Data-Driven Reachable Set Computation using Adaptive Gaussian Process Classification and Monte Carlo Methods

A PREPRINT

Alex Devonport Electrical Engineering and Computer Sciences University of California, Berkeley alex_devonport@berkeley.edu Murat Arcak Electrical Engineering and Computer Sciences University of California, Berkeley arcak@berkeley.edu

October 8, 2019

ABSTRACT

We present two data-driven methods for estimating reachable sets with probabilistic guarantees. Both methods make use of a probabilistic formulation allowing for a formal definition of a datadriven reachable set approximation that is correct in a probabilistic sense. The first method recasts the reachability problem as a binary classification problem, using a Gaussian process classifier to represent the reachable set. The quantified uncertainty of the Gaussian process model allows for an adaptive approach to the selection of new sample points. The second method uses a Monte Carlo sampling approach to compute an interval-based approximation of the reachable set. This method comes with a guarantee of probabilistic correctness, and an explicit bound on the number of sample points needed to achieve a desired accuracy and confidence. Each method is illustrated with a numerical example.

1 Introduction

Reachable sets characterize the states to which a system may evolve using the knowledge of where it starts, what inputs may affect the system, and how long the system may evolve. Computing reachable sets is a critical step in the solution to control problems involving objectives such as safety, recurrence, and more complicated requirements expressed as automata or temporal logic specifications. However, accurate reachable sets are generally very expensive to compute, and common practice is to use a tractable relaxation, such as an overapproximation that is guaranteed to contain the true reachable set.

In relaxing the problem, the analyst must make a trade-off between computational tractability and accuracy of the overapproximation. There are many reachable set overapproximation methods that lie at different points of the tractabilityaccuracy spectrum. At one extreme, reachability methods based on the Hamilton-Jacobi-Bellman equations [1], [2] and dynamic programming [3], such as those used in the Level Set Toolbox [4], yield reachable set approximations that are very accurate but slow to compute. Zonotope-based methods [5], such as those used in the CORA toolbox[6], are faster to compute at the cost of some accuracy. At the opposite extreme, interval reachability methods [7]–[9] give overapproximations that require a minimum of resources to compute and store, but due to their strict geometry they are generally conservative.

In this paper we introduce a *data-driven* approach that allows for improvements in both tractability and accuracy, at the cost of a relaxed guarantee of correctness. The essence of this relaxation is to place a suitable probability measure over the initial set and the controls, and to define reachable sets as events on the induced probability space. Then, a sample of simulated system trajectories can be used to make probabilistic estimates of the true reachable set. To achieve the lowest computational complexity possible, we minimize the number of sample trajectories while maintaining a probabilistic guarantee of a given accuracy.

Probabilistic methods have been used to analyze the reachability of stochastic systems [10]–[13] and as an exploratory tool to guide deterministic reachability analysis [14]. Here, we investigate the probabilistic approach as a rigorous method in its own right to analyze the reachability of deterministic systems. Data-driven methods have also been used as a tool for robustness analysis of uncertain control systems [15], which allow for probabilistic verification of

robustness against various types of uncertainty. This paper provides a similar approach to the problem of reachable set computation.

We present two data-driven methods for computing reachable set approximations that make use of the probabilistic relaxation. The first method uses a Gaussian process classifier (GPC) to construct a probabilistic reachable set of arbitrary accuracy. The prediction uncertainty of the GPC allows us to employ an *active learning* method [16], where we sequentially select samples in order to maximize information gain. The second method uses a Monte Carlo sampling approach to construct an interval overapproximation of the probabilistic reachable set. Although less accurate, this method comes with a provable probabilistic guarantee. The two methods are complementary: the GPC method allows for approximations of higher accuracy (since it is not restricted to interval approximations), while the Monte Carlo method can make faster approximations. When probabilistic guarantees are acceptable for the problem at hand, the formalism and methods described in this paper can offer a significant computational speedup. An additional advantage of the data-driven approach is that it may be used in a model-free way: we need only to be able to sample system trajectories, so the system itself is allowed to be a black box or otherwise inaccessible. Indeed, many high-fidelity models are either available only in black-box form, or are too complex to analyze with standard reachability tools.

