Hybrid Control for Autonomous Spacecraft Rendezvous Proximity Operations and Docking

Jason R. Crane*, Christopher W. T. Roscoe*, Bharani P. Malladi, Giulia Zucchini***, Eric Butcher**, Ricardo G. Sanfelice***, Islam I. Hussein***

**Applied Defense Solutions, Columbia, MD 21044 USA*
(Tel: 410-715-0005; e-mail: croscoe@applieddefense.com).
***University of Arizona, Tucson, AZ 85721 USA*
(e-mail: ebutcher@email.arizona.edu)
**** University of California Santa Cruz, Santa Cruz, CA 95064 USA*
(e-mail: ricardo@ucsc.edu)

Abstract: A hybrid control methodology is presented for autonomous rendezvous, proximity operations and docking of a pair of spacecraft. For the theoretical development of the control algorithms, the dynamics of the spacecraft are modeled using the Clohessy-Wiltshire-Hill equations, which result in a linear system of relative motion equations. Only in-plane motion is considered, resulting in a two-dimensional system, and the control input is the acceleration vector of the active spacecraft, constrained by a maximum thrust value. Individual controllers are designed for different phases of the of approach and transitions are governed by a hybrid supervising algorithm. The hybrid control algorithm is implemented both in MATLAB, using a simplified dynamic model, as well as in actual spacecraft flight code and tested in a high-fidelity spacecraft simulation test environment.

Keywords: hybrid systems, spacecraft autonomy, satellite control, aerospace control, Clohessy-Wiltshire-Hill equations

1. INTRODUCTION

The spacecraft Rendezvous, Proximity Operations and Docking (RPOD) mission has been actively studied going back to and before the days of the NASA's Gemini program. Missions include human and cargo transport, satellite repair, refueling, inspection, anomaly root cause analysis, space debris disposal, and international agreement compliance monitoring. The proliferation of small satellites with ever greater sensor and computational capability has opened the possibility of robustly performing these operations with small satellites without the need for human-in-the-loop control methodologies.

In this paper, we implement and validate the hybrid control algorithm proposed by Malladi et al. (2016), to solve the problem introduced by Jewison and Erwin (2016) pertaining to rendezvous, proximity operations, and docking of an autonomous spacecraft with equations of motion given in terms of the 2D Clohessy-Wiltshire-Hill (CWH) equations (Clohessy and Wiltshire, 1960; Hill, 1878). The algorithm developed by Malladi et al. (2016) conveniently supervises individual algorithms to cope with the individual phases of the problem: 1) rendezvous with angles-only measurements; 2) rendezvous with range measurements; 3) docking phase; and 4) docked phase. In light of the different constraints, available measurements, and tasks to perform on each phase, the problem requires a hybrid systems approach, in which the system has different modes of operation for which a suitable controller is to be designed. The results by Malladi et al. (2016) are general in the sense as they characterize a family of individual controllers and the required properties to solve the problem within each phase of operation. This paper provides specific controller designs that appropriately solve the control problems for individual phases and, when properly coordinated by a hybrid supervisor, solves the entire mission. The main contribution of this paper is the implementation of this hybrid controller within a real spacecraft guidance, navigation, and control (GNC) flight software (FSW) system. The resulting hybrid GNC FSW system is tested and validated in a high-fidelity software-in-the-loop (SITL) simulation environment.

2. PROBLEM DESCRIPTION

We consider a model of the relative motion of the chaser spacecraft with respect to a resident space object (RSO) given by the in-plane Clohessy-Wiltshire equations, namely,

$$\begin{aligned} \ddot{x} - 2n\dot{y} - 3n^2 x &= \frac{F_x}{m_c} \\ \ddot{y} + 2n\dot{x} &= \frac{F_y}{m_c} \end{aligned} \tag{1}$$

