EPiC Series in Computing

Volume 48, 2017, Pages 20–32

ARCH17. 4th International Workshop on Applied
Verification of Continuous and Hybrid Systems

Verifying safety of an autonomous spacecraft rendezvous mission (Benchmark Proposal)

Nicole Chan and Sayan Mitra

Coordinated Science Laboratory,
University of Illinois at Urbana-Champaign
{nschan3,mitras}@illinois.edu

Abstract

A fundamental maneuver in autonomous space operations is known as rendezvous, where a spacecraft navigates to and approaches another spacecraft. In this case study, we present linear and nonlinear benchmark models of an active chaser spacecraft performing rendezvous toward a passive, orbiting target. The system is modeled as a hybrid automaton, where the chaser must adhere to different sets of constraints in each discrete mode. A switched LQR controller is designed accordingly to meet this collection of physical and geometric safety constraints, while maintaining liveness in navigating toward the target spacecraft. We extend this benchmark problem to check for passive safety, which is collision avoidance along a passive, propulsion-free trajectory that may be followed in the event of system failures. We show that existing hybrid verification tools like SpaceEx, C2E2, and our own implementation of a simulation-driven verification tool can robustly verify this system with respect to the requirements, and a variety of relevant initial conditions.

1 Introduction

The National Research Council published a report on NASA's roadmaps for 2011-2021, where 83 specific technologies were ranked with high-priority for enabling next-generation space missions [2]. One such technology is relative guidance algorithms, or the autonomous control of spacecraft to perform fundamental maneuvers on-orbit. The benchmark model of autonomous rendezvous, proximity operations, and docking (ARPOD) presented in [14] captures the essence of relative guidance problems in a generic, reusable scenario. This scenario and its components are essential in applications such as on-orbit transportation of personnel [22], resupply for personnel [19], assembly of space stations [24], and repair and refueling of spacecraft [10].

In particular, *rendezvous* refers to navigating a primary spacecraft towards a secondary free-flying object (which we fix to be a satellite) and maneuvering within close-proximity of the target without collision. This presents a challenging constrained, optimal control guidance problem. An overview of general state-of-the-art solutions to this particular problem is given in [20]. While such approaches have lend to some successes, recent examples suggest they are not fully mature technologies nor guaranteed to behave safely. For example, NASA's DART

G. Frehse and M. Althoff (eds.), ARCH17 (EPiC Series in Computing, vol. 48), pp. 20–32

spacecraft was designed to rendezvous with the MUBLCOM satellite [18]. In 2005, approximately 11 hours into a 24-hour mission, DART's propellant supply depleted due to excessive use of thrusters, and it began a mission abort sequence. In the process it collided with MUBLCOM; it met only 11 of 27 mission objectives, rendering the loss of a $110 million project. Several other incidents in [23] highlight the consequences of failures in space applications and demonstrate the need for more rigorous testing before deployment.

Although formal verification has played an important role in design and safety analysis of spacecraft hardware and software (see, for example [11] and the references therein), they have not been used for model-based design and system-level verification and validation.

In this paper, we present a suite of hybrid models for the rendezvous portion of the ARPOD mission from [14] that can serve as benchmarks for verification tools and serve as building-blocks for more complex missions. These models consist of either nonlinear orbital dynamics or linearized dynamics using the Clohessy-Wiltshire-Hill (CWH) equations [1]; a switched linear state-feedback controller; and mission goal of rendezvous either with or without a failure event. The rendezvous mission is subdivided into two parts, such that there is a set of constraints on the spacecraft's motion when its position is within close proximity of the target position and a relaxed subset of constraints that hold when the spacecraft is outside of this proximity range. The controller exhibits switching behavior to handle this change of constraints during the course of the rendezvous mission. In the event of a system failure, we propose the spacecraft change its mission from rendezvous to passive abort. A passive abort involves shutting off the vehicle's thrusters when any anomalies are detected in the system, and the only requirement here is to avoid collision with the target, which is also referred to as the *passive safety* property. A failure event is captured by some subinterval of time in our bounded mission time horizon in which the spacecraft will nondeterministically transition to the passive abort strategy.

We have successfully verified the requirements for most of these models using existing hybrid verification tools SpaceEx [9] and C2E2 [4, 6], and our MatLab implementation of a simulation-driven verification algorithm (SDVTool) for linear hybrid models applied to this particular rendezvous problem. SpaceEx and C2E2 employ different algorithms for computing reachability, and while both work with linear hybrid models, only C2E2's approach works with nonlinear models and was expected to verify all the models presented in this paper. We found that C2E2 did not perform well for the subset of models involving the passive abort sequence, and introduced SDVTool to improve C2E2's capabilities for checking linear systems. The reachability algorithm in SDVTool is presented in [5]. Overall, we believe that our results and approaches establish feasibility of system-level verification of autonomous space operations, and they provide a foundation for the analysis of more sophisticated maneuvers in the future.

