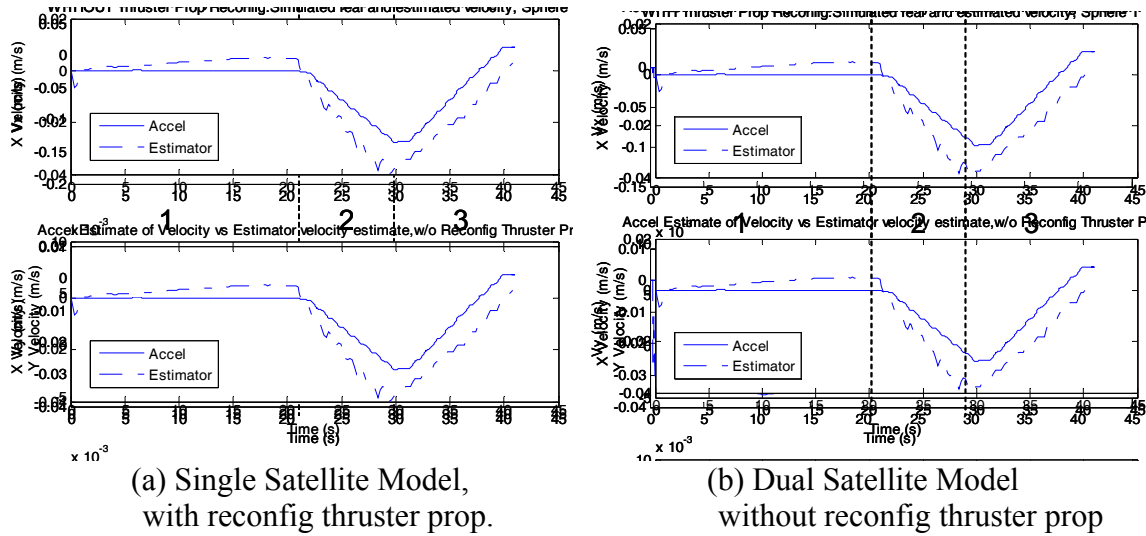


Figure 9: Simulation results for reconfigurable estimator

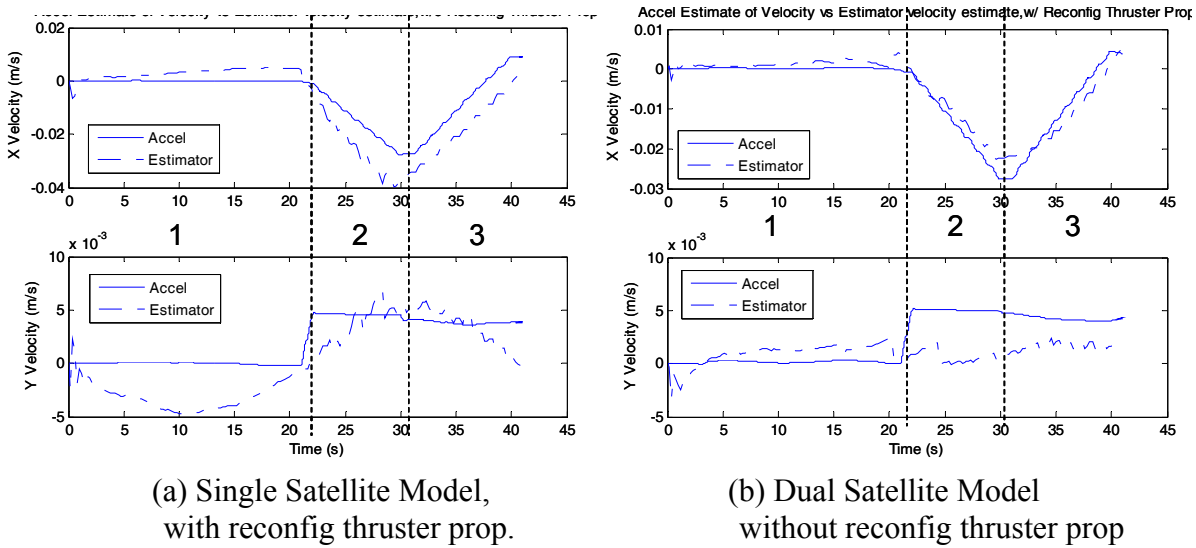


Figure 10: Hardware results for reconfigurable estimator

Performance when integrating ultrasound receiver measurements from both satellites has mixed results. Accuracy of the estimator was tested using 3, 4, and 4 beacons, and either receivers from single satellite or receivers from both satellites. Nominal operations use 5 beacons, while 3 beacons is the minimum needed for the state estimator to converge. The positioning of the satellites was such that the second satellite was directly between the first satellite and one of the beacons. SPH2 blocks SPH1's face that is directly pointing to Beacon 1 (Fig. 11). Thus, when SPH1 processes the measurements from Beacon 1, it must use a measurements received from a side face. The EKF estimates the position of the center of SPH1.