Where $(x, y) \in \mathbb{R}^2$ and $(\dot{x}, \dot{y}) \in \mathbb{R}^2$ are the planar position and velocity, respectively, $F_x \in \mathbb{R}$ and $F_y \in \mathbb{R}$ are the control forces in the x and y directions, respectively, m_c the mass of the chaser, and $n \coloneqq \sqrt{\frac{\mu}{r_0^3}}$ where μ is the gravitational parameter

of the Earth and r_o is the orbit radius of the RSO spacecraft. The RSO spacecraft is located at $(x, y) = (0,0)$ and has mass m_t. The state space representation of (1) is given by

$$\dot{\eta} = A\eta + Bu \tag{2}$$

where $\eta \coloneqq [x\ y\ \dot{x}\ \dot{y}]^{\mathrm{T}} \in \mathbb{R}^4$ is the state vector, $u \coloneqq [F_x\ F_y]^{\mathrm{T}} \in \mathbb{R}^2$ is the input vector, and

$$A \coloneqq \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix}, \quad B \coloneqq \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_c} & 0 \\ 0 & \frac{1}{m_c} \end{bmatrix}$$

are the state and input matrices, respectively. The relative position between the chaser and the RSO is represented by $\rho(x, y) \coloneqq \sqrt{x^2 + y^2}$.

With this linear spacecraft model, we intend to design a controller that steers chaser spacecraft towards the RSO and docks with the RSO. The design goal for the controller is summarized in the following problem statement, which comes directly from Malladi et al. (2016) and states the problem by Jewison and Erwin (2016) in the context of our hybrid control theoretical approach.

Problem 1: Given positive constants m_c, m_t, μ, r_o, u_{max}, $\rho_{max} > \rho_r > \rho_d$, $\bar{V}$, V_{max}, σ_1, σ_2, σ_3, σ_4, and $t_f > t_e$, a constant $\theta \in \left[0, \frac{\pi}{2}\right)$ and a point $(x_p, y_p) \in \mathbb{R} \times \mathbb{R}$, design a feedback controller that measures

$$y = h(\eta) + v$$

and assigns u such that for every initial condition

$$\eta_0 \in \mathcal{M}_0 \coloneqq \{\eta \in \mathbb{R}^4 : \rho(x, y) \in [0, \rho_{max}], \rho(\dot{x}, \dot{y}) \in [0, \bar{V}]\}$$

of the chaser with dynamics as in (2) under the constraints

- The control signal $t \mapsto u(t)$ satisfies the "maximum thrust" constraint
  $\sup_{t \geq 0} max\{|F_x(t)|, |F_y(t)|\} \leq u_{max}$
  namely,
  $u \in \mathcal{U}_P \coloneqq \{u \in \mathbb{R}^2 : max\{|F_x|, |F_y|\} \leq u_{max}\}$ (3)
- For each
  $\eta \in \mathcal{M}_1 \coloneqq \{\eta \in \mathbb{R}^4 : \rho(x, y) \in [\rho_r, \infty)\}$,
  only angle measurements are available, namely,
  $h(\eta) = arctan\left(\frac{y}{x}\right)$
  where arctan : $\mathbb{R} \to [-\pi, \pi]$ is the four-quadrant inverse tangent, and $v \in \mathcal{N}(0, \sigma_1^2)$;
- For each $\eta \in \mathcal{M}_2 \coloneqq \{\eta \in \mathbb{R}^4 : \rho(x, y) \in [\rho_d, \rho_r)\}$, angle and range measurements are available, namely,
  $$h(\eta) = \begin{bmatrix} \arctan\left(\frac{y}{x}\right) \\ \sqrt{x^2 + y^2} \end{bmatrix} \tag{4}$$
  and $v \in \mathcal{N}^2(0, \sigma_2^2)$;
- For each $\eta \in \mathcal{M}_3^a \coloneqq \{\eta \in \mathbb{R}^4 : \rho(x, y) \in [0, \rho_d)\}$, angle and range measurements are available, that is, we have h as in (4) and $v \in \mathcal{N}^2(0, \sigma_3^2)$ while, in addition, if $\eta \in \mathcal{M}_3^a \cap \mathcal{M}_3^b$, where
  $$\mathcal{M}_3^b(\theta) \coloneqq \left\{\eta \in \mathbb{R}^4 : \begin{bmatrix} \sin\left(\theta/2\right) & \cos\left(\theta/2\right) \\ \sin\left(\theta/2\right) & -\cos\left(\theta/2\right) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right\}$$
  namely, the state is in a cone with aperture θ centered about the x axis, then the following closing/approaching velocity constraint is satisfied
  $$\eta \in \mathcal{M}_3^c \coloneqq \left\{\eta \in \mathbb{R}^4 : -\left\langle \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \begin{bmatrix} x \\ y \end{bmatrix} \right\rangle \leq V_{max}\rho(x, y)\right\}$$