2 Related work

There are few academic works on system-level verification of autonomous spacecraft. A survey of general verification approaches and how they may apply to small satellite systems is presented in [13]. Architecture and Analysis Design Language (AADL) and verification and validation (V&V) over AADL models for satellite systems have been reported in [3]

An feasibility study for applying formal verification of autonomous satellite maneuvers is presented in [16]. That approach relied on creating rectangular abstractions (dynamics of the form $\dot{x} \in [a, b]$) of the satellites dynamics through hybridization and verification using PHAVer [8] and SpaceEx [9]. The generated abstract models have simple dynamics but hundreds of locations, and also, the analysis is necessarily conservative. In contrast, the approaches presented in this paper work directly with the linear (nonlinear) hybrid dynamics.

The ARPOD challenge [14] has been taken up by several researchers in proposing a variety of control strategies. A two-stage optimal control strategy is developed in [7], where the first part involves trajectory planning under a differentially-flat system and the second part implements Model Predictive Control on a linearized model. A supervisor is introduced to robustly coordinate a family of hybrid controllers in [17]. Safe reachsets (i.e. ReachAvoid sets) are computed for the ARPOD mission in [12] and used to solve for minimum fuel and minimum time trajectories.

3 Spacecraft Rendezvous Model

In this section, we present the detailed development of the hybrid models. First we present the orbital dynamics of the spacecraft in Sections 3.1-3.2. Then in Sections 3.3-3.4 we present a hybrid controller. Finally, we state the various mission constraints in Section 3.5.

3.1 Nonlinear relative motion dynamics

The dynamics of the two spacecraft in orbit—the *target* and the *chaser*—are derived from Kepler's laws. We use the simplest case for relative motion in space, where the two spacecraft are restricted to the same orbital plane, resulting in two-dimensional, planar motion. The so called Hill's relative coordinate frame is used. As shown in Figure 2, Hill's frame is centered on the target spacecraft, with $+\hat{\mathbf{i}}$-direction pointing radially outward from the Earth, $+\hat{\mathbf{k}}$-direction normal from the orbital plane, and $+\hat{\mathbf{j}}$-direction completing a right-handed system. We further assume that the target moves on a circular orbit, and thus, the $\hat{\mathbf{j}}$-direction aligns with the tangential velocity of the target.

The restriction on the target's orbit implies that the target-centered frame rotates with constant angular velocity. We will assume the target is in geostationary equatorial orbit (GEO), so its angular velocity is $n = \sqrt{\frac{\mu}{r^3}}$, where $\mu = 3.986 \times 10^{14} m^3/s^2$ and $r = 42164km$. The chaser's position is represented by the vector $\vec{\rho} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}}$, and the acceleration provided by the chaser's thrusters is denoted $\vec{u} = F_x\mathbf{i} + F_y\mathbf{j}$. The following equations are derived using Kepler's laws and constitute the nonlinear model of the spacecraft dynamics.

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

where $r_c = \sqrt{(r+x)^2 + y^2}$ is the distance between the chaser and Earth and $m_c = 500\text{kg}$ is the mass of the chaser.

3.2 Linear dynamics

Linearization of these equations about the system's equilibrium point at the origin results in the Clohessy-Wiltshire-Hill (CWH) equations [1], which are commonly used to capture the relative motion dynamics of two satellites within a reasonably close range. These equations are:

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

Let the state vector be denoted by $\vec{x} = [x, y, \dot{x}, \dot{y}]^T$. The state-space form of these linear time-invariant (LTI) equations is:

$$\dot{\vec{x}} = A\vec{x} + B\vec{u}, \text{ where,}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_c} & 0 \\ 0 & \frac{1}{m_c} \end{bmatrix}, \vec{u} = \begin{bmatrix} F_x \\ F_y \end{bmatrix}.$$

3.3 Hybrid system model

Varying ranges of the relative distance between the spacecraft give rise to different constraints and requirements, and therefore, require separate controllers. We will present a two-stage hybrid controller for achieving the rendezvous maneuver. Each discrete mode has an *invariant* which specifies the conditions under which the system may operate in that mode, which we will first describe in words.

Mode 1: the chaser is attempting to rendezvous and its separation distance ($\rho = \sqrt{x^2 + y^2}$) from the target is in the range 100-1000m.

Mode 2: the chaser is attempting to rendezvous and its separation distance is less than 100m.

Passive: the chaser is no longer attempting to rendezvous and is not using its thrusters, regardless of its separation distance.