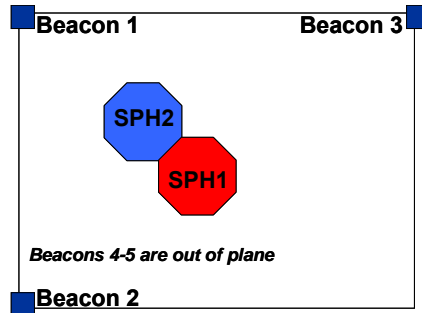


Figure 11: Set-up for test of reconfigurable estimator with joint sensing

Figure 12 shows the performance of the position estimate when ultrasound receivers are incorporated from two attached satellites compared to baseline performance. Four test cases are plotted in Figure 12: single satellite configuration (20 receivers) in simulation and hardware, and dual satellite configuration (40 receivers) in simulation and hardware. The simulation results (dashed lines) showed that for 4 and 5 beacons, using 20 receivers was marginally better than using 40 receivers. The benefit to using more receivers should be to obtain a better estimate by reducing the noise. However, the improvement is minimal after a threshold number of receivers, which differs from case to case. A factor causing a decrease in performance is the method for calculation of the estimated measurements. The EKF uses the estimated attitude to convert the receiver locations from the body frame to the inertial frame. The noise of the attitude estimate has a larger affect on the estimated measurement the farther the receivers are from the center of the satellite. In this set-up, the receivers from SPH1 were located at a radius of 10 cm, while SPH2's receivers were 30 cm away. The combination of these two effects causes a small overall decrease in performance for nominal operations.

In operations with 3 beacons, using the receivers on SPH2 provides an overall better estimate in simulation. The body blockage of SPH1 to Beacon 1 is a bigger source of error when there are only 3 beacons instead of 5. When using SPH2's measurements, it can use the face directly facing the beacon, which leads to a better estimate. In the hardware tests (solid lines), using 20 receivers is always better than using 40 receivers, even with blockage of the beacon. This is most likely due to the variability of the attachment. The satellites are attached via Velcro, which can be a very imprecise mechanism. Discrepancies between assumed receiver locations on SPH2 and the actual locations leads to an inaccurate model, and hence slight worse estimator performance. This source of error can be minimized by using a rigid mechanical interface with high accuracy and repeatability of capture. Given these results, the incorporation of additional sensors, particular if they are the same type as those that already exists, should be implemented if

- 1) the additional sensors cause a significant reduction of noise, or
- 2) the additional sensors replace sensors that can no longer be used due to blockage.
- 3) obtaining the measurements from the additional sensors do not introduce a significant time delay

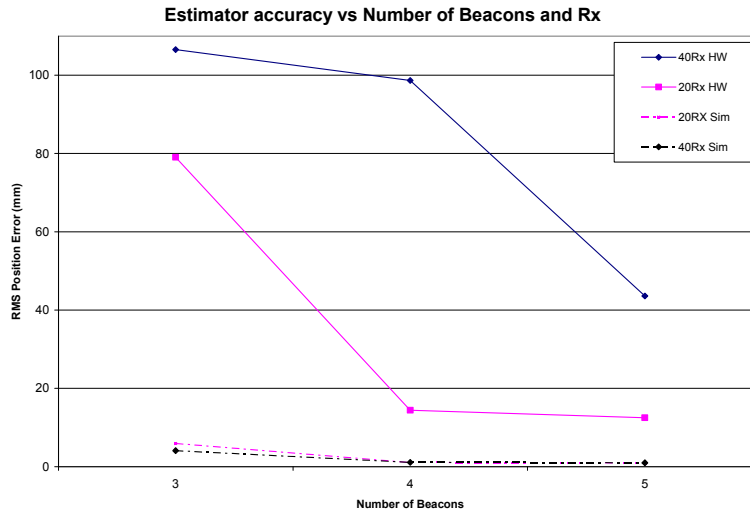


Figure 12: Sensor reconfig results for reconfigurable estimator

The integrated reconfigurable control system will be tested on the ISS through a one satellite assembly test. The tug satellite will maneuver to the payload satellite, attach to it, maneuver it over to a target location, detach from the satellite, and then maneuver away from the payload. Crew interaction is required to attach and detach the satellites due to the limitations of the Velcro docking port. This test simulates the process of assembling a larger structure using a single tug to maneuver each payload. The payload in this test is an active payload, meaning the tug can use the thrusters on the payload satellite to help in the maneuvering. The test was conducted during the June 2009 test session, however due to a misplaced beacon which caused estimator issues, the results were inconclusive. The test will be repeated in the next test session, tentatively scheduled for late August 2009.