and, when the chaser docks to the RSO, chaser-RSO dynamics as in (2) with $m_c + m_t$ in place of m_c under the constraint (3) and with position measurements relative to a partner at location (x_p, y_p) are available, namely,

$$h(\eta) = \begin{bmatrix} arctan\left(\frac{r_x(x)}{r_y(y)}\right) \\ \sqrt{r_x(x)^2 + r_y(y)^2} \end{bmatrix}, r_x(x) = x - x_p, r_y(y) = y - y_p$$

And $v \in \mathcal{N}^2(0, \sigma_4^2)$ the following holds for the η-component $t \mapsto \eta(t)$ of each solution to the closed-loop system: for some $t_{2f} < t_{3f} < t_{4f}$ such that $t_{3f} \leq t_e, t_{4f} \leq t_f$ we have

1. $\eta(t_{2f}) \in \mathcal{M}_3^a \cap \mathcal{M}_3^b$ and $\rho\left(x(t_{2f}), y(t_{2f})\right) = \rho_d$; namely, the chaser reaches the cone first;
2. $\eta(t_{3f}) \in \mathcal{M}_3^c = \{\eta \in \mathbb{R}^4 : \eta = 0\}$; namely, the chaser docks on the RSO next, no later than t_{3f} time units;
3. $\eta(t_{4f}) \in \mathcal{M}_4$, where
   $\mathcal{M}_4 \coloneqq \{\eta \in \mathbb{R}^4 : |x - x_p| = 0, |y - y_p| = 0, \dot{x} = \dot{y} = 0\}$; namely, the docked chaser (or chaser-RSO) reach the partner location no later than t_{4f} time units.

3. OUTLINE OF PROPOSED HYBRID CONTROL SOLUTION

To solve Problem 1, Malladi et al. (2016) proposes a hybrid control algorithm that supervises multiple hybrid controllers. The individual controllers are designed to cope with the individual constraints and to satisfy the desired temporal properties. The supervising algorithm, which is also hybrid and denoted as $\mathcal{H}_s$, is in charge of supervising the following individual hybrid controllers:

- Hybrid controller for rendezvous from distances far from RSO (Phase I): the goal of this controller, denoted $\mathcal{H}_{C,1}$, is to steer the chaser to a point in the interior of $\mathcal{M}$, in particular from points in $\mathcal{M}_1 \cap \mathcal{M}_0$.

- Hybrid controller for rendezvous in close-proximity to RSO (Phase II): the goal of this controller, denoted $\mathcal{H}_{C,2}$, is to steer the chaser to a point in the interior of X_{los}, in particular, from points in $\mathcal{M}_2$.
- Hybrid controller for docking to RSO (Phase III): the goal of this controller, denoted $\mathcal{H}_{C,3}$, is to steer the chaser to nearby $\eta = 0$ from points in $\mathcal{M}_2 \cup \mathcal{M}_3^a$.
- Hybrid controller for relocation of RSO (Phase IV): the goal of this controller, denoted $\mathcal{H}_{C,4}$, is to steer the chaser and RSO to nearby the partner position (x_p, y_p) from $\mathcal{M}_3^c$.

The tasks performed by the individual controllers described above are also subjected to the constraints stated in Problem 1. In the next section, we present the design details of each of the controllers, see also Malladi et al. (2016).

4. DESIGN FOR NOMINAL CASE

To design the controllers for individual phases, we start by considering that the full state is estimated in each phase by an appropriate estimator. Each estimator considers the available measurements as described in the problem definition and reconstructs, full state, which is then used in the feedback design. Initial design associated with these controllers is presented in Malladi et al. (2016).

