Verified hybrid LQ control for autonomous spacecraft rendezvous

Nicole Chan and Sayan Mitra

Abstract— **Rendezvous is a fundamental maneuver in autonomous space operations in which an active *chaser* spacecraft is required to navigate safely to the proximity of a second passive *target* spacecraft. Ensuring safety of such control maneuvers is challenging and design errors can be expensive. We present the first verified control solution to a benchmark formulation of spacecraft autonomous rendezvous in the form of a hybrid LQR controller verified using a data-driven algorithm. Our hybrid LQR scheme is motivated by enforcing safety constraints rather than optimizing performance, and the control law is formulated by periodically solving optimization problems that depend on the current state. The resulting hybrid system presents a challenge for existing automated formal verification tools due to its lack of a closed-form model description. We overcome this challenge by using a data-driven approach (implemented in the new verification tool DryVR). DryVR relies on simulation traces to compute reachable states of the system over bounded time horizon and initial conditions to rigorously verify that the system does not violate any safety requirements.**

I. Introduction

A new age of deep space exploration is underway with several ongoing public-private partnerships. Autonomous operations where a spacecraft can operate independent of human intervention in a wide variety of conditions are essential for deployment, construction, and maintenance missions in deep space. Despite many spectacular successes like the Mars landing of the Curiosity rover, ensuring safety of autonomous spacecraft operations remains a daunting challenge and failures can be extremely expensive. For example, NASA's DART spacecraft was designed to rendezvous with the MUBLCOM satellite. In 2005, approximately 11 hours into a 24-hour mission, DART's propellant supply depleted due to excessive thrusting, and as it began maneuvers for departure, it collided with MUBLCOM. Most mission objectives were not met, and the failure resulted in a loss exceeding $1 million. In another incident, a navigation error caused the Mars Climate Orbiter to reach a low altitude of only 57 kms, instead of the intended 140-150 kms for entering orbit. The spacecraft was destroyed by the resulting atmospheric stresses and friction and the cost incurred was $85 million. These and several others [30] highlight the consequences of failures and the need for more rigorous verification and validation (V&V) before deployment.

Although formal verification has played an important role in design and safety analysis of spacecraft hardware and software (see, for example [16] and the references therein), they have not been used for model-based design and system-level V&V. In this paper, we present and verify a challenging spacecraft maneuver, the autonomous rendezvous problem. The original hybrid control design problem are introduced by Jewison and Erwin in [19], where rendezvous is only a part of an overacrching mission called autonomous rendezvous, proximity operations, and docking (ARPOD). ARPOD is a fundamental set of operations needed for a variety of space missions such as on-orbit transfer of personnel [29], resupply for on-orbit personnel [27], assembly [31], servicing, repair, and refueling [15].

A generic ARPOD scenario involves a passive module or a *target* (launched separately into orbit) and a *chaser* spacecraft that must transport the passive module to an on-orbit assembly location. The chaser maintains a relative bearing measurement to the target, but initially it may be too far away from the target to use its range sensors. Range measurements become available within a given range, giving the chaser accurate relative positioning data so that it can stage itself to dock the target. The target must be docked with a specific angle of approach and closing velocity, so as to avoid collision and ensure that the docking mechanisms on each body will mate. The controller in this paper is designed for the portion of the mission where the chaser must carefully approach the target, while maintaining knowledge of its exact relative position.

In this paper, we present a hybrid control scheme and apply it to the constrained, linear time-invariant benchmark rendezvous model given in [19]. We present a switched linear quadratic (LQ) control scheme, where a piecewise continuous feedback control law is periodically updated by recomputing a quadratic optimization problem that implicitly enforces system constraints. The optimization program itself is unconstrained, which may result in a more efficient and lightweight controller than existing optimization control schemes like Model Predictive Control (MPC). In fact, each subproblem takes the form of an infinite-horizon LQR problem, where at every periodic update, the LQR cost functional evolves according to a function of the current state. This function is responsible for implicitly capturing system constraints. We will refer to this control scheme as the *state-dependent LQ (SDLQ)* control. Formal verification algorithms are applied to the closed-loop system model to check that the particular control solution does indeed satisfy all constraints because the LQR formulations do not admit constraints.