The state of the overall hybrid system is defined by the mode and the valuations of a set of continuous variables: relative position x, y, relative velocity $\dot{x}, \dot{y}$, thrusts F_x, F_y, and a global timer *tmr*. There are two timing parameters of the model t_1 and t_2 that specify the time interval over which the chaser spacecraft may enter the Passive mode. When the system is in a particular mode, the continuous variables $\vec{x}$ evolve according to the (linear (2) or nonlinear (1)) differential equations of the previous section. The thrust inputs $\vec{u}$ are computed according the full-state feedback controller designed in Section 3.4.

We refer to the time elapsed in the mission with the variable *tmr*. In dynamical systems, the time parameter t is independent and not treated as a state variable, or the dependent term $\vec{x}$. However, transitions introduce nondeterminism, even in the independent variables. The transition to Passive is dependent on time, thus we explicitly model time by *tmr*.

A transition *may* be taken as long as the *guard* (i.e. label on the edges of Figure 1) is satisfied. When the mode invariant aligns with a guard so that the transition must be taken as soon as it is enabled, this is called an *urgent* transition. Such transitions may preserve determinism, in that we may know the precise state of the system just before a transition. On the other hand, the transitions to Passive are not urgent. Even if we know precisely the valuation of *tmr* before a transition is taken, we can only know that $tmr \in [t_1, t_2]$ right after a transition is taken.

3.4 Linear Quadratic Control

We present a full-state feedback controller, namely, a Linear Quadratic Regulator (LQR), to drive the chaser towards the target's position. LQR is a linear function of the state, of the form: $\vec{u} = K_i\vec{x}$, where i indicates the discrete mode. Since we turn off thrusters in Passive mode, $K_{passive} = 0$.

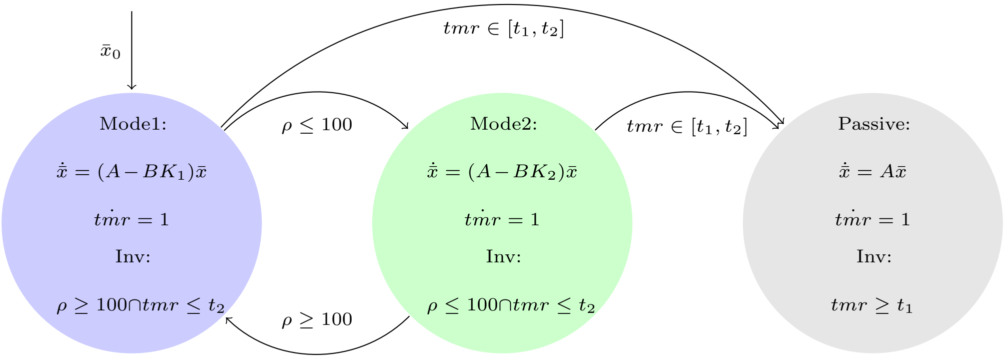

Figure 1: Hybrid model for spacecraft rendezvous, with linear flow equations shown. The invariants in Mode 1 and Mode 2 are defined exclusively by the chaser's position, as shown by corresponding colors in Figure 3. Transitions between Mode 1 and Mode 2 are *urgent*, while transitions to Passive occur nondeterministically within an interval of time.

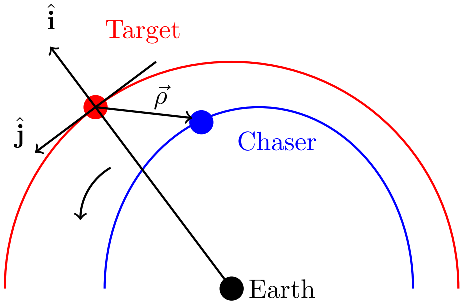

Figure 2: Hill's relative coordinate frame. Chaser's relative position vector is $\vec{\rho} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}}$. (Direction $+\hat{\mathbf{k}}$ out of the page not shown.)

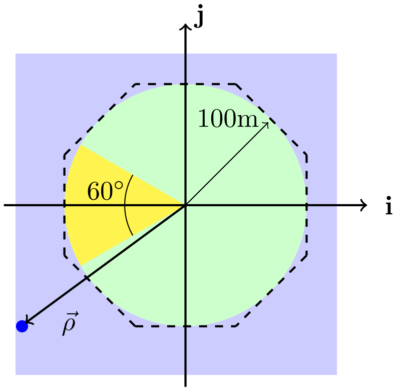

Figure 3: The invariants for Mode 1 and Mode 2 in Figure 1 are shown in corresponding colors (blue and green). The LOS region (yellow) overlaps with the invariant of Mode 2 and indicates the desired region of operation whenever the chaser is in Mode 2. The dashed lines show the octagon used to approximate $\rho = 100$ using only affine functions of x, y.

In general, closed-loop feedback control is desirable because the system can measure and adjust for errors, and ultimately guarantee liveness (i.e. eventually the target will be reached). LQR is specifically chosen because it is constructed by minimizing a quadratic cost function, which we can choose so as to roughly satisfy our safety constraints. LQR is only applicable to linear systems, so we design the control for the linearized model in (2), but we will use the same control with nonlinear dynamics (1) when applying verification tools.