2 Reachable Sets

Suppose we have a dynamical system with state transition function $\Phi(t; t_0, x_0, u)$ that maps an initial state $x_0 \in \mathbb{R}^n$ at time t_0 to a unique final state at time t_1 , under the influence of an input $u \in C^{[t_0, t_1]}$ and the system dynamics. For example, if the system is defined as a vector ordinary differential equation

$$\dot{x}(t) = f(x(t), u(t), t) \tag{1}$$

whose solutions are well-defined and unique on the interval $[t_0, t]$, then $\Phi(t; t_0, x_0, u)$ is the solution to (1) satisfying the initial condition $\Phi(t_0; t_0, x_0, u) = x_0$.

Now, suppose we have an *initial set* $\mathcal{X}_0 \subset \mathbb{R}^n$, and an *input set* $\mathcal{U} \subset C^{[t_0,t_1]}$. We would like to know all of the states to which the system may evolve between times $[t_0, t_1]$ starting in the initial set, and subjected to any allowable input. The set of all such states is the *forward reachable set*, which we write as

$$R_{[t_0,t_1]} = \{ x | x = \Phi(t_1; t_0, x_0, u) \text{ for some } x_0 \in \mathcal{X}_0, u \in \mathcal{U} \}.$$
(2)

When the state transition function is invertible, we also consider the inverse of this problem. Suppose we have a *final* set set $\mathcal{X}_1 \subset \mathbb{R}^n$, and we would like to know all of the states that can reach \mathcal{X}_1 in the time $[t_0, t_1]$. The set of all such states is called the *backward reachable set*, that is

$$B_{[t_0,t_1]} = \{ x | \Phi(t_1; t_0, x, u) \in \mathcal{X}_1 \text{ for some } u \in \mathcal{U} \}.$$
(3)

We may also be interested in finding the set \mathcal{X}_e of all initial states for which some *event*, characterized by h(x, t, u) = 0, occurs at some time $t_e \ge t_0$. The set of all such states is called the *event set*, that is

$$E_{t_0} = \{ x | h(\Phi(t_e; t_0, x, u), t_e, u(t_e)) = 0 \text{ for some } t_e \ge t_0, u \in \mathcal{U} \}.$$
(4)

This is similar to the backwards reachable set problem, except that t_e is not known *a priori*. Further, t_e will in general not be the same for each state that leads to the event.

3 Probabilistic Reachable Sets

To frame the data-driven approach, we consider a *probabilistic relaxation* of the reachable set problems described above. The methods in this paper consist of sampling initial states and inputs, evaluating the transition function at these sample points, and using the results to estimate the reachable set. The state transition function may be available directly through numerical integration of (1), through more advanced computer simulations, or even through physical experiments.

A reachable set computed using a sample-based method can be at best only *probabilistically accurate*, so we would like a way to represent this notion as well. To formalize the notion of sampling from \mathcal{X}_0 , we define a random variable $X_0 \sim p_0$ over the initial set. The probability distribution $p_0 : \mathcal{X}_0 \to [0, 1]$ is called the *initial distribution*, and may be any distribution whose support is \mathcal{X}_0 . Similarly, we will define a random variable $U \sim p_u$ over the input set, with *input distribution* $p_u : \mathcal{U} \to [0, 1]$.

These two random variables, together with the state transition function, define the family of successor random variables $X_t = \Phi(t; t_0, X_0, U) \sim p_t$ for $t \ge t_0$. In general, the distribution p_t will be unknown, since the state transition



Figure 1: A diagram of a forward reachable set (upper graph), and its relaxation to an ϵ -accurate probabilistic forward reachable set (lower graph), and an overapproximation of the probabilistic reachable set.

function is not known. The successor distribution can be used to define a probability space whose sample space is the state space \mathbb{R}^n , whose events are the Borel sets of \mathbb{R}^n , and whose probability measure is p_t . In this probability space, the probability of an event ω corresponds to the probability that the successor of a random initial state and input is an element of ω . This means that the true forward reachable set $R_{[t_0,t_1]}$ corresponds to the smallest event of probability 1. With that in mind, we define the ϵ -accurate reachable sets, denoted $R_{[t_0,t_1],\epsilon}$ as the smallest events with probability $1 - \epsilon$. A set $R \subset \mathbb{R}^n$ such that $p_t(R) \ge 1 - \epsilon$ is an overapproximation of an ϵ -accurate reachable set, since it must contain an ϵ -accurate reachable set. The relationship between the deterministic and probabilistic cases for forward reachable sets for a two-state system is illustrated in Figure 1.