4.1 A LQR-based design of $\mathcal{H}_{C,1}$

The controller $\mathcal{H}_{C,1}$ is designed such that the chaser reaches Phase II in finite time. Due to relative distance of the chaser spacecraft from the RSO, only angle $\alpha \in \mathbb{R}$ is available in Phase I. Hence (2) is unobservable or the position (x, y) and velocity $(\dot{x}, \dot{y})$ cannot be reconstructed from the angle measurements. Kaufman et al. (2016) and Butcher et al. (2017) have shown that considering the full nonlinear dynamics of the spacecraft will result in improved state estimation in otherwise weakly observable or unobservable linear models. Therefore, quadratic terms of the gravity field are included to attain observability. Precisely, following Alfriend et al. (2009), we feedforward the term

$$\Gamma(\eta) \coloneqq \frac{\mu}{r_0^4}\left[0 \quad 0 \quad -3x^2 + \frac{3}{2}y^2 \quad 3xy\right]^{\mathrm{T}}$$

into the plant and implement a sequential Kalman filter on the resulting plant and design an LQR feedback controller. To overcome the discontinuities associated with angle calculations, we embed the angle on a unit circle, in other words we consider line of sight (LOS) measurements given by, $y_a = h(\eta) \coloneqq \left[\frac{x}{\rho(x,y)} \quad \frac{y}{\rho(x,y)}\right]^{\mathrm{T}}$. In this way, $\mathcal{H}_{C,1}$ is an observer-based (Kalman) LQR controller.

4.2 A logic-based line-of-sight controller of $\mathcal{H}_{C,2}$

The chaser spacecraft is relatively close to the RSO spacecraft, and hence both angle ($\alpha \in \mathbb{R}$) and range ($\rho \in \mathbb{R}$) are available in Phase II. Therefore, we consider the position and velocity of the chaser relative to the RSO frame are available for feedback. Once in Phase II, a continuous PD feedback control law that guides the chaser to dock with the RSO at a desired approaching direction and speed as described by Kluever (1999) is implemented. Since the objective for this phase is to complete the rendezvous maneuver along an arbitrary docking corridor defined by the position $\rho^* \in \mathbb{R}$ and docking direction, $\alpha^* \in \mathbb{R}$; a new polar frame fixed to the RSO spacecraft with its origin moving at a constant angular rate n is defined. By using a feedback linearization approach, line of sight and thrust acceleration controls in this new frame are defined. Using the initial condition of the chaser in Phase II, a continuous feedback law will result in unwinding and the chaser will sometimes take longer route to reach the desired line of sight. In such a scenario, a discontinuous controller $sgn(k_2(\eta, h))$, where 'sgn' is defined arbitrarily in the set $\{-1,1\}$ can be applied to the chaser nearby $\alpha = 0$. But such a discontinuous controller would not be robust to small measurement noise as previously shown by Sanfelice et al. (2006). Hence, a hysteresis based hybrid control approach is implemented so that the chaser can rotate either clock-wise or counter clock-wise to take shortest route and reach a desired location in the line of sight cone of Phase III and be robust to small perturbations.

4.3 A uniting local and "global" design of $\mathcal{H}_{C,3}$

Similar to Phase II, we consider the position and velocity is of the chaser relative to the RSO frame are available for feedback. Once the chaser is inside the line of sight cone of Phase III, a hybrid controller with local and global stabilizing capabilities is implemented so as to render forward invariance and satisfy closing speed constraints for the chaser (Sanfelice and Prieur, 2013). Here, the first hybrid controller $k_3^1: \mathbb{R}^4 \to \mathbb{R}^2$, thrusts the chaser towards the some reference way-point $\eta_r \coloneqq [x_r \quad 0 \quad 0 \quad 0]^T$ inside the line of sight cone in T_{3a} seconds to guarantee forward invariance; while, the second hybrid controller $k_3^2: \mathbb{R}^4 \to \mathbb{R}^2$, implements a damping control that guides the chaser from η_r location towards the RSO within T_{3b} seconds, thus slowing down the vehicle and satisfying the closing constraint.

4.4 Controller for Regulation $\mathcal{H}_{C,4}$

In this problem, we design a controller that moves the chaser-RSO to a relocation position. Once again we consider the position and velocity of the chaser relative to the RSO frame are also available for feedback. Based on relocation position, a LQR controller with linear feedback $k: \mathbb{R}^4 \to \mathbb{R}^2$ given by

$$k(\eta) \coloneqq -K\eta$$

where $K \in \mathbb{R}^{2\times4}$, is implemented to move the chaser close to the RSO spacecraft. The gain K is designed to satisfy the additional constraint on the maximum thrust, namely, $\|u\|_\infty \leq u_{max}$. The controller gain $K \in \mathbb{R}^{2\times4}$ is designed by choosing the corresponding weight matrices $Q \in \mathbb{R}^{4\times4}$, $R \in \mathbb{R}^{2\times4}$, in

LQR control, such that the closed-loop system matrix $(A - BK)$ is Hurwitz.