The latter step is achieved using automated tools for simulation-driven reachability analysis. Such tools conveniently provide rigorous guarantees on all possible system behaviors starting from a bounded set of states over a finite

The authors are with the Coordinated Science Laboratory at the University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA {nschan3,mitras}@illinois.edu

time horizon. Until recently, we did not have the tools to perform this type of analysis on models that could not be precisely defined by closed-form mathematical models. The new verification tool DryVR [10] treats these complex systems as black boxes and employs learning techniques to infer properties of the black box, which are subsequently used to compute reachsets.

In this paper, we will present an example SDLQ controller for the autonomous rendezvous mission and demonstrate that it meets the requirements we design for using DryVR. We compare its expected performance with existing controllers proposed for rendezvous. The results of this paper are the first to demonstrate feasibility of system-level verification of autonomous space operations, and they provide a foundation for future analysis of more sophisticated spacecraft models.

II. Related work

A survey of system-level verification approaches and how they may apply to small satellite systems is presented in [18]. Architecture and Analysis Design Language (AADL) and verification and validation (V&V) over AADL models for satellite systems have been reported in [3]

A feasibility study for applying formal verification of autonomous satellite maneuvers is presented in [22]. That approach relied on creating rectangular abstractions (dynamics of the form $\dot{x} \in [a, b]$) of the system through hybridization and verification using PHAVer [13] and SpaceEx [14]. The generated abstract models have simple dynamics but hundreds of locations, and also, the analysis is necessarily conservative.

The ARPOD challenge [19] has been taken up by several researchers in proposing control strategies: A two-stage optimal control strategy is developed in [12], where the first part involves trajectory planning under a differentially-flat system and the second part implements MPC on a linearized model. A supervisor is introduced to robustly coordinate a family of hybrid controllers in [24]. Safe reachsets are computed for the ARPOD mission in [17] and used to solve for minimum fuel and minimum time trajectories, from which a control strategy could be inferred.

The notion of a state-dependent LQR function was first introduced in the state-dependent Riccati equations (SDRE) [26][28][25]. SDRE is a strategy for designing optimal feedback control (as well as observers and filters,) for nonlinear systems. The SDRE method is utilized for constrained discrete-time systems in [5], which provides a similar context to our constrained (continuous-time) problem. However to the best of our knowledge, state-dependent LQR methods have not been proposed for linear systems nor for the purpose of implicitly constraining solutions.

III. Spacecraft Rendezvous Model

In this section, we outline the system model, beginning with the uncontrolled spacecraft dynamics in Section III-A, and followed by the constraints in Section III-B. Both the structure of the spacecraft motion and mission constraints are derived from a much more generalized benchmark in [19]. Then, we present the hybrid control solution in Section IV.

A. Linearized dynamics

We assume the two spacecraft orbit in the same plane and that the target maintains a circular geostationary equatorial orbit (GEO). Hill's relative coordinate frame is used to describe the resulting two-dimensional, planar relative motion. The chaser's thrusters provide an external force or control input represented by the vector $\vec{u} = [F_x, F_y]^T$.

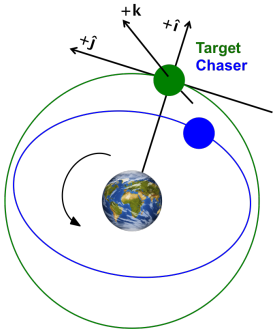


Fig. 1: Hill's relative coordinate frame. The chaser's relative position vector is $\vec{\rho} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}}$.

The uncontrolled dynamics are described by the Clohessy-Wiltshire-Hill (CWH) equations [2], which are commonly used to capture relative dynamics of two satellites within reasonably close range. Then the system dynamics are captured by the following linear time-invariant model:

$$\dot{\vec{x}} = A\vec{x} + B\vec{u}, \text{ where}$$
$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_c} & 0 \\ 0 & \frac{1}{m_c} \end{bmatrix}, \quad (1)$$

where $m_c = 500$kg is the mass of the chaser and $n = \sqrt{\frac{\mu}{r^3}}$ is the mean-motion parameter, given $\mu = 3.698 \times 10^{14} \mathrm{m}^3/\mathrm{s}^2$ and $r = 42164$km for GEO. Furthermore, we will denote and often refer to the separation distance between the two spacecraft by $\rho = \sqrt{x^2 + y^2}$.