We define a similar probabilistic formulation for backward reachable sets. The only difference is that we will choose a final random variable X_1 and U, and let $X_0 = \Phi^{-1}(t; t_0, X_1, U)$, where

$$\Phi^{-1}(t;t_0,x_1,u) = \{x|x_1 = \Phi(t;t_0,x,u), u \in \mathcal{U}\}.$$
(5)

For event sets, we are not interested specifically in the probabilistic behavior of Φ , but instead in the likelihood that a given sample in an initial set will lead to the event. Essentially, we would like to use samples to inform our belief about the location of the event set, so it is sensible to adopt a Bayesian formulation for the probabilistic event set. We employ a distribution over the initial set, $p_0(x)$, which represents our belief that x is in the event set prior to seeing any samples. Then the posterior distribution conditioned on the sample trajectories represents an updated belief that the point x belongs to the event set that takes information from the sample trajectories into account. We call this posterior distribution the *event distribution*, p_e .

4 Gaussian Process Classification (GPC) With Adaptive Sampling

The first method we present uses a Gaussian stochastic process to construct a binary classifier that identifies an estimate of the reachable set. A point in the state space is either in the reachable set or out of it, so determining the set of points in the reachable set has a natural representation as a binary classification problem.

A Gaussian process g is a random variable defined over a space of functions with the property that the joint distribution of any finite selection of point evaluations of the function is distributed as a joint Gaussian random variable [17]. The covariance between any two point evaluations $g(x_1)$ and $g(x_2)$ is $k(x_1, x_2)$, where k is the *kernel function* of the process.

Suppose we have a set of m sample points $x^{(i)} \in \mathbb{R}^n$ and their associated labels $y^{(i)} \in \{a, b\}$, where $a, b \in \mathbb{R}$, such that $y^{(i)} = a$ if $x^{(i)}$ is in the reachable set, and $y^{(i)} = b$ otherwise. For example, a suitable choice of labels would be a = 1, b = 0. We use a Gaussian process to construct a classifier that minimizes the *regularized least-squares classification risk*, that is a function $g : \mathbb{R}^n \to \mathbb{R}$ that minimizes

$$\sum_{i=1}^{m} \left(g(x^{(i)}) - y^{(i)} \right)^2 + ||g||_k \tag{6}$$

where $||\cdot||_k$ is a norm that depends on the kernel function. A classifier that minimizes this risk is called a *least-squares* classifier. Least-squares classification is attractive here because the mean $\mu_{\hat{g}}$ and variance $\sigma_{\hat{g}}$ of the Gaussian process \hat{g} that minimizes this risk have analytic expressions that can be computed quickly.

To make predictions using this classifier, we select a *threshold* $\gamma \in (a, b)$, and declare that a point x is predicted to be in the reachable set if $\hat{g}(x) \ge \gamma$, and not in the reachable set otherwise. For example, in the a = 1, b = 0 case, $\gamma = 0.5$

is suitable. With a threshold chosen, the reachable set estimate produced by this method is the sublevel set

$$\dot{R} = \{x | \mu_{\hat{g}}(x) < \gamma\}. \tag{7}$$

To construct a data set, we select a set of sample points $x^{(i)}$, and use the state transition function to assign a label $y^{(i)}$ to each of the sample points based on whether or not it is in the reachable set.

In principle, we may select the sample points in any way we like, e.g. uniform sampling over the region of interest, or using Latin hypercube sampling. However, since we wish to minimize the number of transition function evaluations, we use the GPC model of the reachable set to inform our choice of future sample points. This kind of sampling is called *adaptive sampling*, since our selection method adapts according to the incoming data, and is an *active learning* method. The use of adaptive sampling to guide the construction of a Gaussian process model is motivated by a method from optimal experiment design known as *Adaptive Kriging* [18], [19], in which a Gaussian process regression is used to form a *surrogate model* for an expensive computational model.