5. RESULTS

The supervisory hybrid control algorithm was initially implemented in MATLAB in order to validate the overall approach for solving the RPOD problem using a simplified dynamic model and spacecraft architecture. Next, the algorithm was implemented in a real spacecraft GNC FSW system and tested in a high-fidelity simulation environment. The following sections describe the results of these two simulations.

5.1 In-Plane CWH MATLAB Simulation

The MATLAB simulation environment considers the two spacecraft to be point masses and models their relative motion simply using the 2-dimensional in-plane CWH equations, (1). The overall chaser trajectory resulting from the implementation of the supervisory hybrid controller is shown in Fig. 1, with the chaser initial position approximately 5 km behind the RSO. Fig. 2 shows the trajectories during the transitions between approach phases, highlighting transient behavior and successful achievement of the terminal guidance goal.

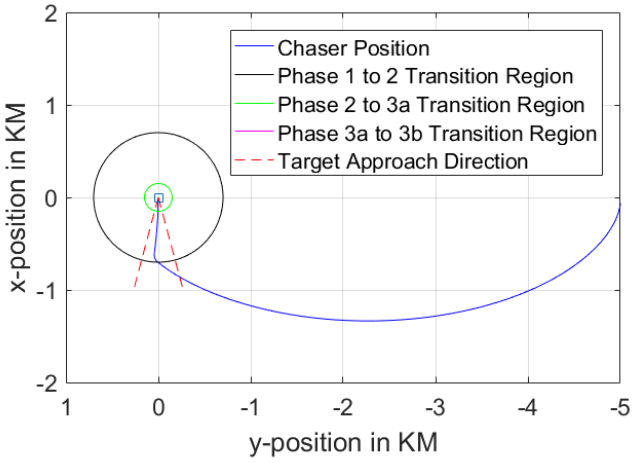

Fig. 1. Phase 1 through Phase 3b Supervisory Control trajectory in MATLAB testbed. Note by convention "x-position" is graphed along the y-axis here as it corresponds to 'up and down' with respect to the earth's surface.

5.2 High-Fidelity GNC FSW SITL Simulation

After initial testing with the simplified dynamic model, the supervisory hybrid control algorithm was implemented with a real spacecraft GNC FSW system. The GNC FSW was originally developed for the NASA CubeSat Proximity Operations Demonstration (CPOD) mission (Roscoe et al., 2015). The CPOD mission comprises two identical 3U CubeSats which will demonstrate RPOD using miniaturized spacecraft components and novel GNC FSW and algorithms to overcome the challenges associated with limited power, storage, and processing capability associated with small satellites employing COTS components.

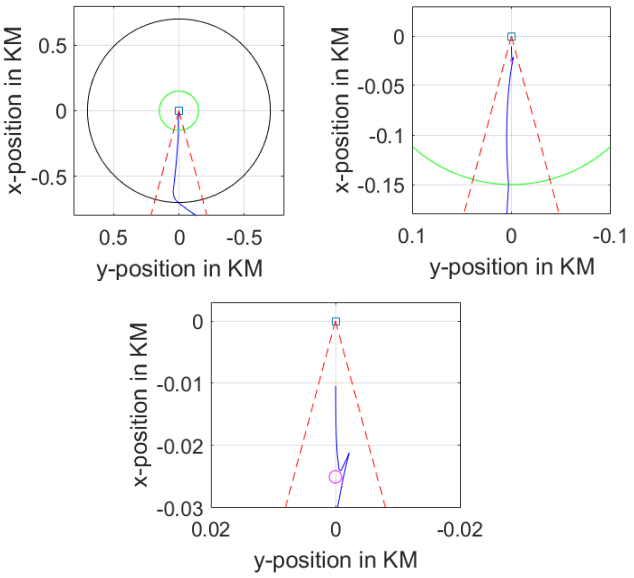

Fig. 2. Phase 1 through Phase 3b Supervisory Control trajectory in MATLAB testbed, zoomed in views. Phase jumps occur at the transition regions marked by black, green, and magenta circles. Testbed based on 2D CWH equations with gaussian noise added.