The CWH equations ($\dot{\vec{x}} = A\vec{x}$) constitute the linear approximation of the chaser's Keplerian orbit under the gravitational two body problem (i.e. motion between two point masses governed only by a mutual gravitational force.)

B. Mission Safety and Progress

We will define a couple of safety properties motivated by physical phenomena that will be checked using a software verification tool discussed in Section V.

The maximal output that can be provided by the spacecraft's thrusters are modeled by a constraint on the control as follows:

$$|F_x|, |F_y| \leq 10\text{N}.$$

Sensors and the docking bay on the target vehicle are assumed to be positioned facing the Earth, so the chaser must approach the target along the $-\hat{\mathbf{i}}$ direction. This forms an acceptable region of operation in the plane of motion, called the *line-of-sight (LOS) region* (see Figure 2). This constraint

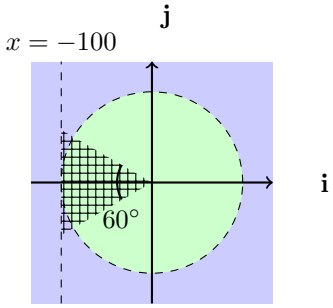


Fig. 2: The LOS region is gridded here. The blue and green regions correspond to the different modes in Figure 3a.

on the chaser's position is only required when it arrives within 100m of the target, and is specified by:

$$x \geq -100 \cap y \leq x \tan\left(\frac{\pi}{6}\right) \cap -y \leq x \tan\left(\frac{\pi}{6}\right).$$

Within the LOS region, relative velocity is further restricted to reduce impact forces when the vehicles dock, or:

$$\sqrt{\dot{x}^2 + \dot{y}^2} \leq 5\text{cm/s}.$$

Irrespective of constraints, the metric for performance of the rendezvous operation is fuel consumption, which is generalized to the total amount of control effort exerted:

$$J = \int_0^T ||\vec{u}||dt. \tag{2}$$

Note that this is different from the cost functional that will be applied to solve for control inputs.

Finally, the mission completion time can be considered both a constraint and performance metric, though we will not check that the constraint is met using verification tools. Instead we restrict the time horizon of our simulations accordingly and observe that the terminal state reaches a small neighborhood of the target. The mission *must* be completed before the time of eclipse, which is given to be 4 hours in [19].

IV. Hybrid controller design

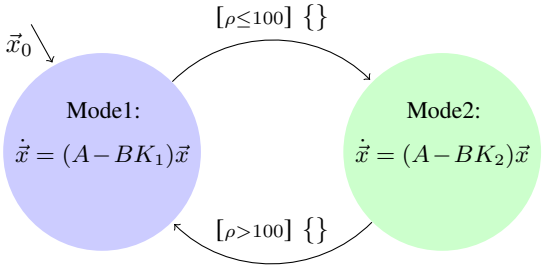
In this section, we present an overview of the control scheme we have developed. We begin with a review of the hybrid LQR-based controller for the rendezvous presented in [4]. The the state-dependent extension used in this case study is presented in Section IV-B.

A. Switched linear quadratic control

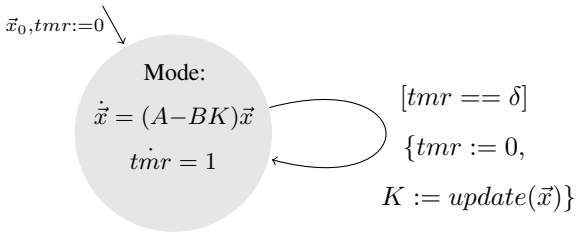
In [4], the state space is partitioned such that the chaser operates in two discrete modes: either inside or outside of the LOS region. State-feedback control is used in each mode, so the control law is of the form: $\vec{u} = -K_i\vec{x},\ i \in \{1, 2\}$. Thus, the result is a switched system with two subsystems and a state-dependent switching signal.