Each constant $K_i \in \mathbb{R}^{4\times 2}$ is obtained by solving the following optimization problem:

$$\min_{\vec{u}} \int_0^\infty (\vec{x}^T Q_i \vec{x} + \vec{u}^T R_i \vec{u}) dt,$$

$$\text{subject to } \dot{\vec{x}} = A\vec{x} + B\vec{u},$$

where Q_i and R_i are positive definite matrices.

Bryson's method [15] is used to help determine an appropriate cost function. Begin with Q and R as diagonal matrices, and choose their values so as to normalize each of the state and input variables. In other words, choose the diagonal elements so that $q_{ii} = \frac{1}{max(x_i^2)}$ and $r_{ii} = \frac{1}{max(u_i^2)}$. Here, the denominators refer to the largest *desired* value of each variable, which will be determined by the safety constraints and mode invariants. While the LQR gains are obtained with our constraints in mind, the resulting controller does not guarantee these constraints are never violated. This is why further verification is still required. This design process is repeated for modes 1 and 2, and the result is two distinct LQR controllers for each of these modes in our hybrid system.

Notice that the flow equations given in Figure 1 already account for the form of the LQR controller in the context of the linear dynamics in (2). Similarly substituting the LQR solution into the nonlinear system in (1) results in the following nonlinear flow equations:

$$\begin{aligned} \ddot{x} &= \left(3n^2 - \frac{k_{11}}{m_c}\right) x - \frac{k_{12}}{m_c} y - \frac{k_{13}}{m_c} \dot{x} + \left(2n - \frac{k_{14}}{m_c}\right) \dot{y}, \\ \ddot{y} &= -\frac{k_{21}}{m_c} x - \frac{k_{22}}{m_c} y - \left(2n + \frac{k_{23}}{m_c}\right) \dot{x} - \frac{k_{24}}{m_c} \dot{y}. \end{aligned} \tag{3}$$

3.5 Constraints and safety requirements

In this section, we enumerate the properties that define a safe and successful mission, and how they are modeled for verification tools.

Thrust constraints During the rendezvous stages (Mode 1 and Mode 2), the thrusters cannot provide more than $10N$ of force in any single direction, therefore, we have the constraints:

$$|F_x|, |F_y| \leq 10.$$

LOS cone and proximity During close-range rendezvous Mode 2, the chaser must remain within a line-of-sight (LOS) cone (see Figure 3), and its total velocity must remain under 5cm/s, so $\sqrt{\dot{x}^2 + \dot{y}^2} \leq$ 5cm/s. The total velocity constraints cannot be exactly modeled using linear constraints, and a polytopic approximation over $\dot{x}, \dot{y}$ is used. This is done in the same way as $\sqrt{x^2 + y^2}$ is approximated (see Figure 3).

Separation During the Passive mode, the chaser must avoid collision with the target, which is theoretically a point mass at the origin. Even in a theoretical model, a small ball or box should be used to bound this point to account for limitations in numerical precision. In reality, the target satellite's dimensions may range from the order of 1m to 100m, so the size of this bounding box will vary depending on the situation. We use a box with a 0.1m circumradius.

4 Verification approaches

In this section, we briefly discuss our experience in using hybrid system verification tools. **SpaceEx** [9], is a well-established reachability analysis tool for linear and affine hybrid systems. It implements a support function-based reachability algorithm. The support function representation of sets is amenable to effective computation of convex hulls, linear transforms, Minkowski sums, etc.—operations that are necessary for safety verification.

C2E2 [4, 6] is a simulation-driven bounded verification tool for nonlinear hybrid models. The core algorithm of C2E2 relies on computing reachset over-approximations from validated numerical simulations and what are called *discrepancy functions.* A discrepancy function for a model bounds the sensitivity of the trajectories of the hybrid system to changes in initial states and inputs. Candidate discrepancy functions can be obtained using a global Lipschitz constant or using a matrix norm for linear systems. However, typically these approaches give discrepancy functions that blow-up exponentially with time, and therefore, are not useful for verifying problems with long time horizons. The automatic on-the-fly approach implemented in [6] uses bounds on the Jacobian matrix of the system to get tighter local discrepancy functions and it has been used to verify several benchmark problems. Recently the tool has been extended to handle nonlinear models with dynamics with exponential and trigonometric functions.

For a (possibly nonlinear) mode with $\dot{x} = f(x(t))$, the discrepancy computed by the algorithm of [6] uses the Jacobian matrix $J(x)$ of $f(x)$ and the condition number of $J(x_0)$ evaluated at certain points x_0 in the state space. For ill-conditioned matrices, such as what we have in the passive mode, (the A-matrix representation of (2)), the over-approximation error may still blow-up. Ill-conditioned systems may not only arise from passive dynamics but also from extremely large and small coefficients appearing together in $J(x_0)$.