The supervisory hybrid control algorithm was rewritten in C++ and integrated into the existing GNC FSW using an object-oriented approach. For CPOD, multiple guidance modes were already defined for the various far-field and near-field phases of the mission, comparable to the different phases of the supervisory hybrid control algorithm. However, transition between guidance modes is handled through either ground commands or a sequence of pre-defined, rules-based transitions rather than based on the current state of the spacecraft (as in the supervisory hybrid control approach). In particular, no mathematical guarantees exist with respect to stability of the overall multi-mode guidance system, but rather performance under expected mission conditions is validated through extensive Monte Carlo testing and a combination of SITL and hardware-in-the-loop (HITL) testing (Roscoe et al., 2016).

The object-oriented design of the GNC FSW code makes implementing the new supervisory hybrid control algorithm straightforward. Fig. 3 shows the logical flow of the primary FSW GNC loop, with original implementation highlighted in blue and new supervisory hybrid control functions shown in green. The original "Guidance Manager" block contains the rules-based autonomous mode transitions as well as all control laws and maneuver computation algorithms for the various phases of the RPOD sequence. The new "Supervisory Guidance Manager" replaces the rules-based autonomous mode transitions with the supervisory hybrid controller defined in Section 3, as well as the individual controllers

defined in Section 4 (with the exception of the Phase IV controller, which was not evaluated in the present paper).

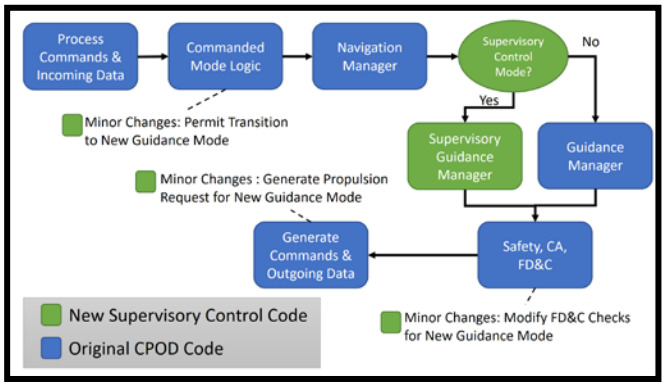

Fig. 3. A "Supervisory Guidance Manager" was created parallel to the default CPOD guidance manager. Several minor changes were applied to other parts of the code to accommodate the new control mode without throwing safety flags. Modular insertion allows for easier follow on supervisory control development.

Within the Supervisory Guidance Manager, the supervisory hybrid control Flow Set, Flow Map, Jump Set, and Jump Map are implemented such that the code structure and logic of the original GNC FSW are retained (Fig. 4). The Jump Set and Jump Map directly take the place of the original autonomous guidance mode transition logic and use the same state machine variables. The Flow Set and Flow Map directly take the place of the original Guidance Manager control update step, in which the current trajectory error is evaluated and control maneuvers are calculated. Maneuver commands are then generated and sent to the thruster control component in exactly the same fashion as in the original GNC FSW.

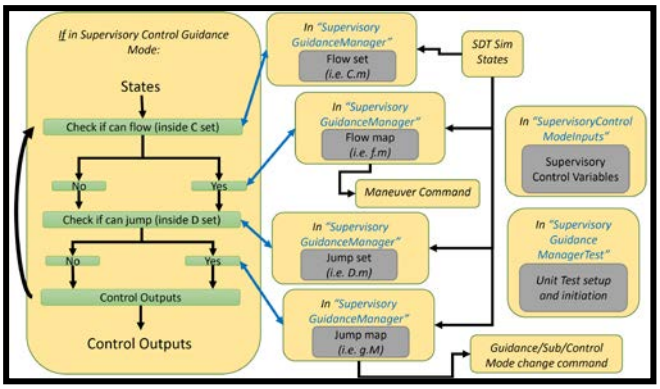

Fig. 4. Flow set, Flow map, Jump set, and Jump map were coded as separate functions within Supervisory Guidance Manager to closely match the MATLAB code as well as enable easier follow one development. The maneuver command interface formats remain the same as the default Guidance Manger. Supervisory Control Model Inputs and unit test files were added as separate modular scripts.