Each gain matrix $K_i \in \mathbb{R}^{2\times 4}$, $i \in \{1, 2\}$, is obtained from two independent infinite-horizon LQR problems, as follows:

$$\min_{\vec{u}=-K_i\vec{x}} \tilde{J}_i,$$



(a) Switched LQ control from [4].



(b) State-dependent LQ control.

Fig. 3: Hybrid models for spacecraft rendezvous with two different controllers. Transition guards are specified in "[]" and resets "{}". The $update(\cdot)$ method solves SDLQ problem.

where we distinguish the fuel consumption cost in (2) from the quadratic objective function used here:

$$\tilde{J}_i = \int_0^\infty (\vec{x}^T Q_i \vec{x} + \vec{u}^T R_i \vec{u})dt. \tag{3}$$

The solution to this particular optimization problem is obtained by solving the continuous algebraic Riccati equation (ARE). Part of the motivation for relying on a LQR-like framework is that ARE solvers are well-studied and readily available. Given the controllability of the relative motion model (1) and a restriction of Q_i, R_i to symmetric, positive definite matrices, there exists a unique optimal control solution $\vec{u} = -K_i\vec{x} = -R_i^{-1}B^T P_i$, where P_i is a positive definite solution to the ARE. The solution is not only optimal but ensures the closed-loop system is globally asymptotically stable (GAS) about the origin *in each mode*. This does not guarantee the overall system is GAS under switching.

The objective function in each mode is formulated by loosely accounting for the constraints on the states and inputs. *Bryson's rule* [21] is adapted, where Q_i and R_i consist of diagonals such that $Q_{ii} = \frac{1}{max(x_i^2)}$ and $R_{ii} = \frac{1}{max(u_i^2)}$. Typically these terms correspond to the expected range of values for each variable, which then normalizes the cost of errors in each direction. We take these terms to refer to the largest *desired* value of each variable, given by the constraints.

We note that a conventional switched LQR controller would require the solution of the following finite horizon problem:

$$\min_{\{\vec{u}_i\}} \sum_{i=1}^{N} \int_{t_i}^{t_{i+1}} (\vec{x}^T Q_i \vec{x} + \vec{u}^T R_i \vec{u})dt, \tag{4}$$

where t_i denotes switching times with N-total switches. Thus, an optimal LQR solution would drive the state of the system

to the origin by the specified time horizon T_{N+1}, while minimizing (4). The infinite-horizon formulation in (3) is easier to solve, but it loses the guarantee that the state reaches the origin when it is applied to a finite-horizon problem. This is acceptable for the rendezvous maneuver as the goal is to drive the spacecraft within closer range, not to a terminal point. The infinite-horizon LQR result will still drive the spacecraft towards the origin, which can be observed by formulating invariant sets of states using Lyapunov functions. At each switching time t_i, the invariant set of states at t_i is strictly contained in the invariant set at t_{i-1}. It follows that this switched controller with $i = \{1, 2\}$ (and the SDLQ in Section IV-B) are stable in the sense of Lyapunov.

B. State-Dependent Linear Quadratic (SDLQ) Control

We extend the two-stage, switched LQ control from the previous section to $N > 2$ finitely many switches where the switches are brought about by a time-dependent switching signal. Additionally, subsystem dynamics are not computed a priori but determined by a function of the current state, thus we refer to the new control scheme as state-dependent linear quadratic (SDLQ) control. The switched LQ and SDLQ schemes applied to the system model in (1) results in a hybrid automaton as shown in Figure 3.