In order to address this problem, we have created a MatLab implementation of C2E2's verification algorithm for linear models and then modified the reachability computation component. SDVTool does not rely on a discrepancy function, but instead computes the reachable states under a given linear mode directly. The particular algorithm implemented is the one presented in [5]: For an n-dimensional system, $n+1$ simulations are performed. From these simulations, special sets called *generalized star sets*, are generated to represent the exact reachsets. For our purposes, a generalized star set is represented by a pair $\langle x_0, V\rangle$, where $x_0 \in \mathbb{R}^n$ is the center state and $V = \{v_1, ..., v_n\} \subseteq \mathbb{R}^n$ is a standard basis (not necessarily unit vectors), and the set defined by $\langle x_0, V\rangle$ is

$$\{x \in \mathbb{R}^n \mid \exists \alpha_1, \ldots, \alpha_n \in [-1,1], x = x_0 + \sum_{i=1}^{n} \alpha_i v_i\}.$$

As reachsets are calculated for time steps, x_0 and V are transformed. When the reachtube from a given mode intersects the guards for a transition, the star sets are aggregated and over-approximated with hyperrectangles. If $R_i^* =< x(t_i), V_i >$ is the star set reachset obtained at time t_i, then the hyperrectangular reachset is:

$$R_i = \{x \mid x \leq x(t_i) + \sum_{j=1}^{n} max(-v_j, v_j)$$

$$\text{and } x \geq x(t_i) + \sum_{j=1}^{n} min(-v_j, v_j)\}.$$

C2E2 and SDVTool currently accumulates all the reachable sets in Mode 1 and Mode 2 that *may* transition to Passive, and uses their convex hull to begin reachset computations under the Passive mode. It follows that if the time interval during which a transition may occur is large, then the initial set of states under the Passive mode is large, making it very difficult to prove safety. One solution is to allow partitioning and refinement of the initial passive mode set. Since this is not currently implemented in C2E2 or SDVTool, we restrict our experiments to transition interval lengths of 5 minutes or less. For example, checking if the system is safe for a transition $tmr \in [50, 200 \text{ min}]$ could be achieved by running several experiments with small subintervals that cover the original interval.

5 Verification results

The collection of hybrid automaton models for each of the tools are available from from this link[1]. In this section, we will elaborate on how the hybrid model in Figure 1 with its linear (2) and nonlinear (1) variants are actually modeled in the semantics of each of the software tools.

First, all the tools operate on a bounded set of initial states and bounded execution time. Our choice of these parameters were motivated by the encompassing mission from [14], where an upper bound on the total mission time is given and an initial state is given for a mode that precedes Mode 1 in our setup. Since we are only performing a subset of the mission, we choose a time horizon strictly shorter than that bound and one that aligns with average rendezvous times outlined in [21]. Our time bound is 4 hours and the set of initial states is the hyperrectangle around the point $\vec{x}_0 = [-900m, -400m, 0m/s, 0m/s]$ with radius $[25m, 25m, 0, 0]$. This can be interpreted as uncertainty in the chaser's initial position but knowing it begins with zero initial relative velocity.

Next, in both C2E2 and SpaceEx, unsafe properties are modeled and checked individually. Furthermore each property must be described by a conjunction of affine inequalities. In Section 3.5, we described *safe* regions of operation. Some of these properties are represented by nonlinear inequalities (i.e. LOS region and total velocity) so we use polytopes to approximate such subsets of the state space. Since each property is 2-dimensional in this benchmark, their approximations are easily visualized. The LOS region (shown in Figure 3) is approximated by the following:

$$(x \geq -100) \cap (y \leq x \tan{(30^\circ)}) \cap (-y \leq x \tan{(30^\circ)}),$$

which describes a triangle circumscribing the true LOS region. On the other hand the safe total velocity region is originally: $\sqrt{\dot{x}^2 + \dot{y}^2} \leq 0.05$, and is approximated using a conjunction of affine inequalities that forms an octagon, similar to the dashed lines in Figure 3 approximating $\sqrt{x^2 + y^2} \leq 100$. Since safe sets are described by a conjunction of affine inequalities, we see that *unsafe* regions are described by a disjunction of affine inequalities. Then, each inequality is independently modeled in the software tool input files. For example, the representation of unsafe LOS regions spans three files/properties:

- LOS1: $x < -100$
- LOS2: $y > x \tan{(30^\circ)}$
- LOS3: $-y > x \tan{(30^\circ)}$

Notice that flow equations are modeled using differential equations. Observe that following systems describe equivalent behaviors:

[1] https://tinyurl.com/verifysat

1. $\dot{\vec{x}} = A\vec{x} + B\vec{u},\ \vec{u} = -K\vec{x}$
2. $\dot{\vec{x}} = (A - BK)\vec{x}$