The Supervisory GNC (SGNC) FSW was then integrated and tested in a high-fidelity SITL simulation environment. The SITL testbed uses the ADS Spacecraft Design Tool (SDT) to provide a 6-degree of freedom (6DOF) dynamic model of the spacecraft including relevant orbital perturbations, physical environment effects, and individual hardware and software components. ADS uses SDT-SITL to test the GNC FSW in a faster-than-real-time flight-representative environment, where realistic messaging interfaces are used to interact with the system, and other spacecraft subsystems are emulated by SDT, including sensor components and actuators.

Starting with the same initial conditions (i.e. starting location for chaser relative to RSO) as in Section 5.1, a scenario was run in SDT-SITL using the new SGNC FSW implementation of the supervisory hybrid control algorithm (Fig. 5). The resulting trajectory is shown in Fig. 6, with phase transitions highlighted in Fig. 7.

Fig. 5. ADS's SDT-SITL test environment enables faster-than-real-time high-fidelity 6DOF spacecraft simulation for GNC FSW in a flight-representative environment.

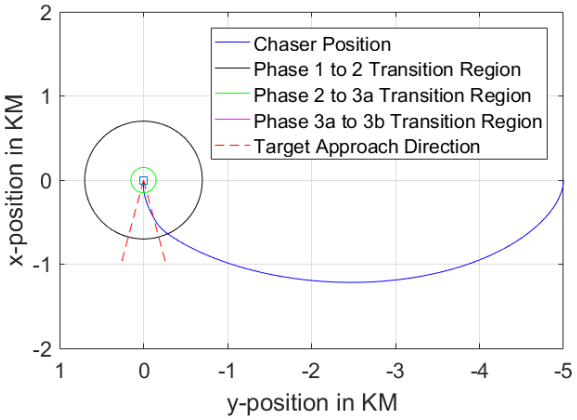

Fig. 6. Phase 1 through Phase 3b Supervisory Control trajectory in SDT-SITL testbed.

Comparing Fig. 6 and Fig. 7 to Fig. 1 and Fig. 2, the overall trajectories are similar, but the SDT-SITL results take a slightly different path relative to the in-plane CWH results. This difference is expected, since the in-plane CWH simulation uses only a simplified model of the relative motion whereas the GNC FSW uses a realistic spacecraft navigation system based on GPS, range, and relative bearing measurements. The SDT-SITL simulation includes 12x12 Earth gravity, atmospheric drag, and lunisolar perturbations.

In addition, actuator implementation in SDT-SITL uses a realistic thruster model which uses a series of finite burns to approximate a commanded impulse, whereas the CWH simulation implements acceleration commands directly. Finally, the CWH simulation implements continuous acceleration control whereas the GNC FSW uses delta-velocity commands on a 2-sec control cycle.

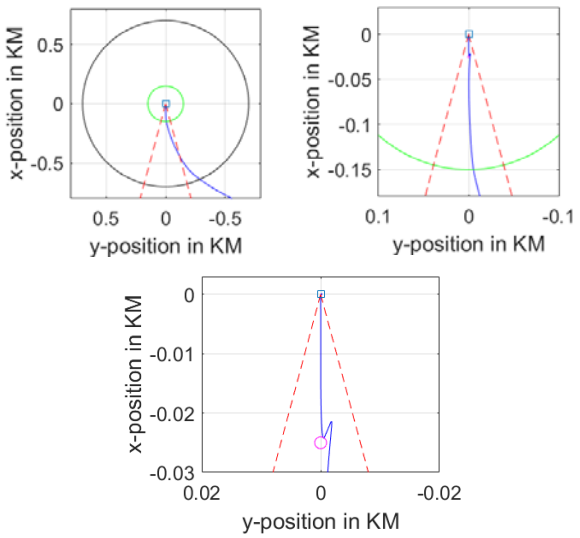

Fig. 7. Phase 1 through Phase 3b Supervisory Control trajectory in SDT-SITL testbed, zoomed in views. Phase jumps occur at the transition regions marked by black, green, and magenta circles.