At every switching time, the state-feedback law is computed according to (3). In the switched LQ scheme, $Q_{\{1,2\}}$, $R_{\{1,2\}}$ are constant matrices. Now, they are functions denoted $Q_i(\vec{x}(t_i))$ and $R_i(\vec{x}(t_i))$. For this paper, we choose a constant $R_i = R,\ \forall i \in \{0, 1, ..., N\}$, and $Q_i(\vec{x}(t_i))$ defined as:

$$
\begin{aligned}
Q_i(\vec{x}(t_i)) &= diag\left(\frac{1}{q_x^2}, \frac{1}{q_y^2}, \frac{1}{q_{\dot{x}}^2}, \frac{1}{q_{\dot{y}}^2}\right), \\
q_x &= 5\left(|x(t_i)| + \epsilon\right)\left(1 + \frac{(|x(t_i)| + \epsilon)^2}{\rho^2}\right), \\
q_y &= 5\left(|y(t_i)| + \epsilon\right)\left(1 + \frac{(|y(t_i)| + \epsilon)^2}{\rho^2}\right), \\
q_{\dot{x}} &= \frac{40\left(|x(t_i)| + \epsilon\right)}{\rho}, \\
q_{\dot{y}} &= \frac{40\left(|y(t_i)| + \epsilon\right)}{\rho},
\end{aligned}
\tag{5}
$$

for some small $\epsilon > 0$ to avoid division by zero.

Notice Q_i remains symmetric positive definite for all $\vec{x} \in \mathbb{R}^4$ and R is also chosen to be symmetric positive definite. Then, as in the switched LQ control, we are guaranteed to find a stabilizing solution $\vec{u}_{[t_i, t_{i+1})} = -K_i\vec{x}$.

The current choice of (5) is motivated by satisfying the LOS constraint, hence why it is only a function of $x,\ y$ and not relative velocities $\dot{x},\ \dot{y}$. Bryson's rule can be observed as a starting design choice in $q_x,\ q_y$ in the $(|x(t_i)| + \epsilon)$, $(|y(t_i)| + \epsilon)$ terms. In other words, the maximum desired values for $x,\ y$ contract as the chaser moves towards the origin. The terms $(|\{x, y\}(t_i)| + \epsilon)/\rho$ approximate $|\cos(\theta)|$ and $|\sin(\theta)|$, where θ is the angular position of the total displacement vector $\vec{\rho}$. These terms could be used to enforce the θ restriction in the LOS region more directly.

V. Data-driven bounded safety verification

In this section, we give an overview of the verification algorithm used to check the space rendezvous problem using the SDLQ controller. The data-driven, bounded-time safety verification algorithm uses simulation data and sensitivity analysis of the model to compute over-approximations of the bounded-time reachable states of the system.

The sensitivity of the system is formalized by the notion of a discrepancy function [6]: A uniformly continuous function $\beta : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a *discrepancy function* of a dynamical (or switched) system if (1) for any pair of states $x, x' \in \mathbb{R}^n$, and any time $t > 0$, the solutions $\xi(\cdot, \cdot)$ of the system satisfy the properties

$$
\|\xi(x, t) - \xi(x', t)\| \leq \beta(x, x', t), \text{and} \tag{6}
$$

(2) for any t, as $x \to x'$, $\beta(., ., t) \to 0$. In [6] an invariant verification algorithm is presented *assuming that the user will provide the necessary discrepancy functions as annotations to the system model*. This algorithm, which is the basis of the C2E2 tool [11][7], proceeds as follows: First, a δ-cover $C = \{x_i\}_{i=1}^k$ of the initial set of states Θ is computed. That is, $\Theta \subseteq \cup_i B_\delta(x_i)$. For each $x_i \in C$, a numerical simulation from x_i of the required time duration T is computed. The simulation from x_i over each sub-interval $[t_1, t_2]$ is expanded by a factor given by the discrepancy function β:

$$
\beta_{max,i} = \max_{x \in B_\delta(x_i), t \in [t_1, t_2]} \beta(x_i, x, t).
$$

The expansion results in the set of states $B_{\beta_{max,i}}(\xi(x_i, t))$, which can be shown to contain all the states reachable from $B_\delta(x_i)$ over time interval $[t_1, t_2]$.