Conclusions and Future Work:

During autonomous tug-based on-orbit assembly, the tug undergoes large mass property variations as it attaches to and detaches from each payload. These properties include mass, inertia, center of mass, etc. The property variations require a reconfigurable control system to maintain performance throughout the assembly process. A framework for the implementation of reconfiguration into a baseline system is developed and implemented onto the SPHERES testbed. This paper describes the need for reconfigurable control systems, previous literature on the topic, the development of a framework for reconfiguration, and the description of the reconfiguration implementation. The components of this design are described: the reconfigurable controller, control allocation algorithm, estimator, and model generation algorithm. Hardware results for the reconfigurable estimator are presented. Initial results indicate successful performance. The estimator converges both in static and dynamics cases, and has sufficient accuracy for control (± 1 cm). The reconfigurable estimator shows better velocity performance than the baseline. Also, criteria for incorporating additional sensor measurements are presented, as well as demonstration in simulation of when the incorporating can lead to an improvement in the estimator. The methodology and framework are general and applicable to many aerospace systems, while the implementation can be used as a guide to implement reconfiguration into other similar systems. The successful implementation of all of the elements

into an integrated assembly test will validate the framework and methodology. The integrated test is currently waiting execution on the ISS.

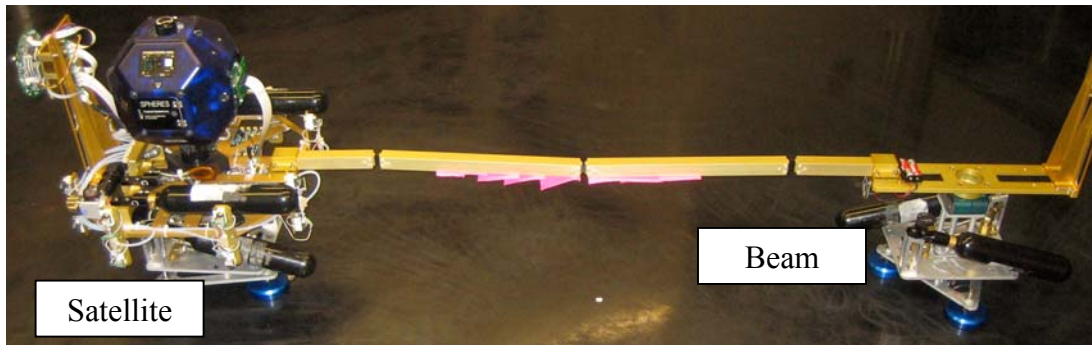


Figure 13: SWARM Hardware testbed

Future work involves expanding this methodology for flexible structures. Applications that motivate on-orbit assembly include flexible payloads, such as solar arrays, large optics, and flexible structures. It is critical to minimize vibrations so as not to damage the payload. This poses challenges as the tug's mass and stiffness properties change at each docking. Accounting for flexibility requires advanced controllers (such as adaptive control) and a refined model update algorithm that can determine mode shapes from the information of the flexible appendage. The information of the mode shapes can be used to design the controller and control allocation algorithms to minimize vibrations.

Hardware testing of reconfigurable control systems for flexible appendages will be done using the SWARM (Self-Assembling Wireless Autonomous Reconfigurable Modules) testbed. The SWARM testbed consists of a SPHERES satellite attached to a flexible beam, with a docking port on the end of the flexible beam (Fig. 13). The objective is to achieve control to enabling docking at the end of the flexible beam. The completion of the reconfigurable control system design for flexible payload will round out the methodology, and enable a base design that can be adapted to a variety of situations.

References:

- [1] Parlos, A. G. and J. W. Sunkel. Adaptive attitude stability and control for space station / orbiter berthing operations. 1992.
- [2] Maybeck P. and R. Stevens. "Reconfigurable flight control via multiple model adaptive control methods." In *IEEE Transactions on Aerospace and Electronic Systems*, volume 27, 1991.
- [3] E. Wilson, C. Lages, and R. Mah. "On-line, gyro-based, mass-property identification for thruster-controlled spacecraft using recursive least squares." In *The 2002 45th Midwest Symposium on Circuits and Systems*, volume 2, pages 334–337. IEEE, August 2002.
- [4] Hess R. and C. McLean. "Development of a design methodology for reconfigurable flight control systems." In *38th Aerospace Sciences Meeting*, 2000.
- [5] Mohan S. "SPHERES Reconfigurable Control Allocation for Autonomous Assembly," *AIAA Guidance, Navigation, and Control Conference*, Honolulu HI, Aug 2008.
- [6] Saenz-Otero, A. & Miller, D.W. "SPHERES: a platform for formation flight research". *UV/Optical/IR Space Telescope: Innovative Technologies and Concepts II conference*, 2005.

- [7] Nolet, S. "The SPHERES Navigation System: from Early Development to On-Orbit Testing," *AIAA Guidance, Navigation & Control Conference*, Hilton Head SC, Aug 2007.
- [8] Mohan, S. "Reconfiguration Methods for On-orbit Servicing, Assembly, and Operations with Application to Space Telescopes," MS Thesis. Massachusetts Institute of Technology. SSL #24-07. 2007.