Since the supervisory hybrid control algorithm only accounts for in-plane relative motion, a small uncontrolled out-of-plane error is also expected in the SDT-SITL results. The error should be small, since the spacecraft are initialized with no out-of-plane displacement and relative motion in that direction is decoupled from in-plane motion to first order in eccentricity. However, actuator errors during control thrusts and higher-order dynamic effects lead to small out-of-plane perturbations throughout the course of the simulation. The resulting out-of-plane motion for the SDT-SITL simulation is shown in Fig. 8.

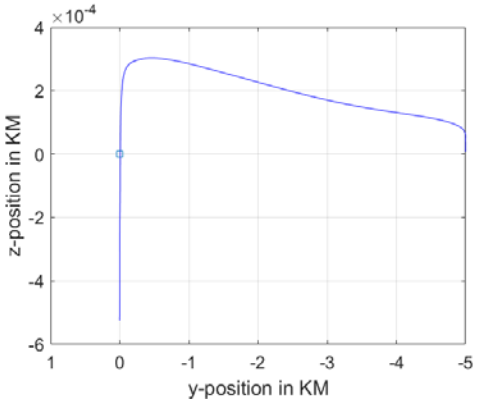

Fig. 8. Out-of-plane relative motion in SDT-SITL simulation.

6. CONCLUSIONS

A supervisory (hybrid) control scheme was designed for autonomous rendezvous, proximity operations and docking of a pair of spacecraft. The results of high-fidelity software-in-the-loop simulation in a real flight software implementation suggest hybrid control is a viable option for future spacecraft missions. Future work will include development of three-dimensional (including out-of-plane relative motion) and six-dimensional (including spacecraft attitude dynamics) supervisory hybrid control, as well as Monte-Carlo analysis of the algorithms to assess performance across varying initial conditions and state uncertainties.

REFERENCES

Alfriend, K., Rao, V., Gurfil P., How, J, & Breger, L., (2009). *Spacecraft formation flying: dynamics, control and navigation*. 2nd ed. Butterworth-Heinemann.

Butcher, E., Wang, J. & Lovell, T. (2017). On Kalman filtering and observability in nonlinear sequential relative orbit estimation. *Journal of Guidance, Control, and Dynamics*, 40(9), pp.2167-2182.

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

Hill, G. (1878). Researches in the lunar theory. *American Journal of Mathematics*, 1:5-26.

Jewison, C. & Erwin, R.S., (2016). A spacecraft benchmark problem for hybrid control and estimation. *2016 IEEE 55th Conference on Decision and Control (CDC)*.

Kaufman, E., Lovell, T.A. & Lee, T., 2016. Nonlinear observability for relative orbit determination with angles-only measurements. *The Journal of the Astronautical Sciences*, 63(1), pp.60–80.

Kluever, C. (1999). Feedback control for spacecraft rendezvous and docking. *Journal of Guidance, Control, and Dynamics*, 22(4), pp.609-611.

Malladi, B.P., Sanfelice, R.G., Butcher, E., Wang, J., (2016). Robust hybrid supervisory control for rendezvous and docking of a spacecraft. *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp.3325–3330.

Roscoe, C.W.T., Westphal, J.J., Lutz, S., & Bennett, T., (2015). Guidance, navigation, and control algorithms for CubeSat formation flying. *Advances in the Astronautical Sciences*, 154, pp.685–699.

Roscoe, C.W.T., Westphal, J.J., Shelton, C.T.., & Bowen, J.A., (2016). CubeSat proximity operations demonstration (CPOD) mission: end-to-end integration and mission simulation testing. *Advances in the Astronautical Sciences*, 156, pp.1925–1940.

Sanfelice, R.G., Messina, M.J., Tuna, S.E., Teel, A.R., (2006). Robust hybrid controllers for continuous-time systems with applications to obstacle avoidance and regulation to disconnected set of points. *2006 American Control Conference*, pp.3352–3357.

Sanfelice, R.G. & Prieur, C., 2013. Robust supervisory control for uniting two output-feedback hybrid controllers with different objectives. *Automatica*, 49(7), pp.1958–1969.