If the complete reachable set is disjoint from the given unsafe $\mathcal{U}$ set then x_i is removed from the cover. Else if the simulation is contained in $\mathcal{U}$ for any of the intervals then the algorithm outputs **Unsafe**. Otherwise, if neither case holds, then x_i is replaced in C by a finer cover of $B_\delta(x_i)$. If the cover C becomes empty, then the algorithm outputs **Safe**. Essentially, the algorithm computes increasingly finer covers of Θ until the reachtubes from each of the elements in the cover are inferred to be disjoint from $\mathcal{U}$ or a counter-example simulation is discovered. Property (2) of the discrepancy function ensures that as the elements in the cover become finer, the over-approximation of the computed reachtubes becomes more precise, and a decision is reached. We refer the interested reader to [6] for more details including the precise statements about soundness and completeness guarantees provided by the algorithm for bounded safety verification.

Unfortunately, discrepancy functions can be difficult to come by. There is a line of work on computing accurate discrepancy functions for general nonlinear models [9], [8], [11]. But even these results are not applicable to the control system of Section IV-B, where there is no closed-form model of the system—the control inputs are computed up to a finite horizon by solving optimization problems. Thus, we use the recent tool DryVR [10] which uses the same safety verification algorithm described above, but implements a new randomized algorithm for learning discrepancy functions of

black-box models. A template for discrepancy called *global exponential discrepancy* is fixed which is of the form:

$$\beta(x_1, x_2, t) = |x_1 - x_2| K e^{\gamma t}.$$

Here K and γ are constants that will be learned from simulation data, such that they satisfy inequality (6) for all x_1, x_2 and t, with high probability. DryVR relies on a method for discovering discrepancy functions that only uses simulations, solving linear programs, and is based on classical results on PAC learning linear separators [23]. The high probability guarantee follows from the PAC-learnability of concepts with low VC-dimension (see [10] for details). Assuming that the learned discrepancy function is correct, the safety verification algorithm is sound and relatively complete. In Section VI, we will present the verification results from DryVR on the rendezvous problem and we will independently check the correctness of the computed discrepancy function through random sampling.

VI. Experimental results

In this section, we present the verification results for our SDLQ control system using the DryVR tool.

A. Safety Verification

The SDLQ control system is implemented in a Matlab program. This involves simulating the plant model described in Equations (1), computing the state-dependent matrices Q_i according to the method described in Section IV-B, and solving the infinite-horizon optimization problem of Equation (3) using Matlab's `lqr()` function. For the current experiments, we only run simulations for the distance range up to $\rho = 20$m—a range we define to be sufficiently close for completion of a generic rendezvous mission. If this state is reached before T, the last state is copied to remaining time steps. The simulations are performed using Matlab's ODE solvers, however, for more rigorously generated simulations the exact same verification approach can be used with validated numerical simulators such as CAPD [1].

The DryVR tool interfaces with the above program as follows: it provides as input to SDLQ an initial state of the form: $\vec{x}_0 = [x, y, \dot{x}, \dot{y}, F_x, F_y]$, and a finite time horizon T; the SDLQ Matlab program generates as output an array of points corresponding to running the control system while updating the control every $\delta = 30$ seconds.

The parameters for bounded-verification used by DryVR are as follows: The total time horizon for is chosen to be $T = 240$ minutes. The initial set Θ is a subset of $\mathbb{R}^6$, for example, for the results reported below it is the set: $-905\text{m} \leq x \leq -895\text{m}$, $-405\text{m} \leq y \leq -395\text{m}$, $\dot{x} = \dot{y} = 0\text{m/min}$, $F_x = F_y = 0\text{kg} \cdot \text{m} \cdot \text{min}^{-2}$. And, the unsafe set is defined by the set of constraints:

- *thrust constraints* $|F_x| > 10$, $|F_y| > 10$,
- *LOS constraints* $(x \geq -100 \cap y > x\tan(\pi/6))$ and $(x \geq -100 \cap y < -x\tan(\pi/6))$.

For these parameters, DryVR checks safety of SDLQ automatically and returns a **Safe**. The reachtubes computed in the process are plotted in Figures 4a and 4b. While the running time of this verification experiment is not immediately practical, it establishes feasibility of the approach and motivates the need for more careful engineering and parallelization. In another experiment with a smaller Θ (1m radius in x and y dimensions), the verification process took just under 1 hour to complete and returned a safe result.