However, only system 2 can be directly modeled in the syntax for hybrid model descriptions in C2E2 and SpaceEx. If any transition guards, mode invariants, or unsafe properties depend directly on $\vec{u}$, these would be captured by affine inequalities $-K\vec{x} < c$. However, the syntax of C2E2 is limited to 1- or 2-dimensional affine inequalities. Thus we revise system 1 to be: $\dot{\vec{x}} = A\vec{x} + B\vec{u},\ \dot{\vec{u}} = -K\dot{\vec{x}}$. This is equivalent to the original system 1 given the appropriate initial conditions. In the tool's syntax, this is a 6-dimensional state vector $[x, y, \dot{x}, \dot{y}, F_x, F_y]$. This 6-dimensional variant is required to verify thrust constraints in C2E2. While it is not required in SpaceEx, the 6-dimensional model may still be used in SpaceEx to obtain reach sets for the F_x, F_y variables.

The last parameter we have yet to define in the software tool models is the guard on transitions to the Passive mode. We choose a small interval at $[120, 125\text{min}]$, which ensures that the chaser will have operated in both Mode 1 and Mode 2 before transitioning to Passive.

Figure 4 shows a sample of the reach sets we obtain from SDVTool, C2E2, and SpaceEx, under the conditions discussed up to this point. Table 1 summarizes the runtimes taken to obtain these results. The linear models with and without a failure event were all successfully verified to be safe with respect to all properties presented. The nonlinear model without a failure event (i.e. no transition to Passive) was successfully verified to be safe in C2E2 with respect to rendezvous-related properties. In C2E2, both linear and nonlinear models that included the Passive mode resulted in an unknown output (neither safe nor unsafe). As mentioned in Section 4, C2E2's reachability algorithm exhibits a state space blow-up for this particular mode, thus reach sets will always intersect with unsafe regions even when no simulation trace can be found to intersect with unsafe regions.

Note that C2E2 and SpaceEx re-executes its verification algorithm for *each* property, so the runtime listed for a property, say LOS, is actually the average runtime to check each sub-property, e.g. LOS1, LOS2, and LOS3. SDVTool is handcrafted to verify this particular rendezvous problem (i.e. there is no specification for general models/properties), so it checks all properties in one execution.

Tool	**Max Thrust**	**LOS**	**Max Vel.**	**Collision**	**All**
SDVTool					5.5
SpaceEx	6.2	6.4	6.5	6.5	
C2E2	85	20	17.8	N/V	

Table 1: Time (seconds) taken to check each safety property across different tools. "N/V" indicates that the property was not successfully checked.

We run an additional experiment in SDVTool to get a qualitative understanding of how well the switched LQR controller works for the scenario of a rendezvous mission with a failure event. Based on the encompassing ARPOD mission in [14], we expect our rendezvous mission to begin at a separation distance around 1000m and at an angle in the third quadrant of the relative coordinate frame. We maintain the same assumptions as before that the initial velocities are zero. We partition this set of initial states Θ (see the colored arc in Figure 5) into smaller sets $\theta_1, ..., \theta_M$, for some $M \in \mathbb{Z}^+$, such that $\cup_i \theta_i = \Theta$. For each initial set θ_i, we repeat the verification algorithm N times. On the j^{th} run, the Passive transition guard is set to