Using the same T and Θ as before, but this time checking a velocity constraint defined by $\mathcal{U} = \{\vec{x} \in \mathbb{R}^6 : (x \geq -100 \cap \sqrt{\dot{x}^2 + \dot{y}^2} > 3)\}$, DryVR returns a **Unsafe** and finds a counter-example. The result of the corresponding simulation trace plotted in Figure 4c. It took DryVR less than 1 minute to find this counter-example, which illustrates its effectiveness as an approach for finding design bugs. This result is not surprising given that our SDLQ design only considered meeting the LOS and thrust constraints.

Finally, we check the correctness of the discrepancy function learned in DryVR, that is, whether the learned function meets the property of discrepancy functions stated in (6). For this experiment, we randomly sample the initial states in Θ, generate their simulation traces, and check whether any pair of these traces provide a counterexample to (6). We do this for 600 samples and find no such counterexamples, thus we conclude that the discrepancy function computed by DryVR in this example holds with high confidence.

B. SDLQ Performance Comparison

Though we do not design explicitly for optimal performance using the SDLQ control scheme, we evaluate and compare its performance against the switched LQR scheme used in [4] and the baseline controller used in [19] (which is based on the MPC problem in [20]) under the same scenarios used for the verification tool.

We simulate from the center initial state of Θ, or $\vec{x}_0 = [-900, -400, 0, 0, 0, 0]$, using each of the three different controllers up to an upper time horizon bound $T = 240$min.

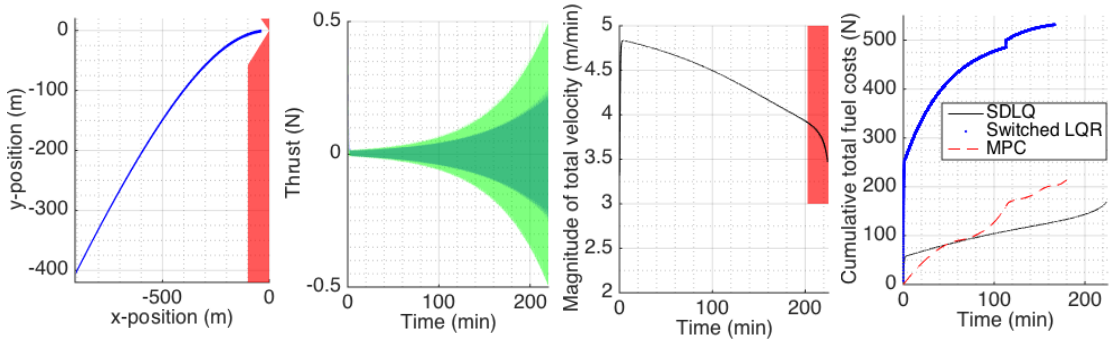
	$\vec{x}_{t_f}$	t_f	$J(t_f)$
SDLQ	[-19.99, -0.23, 3.52, 0.09]	223.7min	168.6N
LQR	[-19.09, -5.97, 0.57, 0.18]	166.4min	531.8N
MPC	[-19.91, -0.01, 0.003, 0.0]	180.0min	213.0N

TABLE I: Comparison of terminal states, completion times, and total fuel cost.

Figure 4d shows the cumulative fuel consumption, as defined by the performance metric in (2). Table I summarizes the different categories each control scheme performs best and worst in. MPC minimizes the closing velocity best without sacrificing too much in completion time and fuel consumption. The switched LQR achieves the best completion time but at a high cost in fuel consumption. Finally, SDLQ achieves the best fuel costs with acceptable increase in completion time but unacceptable ranges of velocity that violate constraints.

VII. Conclusions

We presented a hybrid state-dependent LQ (SDLQ) control framework for constrained linear systems. The SDLQ controller is a piecewise continuous state-feedback controller



(a) (b) (c) (d)

Fig. 4: (a) Reachable positions (blue) and unsafe positions (red). (b) Reachable thrusts: F_x (blue) and F_y (green). (c) Simulated trajectory of total velocity starting from a state in Θ that violates the velocity constraint (red). (d) Cumulative fuel consumption for three different controllers obtained through simulation under same initial state and termination condition.

obtained by periodically recomputing a quadratic optimization problem that implicitly enforces system constraints. The design framework involves choosing an appropriate state-based function for the quadratic objective and then verifying that the resulting closed-loop system does not violate safety constraints using the data-driven reachability analysis tool DryVR. We applied this approach to a benchmark autonomous spacecraft rendezvous (ARPOD) problem of [19] and have demonstrated its promise by creating the first verified controller for this problem. While this approach has proven to be feasible and promising, the quest for creating a fully verified globally stabilizing controller for ARPOD remains open.

Acknowledgments

We thank Richard S. Erwin for providing the MPC controller tested here, and Bolun Qi for support with the verification experiments. This work was supported by NSF grants: CNS 1629949, CNS 1054247, and CCF 1422798.

References

[1] Computer assisted proofs in dynamic groups (capd). `http://capd.ii.uj.edu.pl/index.php`.

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

[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. In *ERTS*, 2010.

[4] N. Chan and S. Mitra. Verifying safety of an autonomous spacecraft rendezvous mission. In *ARCH17*, 2017.

[5] I. Chang and J. Bentsman. Constrained discrete-time state-dependent riccati equation technique: A model predictive control approach. In *CDC*, 2013.

[6] P. S. Duggirala, S. Mitra, and M. Viswanathan. Verification of annotated models from executions. In *EMSOFT*, 2013.

[7] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok. C2E2: A verification tool for stateflow models. In *TACAS*, 2015.

[8] C. Fan, J. Kapinski, X. Jin, and S. Mitra. Locally optimal reach set over-approximation for nonlinear systems. In *EMSOFT*, 2016.

[9] C. Fan and S. Mitra. Bounded verification with on-the-fly discrepancy computation. In *ATVA*, 2015.

[10] C. Fan, B. Qi, S. Mitra, and M. Viswanathan. DRYVR: data-driven verification and compositional reasoning for automotive systems. In *CAV*, 2017.

[11] C. Fan, B. Qi, S. Mitra, M. Viswanathan, and P. S. Duggirala. Automatic reachability analysis for nonlinear hybrid models with C2E2. In *CAV*, 2016.

[12] S. S. Farahani, I. Papusha, C. McGhan, and R. M. Murray. Constrained autonomous satellite docking via differential flatness and model predictive control. In *CDC*, 2016.

[13] G. Frehse. Phaver: Algorithmic verification of hybrid systems past hytech. In *HSCC*, 2005.

[14] 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*, 2011.

[15] 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 Conference & Exposition*, 2003.

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

[17] B. HomChaudhuri, M. Oishi, M. Shubert, M. Baldwin, and R. S. and. Computing reach-avoid sets for space vehicle docking under continuous thrust. In *CDC*, 2016.

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

[19] C. Jewison and R. S. Erwin. A spacecraft benchmark problem for hybrid control and estimation. In *CDC*, 2016.

[20] C. Jewison, R. S. Erwin, and A. Saenz-Otero. Model predictive control with ellipsoid obstacle constraints for spacecraft rendezvous. *IFAC-PapersOnLine*, 48(9):257–262, 2015.

[21] M. A. Johnson and M. J. Grimble. Recent trends in linear optimal quadratic multivariable control system design. *IEE Proceedings D*, 1987.

[22] 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.

[23] M. J. Kearns and U. V. Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

[24] B. P. Malladi, R. G. Sanfelice, E. Butcher, and J. Wang. Robust hybrid supervisory control for rendezvous and docking of a spacecraft. In *CDC*, 2016.

[25] C. P. Mracek and J. R. Cloutier. Control designs for the nonlinear benchmark problem via the state-dependent riccati equation method. *International Journal of Robust and Nonlinear Control*, 1998.

[26] J. D. Pearson. Approximation methods in optimal control i. sub-optimal control. *Journal of Electronics and Control*, 1962.

[27] D. Pinard, S. Reynaud, P. Delpy, and S. E. Strandmoe. Accurate and autonomous navigation for the ATV. *Aerospace Science and Technology*, 2007.

[28] A. Wernli and G. Cook. Suboptimal control for the nonlinear quadratic regulator problem. *Automatica*, 11(1):75–84, 1975.

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

[30] 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.

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