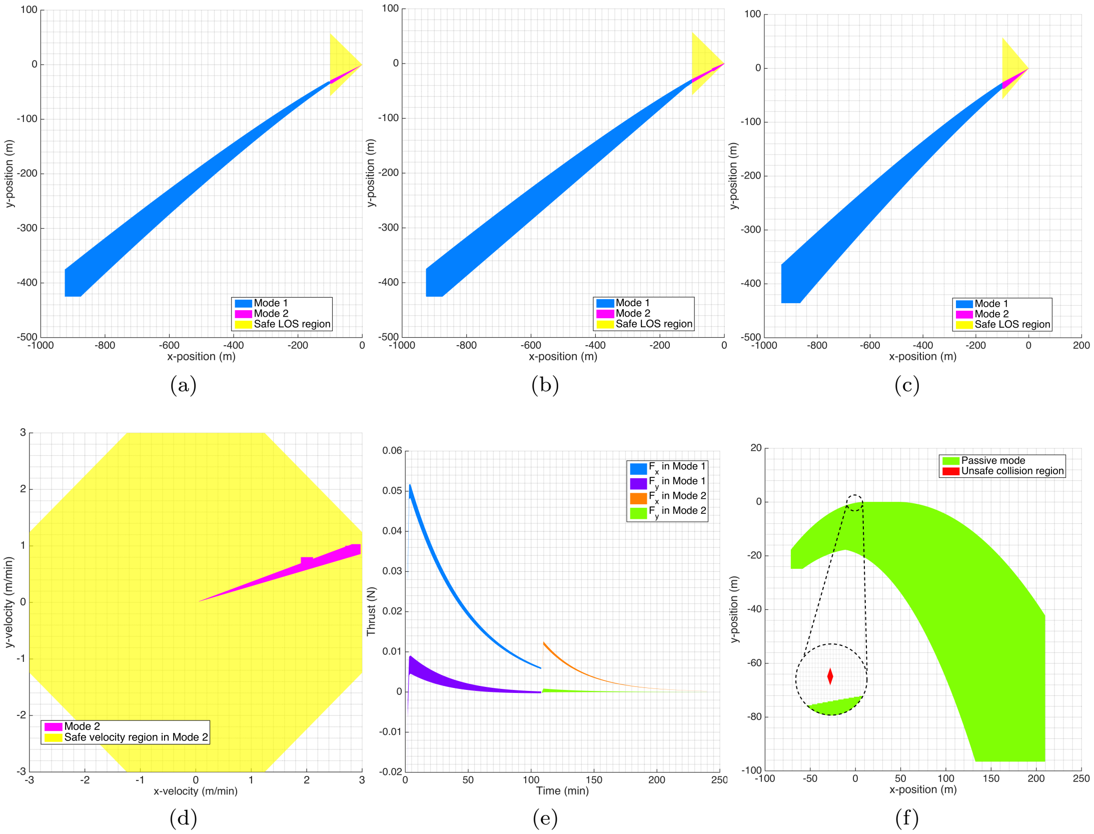

Figure 4: Reachable positions for rendezvous without any transition to Passive as computed by: (a) SDVTool (linear), (b) SpaceEx (linear), and (c) C2E2 (nonlinear). Reachsets computed by SDVTool for: (d) velocity in Mode 2, (e) thrust, and (f) positions reached after a transition to Passive during [120,125min].

$[t_1, t_2] = [t_{j-1}, t_j]$ and $t_0 = 0$, $t_N = 4$ hours. Figure 5 depicts the results of this experiment by assigning a color to the largest value of t_j for which the algorithm returns a safe result. For example, if we look at the initial state $[-900, -400, 0, 0]$ or a small neighborhood of this point, the color is a deep red indicating that if we perform rendezvous starting from this initial state, we can successfully rendezvous *and* abort the mission at any time before the 4 hour time bound. If we look at a neighborhood around $[-200, -900, 0, 0]$, the color is a light blue indicating that we can only guarantee that the rendezvous can safely abort at any time before 100 minutes. If a transition to Passive occurs after 100 minutes, this may result in a violation of the passive safety property *or* a rendezvous property (e.g. max thrust) may be violated after 100 minutes.

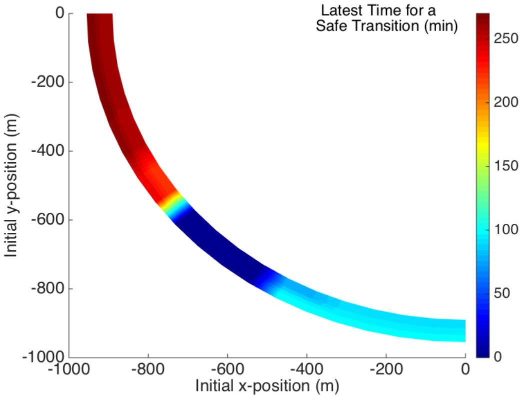

Figure 5: Initial positions (with zero initial velocities) of LinProx that have been verified to be safe. They are safe for Passive transition times up to the time shown by the color map.

6 Conclusions

In this case study paper, we present a sequence of linear and nonlinear, nondeterministic benchmark models of autonomous rendezvous between spacecraft with several physical and geometric safety requirements. We designed an LQR controller and verified its safety across the different models, a variety of initial conditions, parameter ranges, and using three different hybrid system verification approaches. The models and requirements are made available online.

This case study, and in particular the dynamics of the Passive mode, has shed light on the weakness of simulation-driven verification in handling ill-conditioned models.

The results provide a foundation for verifying more sophisticated maneuvers in future autonomous space operations. For example, we proposed a state feedback controller, but it is also possible to consider a situation where full state measurement is not possible and a simple bang-bang controller could be used instead. However a bang-bang controller would result in unstable behavior similar to the Passive mode in C2E2. The autonomous rendezvous problem can easily be made more challenging and realistic, which then invokes innovation and application of problems in observability, optimal and robust control, and the like. There is a concurrent need to address algorithmic verifiability of these sophisticated, innovative control solutions. C2E2 is one example of the ongoing effort to cover a large, growing class of nonlinear models, but it remains to be seen how we can apply rigorous safety-checking to common optimal, constrained control schemes, such as Model Predictive Control (e.g. if optimization constraints are relaxed.)

Acknowledgments

We are grateful for the support of Richard S. Erwin in navigating and modeling the problem presented in this paper, and for Yu Meng's support with the C2E2 experiments. We acknowledge the support of the Air Force Research Laboratory through the Space Scholars Program.

References

[1] Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, September 1960.

[2] *NASA Space Technology Roadmaps and Priorities.* The National Academies Press, May 2012.

[3] M. Bozzano, R. Cavada, A. Cimatti, J.-P. Katoen, V. Y. Nguyen, T. Noll, and X. Olive. Formal verification and validation of aadl models. *Proc. ERTS*, 2010.

[4] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok. C2E2: A verification tool for stateflow models. In *Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, pages 68–82, 2015.

[5] P. S. Duggirala and M. Viswanathan. Parsimonious, simulation based verification of linear systems. In *Computer Aided Verification*, pages 477–494. Springer Nature, 2016.

[6] C. Fan, B. Qi, S. Mitra, M. Viswanathan, and P. S. Duggirala. Automatic reachability analysis for nonlinear hybrid models with C2E2. In *Computer Aided Verification - 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part I*, pages 531–538, 2016.

[7] S. S. Farahani, I. Papusha, C. McGhan, and R. M. Murray. Constrained autonomous satellite docking via differential flatness and model predictive control. In *2016 IEEE 55th Conference on Decision and Control (CDC).* Institute of Electrical and Electronics Engineers (IEEE), December 2016.

[8] G. Frehse. PHAVer: Algorithmic verification of hybrid systems past HyTech. In *Hybrid Systems: Computation and Control*, pages 258–273. Springer Nature, 2005.

[9] G. Frehse, C. L. Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. Spaceex: Scalable verification of hybrid systems. In *CAV*, pages 379–395, 2011.

[10] K. Galabova, G. Bounova, O. de Weck, and D. Hastings. Architecting a family of space tugs based on orbital transfer mission scenarios. In *AIAA Space 2003 Conference & Exposition.* American Institute of Aeronautics and Astronautics (AIAA), September 2003.

[11] G. J. Holzmann. Mars code. *Commun. ACM*, 57(2):64–73, February 2014.

[12] B. HomChaudhuri, M. Oishi, M. Shubert, M. Baldwin, and R. S. Erwin. Computing reach-avoid sets for space vehicle docking under continuous thrust. In *2016 IEEE 55th Conference on Decision and Control (CDC).* Institute of Electrical and Electronics Engineers (IEEE), December 2016.

[13] S. A. Jacklin. Survey of verification and validation techniques for small satellite software development. Space tech expo, NASA Ames Research Center, May 2015.

[14] C. Jewison and R. S. Erwin. A spacecraft benchmark problem for hybrid control and estimation. In *2016 IEEE 55th Conference on Decision and Control (CDC).* Institute of Electrical and Electronics Engineers (IEEE), December 2016.

[15] M. A. Johnson and M. J. Grimble. Recent trends in linear optimal quadratic multivariable control system design. *IEE Proceedings D - Control Theory and Applications*, 134(1):53–71, January 1987.

[16] T. T. Johnson, J. Green, S. Mitra, R. Dudley, and R. S. Erwin. Satellite rendezvous and conjunction avoidance: Case studies in verification of nonlinear hybrid systems. In *FM 2012: Formal Methods*, pages 252–266. Springer Nature, 2012.

[17] B. P. Malladi, R. G. Sanfelice, E. Butcher, and J. Wang. Robust hybrid supervisory control for rendezvous and docking of a spacecraft. In *2016 IEEE 55th Conference on Decision and Control (CDC).* Institute of Electrical and Electronics Engineers (IEEE), December 2016.

[18] NASA. Overview of the dart mishap investigation results. Technical report, 2006.

[19] D. Pinard, S. Reynaud, P. Delpy, and S. E. Strandmoe. Accurate and autonomous navigation for the ATV. *Aerospace Science and Technology*, 11(6):490–498, September 2007.

[20] J. A. Starek, B. Açıkmeşe, I. A. Nesnas, and M. Pavone. *Spacecraft Autonomy Challenges for Next-Generation Space Missions*, pages 1–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[21] J. R. Wertz and R. Bell. Autonomous rendezvous and docking technologies - status and prospects. Technical report, SPIE AeroSense Symposium, April 2003.

[22] D. Woffinden and D. Geller. Navigating the road to autonomous orbital rendezvous. *Journal of Spacecraft and Rockets*, 44(4):898–909, 2007.

[23] W. E. Wong, V. Debroy, and A. Restrepo. The role of software in recent catastrophic accidents. Technical report, IEEE Reliability Society 2009 Annual Technology Report.

[24] D. Zimpfer, P. Kachmar, and S. Tuohy. Autonomous rendezvous, capture and in-space assembly: past, present and future. In *Proc. AIAA Space Exploration Conference*, January 2005.