To have the GPC inform our selection of sample points, we use the prediction uncertainty of the GPC to sequentially select sample points that maximize some measure of information gain. The most relevant measure of information gain for this problem is the *probability of misclassification* [20], that is the probability that a state in the reachable set is identified as being outside of it, or vice versa. With classifier threshold γ , the probability of misclassification is

$$P_{misclass}(x) = \Phi\left(-\frac{|\mu_{\hat{g}}(x) - \gamma|}{\sigma_{\hat{g}}(x)}\right)$$
(8)

where Φ in (8) is the cumulative distribution function of the standard normal distribution.

When selecting a new sample point, ideally we would like to find the point in the state space with the highest probability of misclassification. However, this is a nonconvex and potentially high-dimensional optimization problem. Instead of searching the entire state space for a new sample, we use a stochastic optimization approach proposed in [18] and search over a large pool of randomly-selected *candidate samples*. We calculate the probability of misclassification for each candidate, and select the one with the highest probability of misclassification to be the next sample. The sample pool is selected using a Latin Hypercube, so that the candidate samples will be evenly distributed over a compact region of the state space. Note that this is distinct from selecting samples directly by a Latin hypercube: after we have selected the candidate pool, only a small number of candidate points will be selected as sample points, and the distribution of the selected points will be guided by the probability of misclassification.

Example: Safe Set Estimation for Adaptive Cruise Control

Consider the Adaptive Cruise Control (ACC) scenario depicted in Figure 3. In this scenario, a car being operated by ACC (the *follower*) is driving behind another car (the *leader*). The follower and leader are initially traveling with positive velocities $v_F(0)$ and $v_L(0)$ respectively. At t = 0, the leader begins to brake and eventually comes to a halt. If the distance between the leader and follower becomes zero at any t > 0, then the two cars have collided. To prevent this, we determine what initial states (that is, velocities and relative positions at t = 0) give the follower enough time to prevent a collision. We call the set of all such initial states a "safe set".

We use the following point-mass model for the dynamics of the two vehicles:

$$h(t) = v_L(t) - v_F(t) \tag{9}$$

$$\dot{v}_L(t) = -a - bv_L(t)^2 \tag{10}$$

$$\dot{v}_F(t) = -a - bv_F(t)^2 \tag{11}$$

where $v_L(t)$ and $v_F(t)$ are the velocities of the leader and follower, respectively, and h(t) is the distance between the two cars. The acceleration of each car has a constant term from the brakes, as both cars are applying the brakes fully, and a quadratic term from drag force.

This problem is an event set estimation problem because the safe set we wish to determine corresponds to the complement of the set of initial conditions $x(0) = [h(0) v_L(0) v_F(0)]^T$ for which the event h(x) = h = 0 occurs.

This model can be solved analytically, and the true event set is shown in the Appendix to be

$$E = \{(h, v_L, v_F) | h + \frac{1}{2b} \log\left(1 + \frac{b}{a}v_L^2\right) - \frac{1}{2b} \log\left(1 + \frac{b}{a}v_F^2\right) \ge 0\}.$$
(12)

Since we know the true event set, we can directly observe how well the GPC method approximates the true event set under different conditions.



Figure 2: Estimated safe set boundaries (dashed lines) for the ACC model computed with the GPC method compared with the true safe set boundary (solid lines), which is calculated analytically. The sample locations are also shown: an 'o' indicates that a collision occurred, and an 'x' indicates that it did not. The model parameters are set at a = 4.9, b = 1, and the initial follower velocity is fixed at $v_F(0) = 5$. Top row: m = 50 sample points. Bottom row: m = 200 sample points. For adaptive sampling, a candidate pool of $m_{candidate} = 1000$ samples was used in both cases. For the two sample sizes shown, adaptive sampling is able to make the most accurate approximation of the event set out of the three sampling methods used.



Figure 3: Diagram of the leader and follower, and the associated state variables, in the ACC braking model. If $h(t_e) = 0$ for some $t_e \ge t_0$, the two cars have collided.

For convenience of visualization, we hold the initial velocity of the follower constant at $v_F(0) = 0.5$. We restrict our attention to a compact region of the state space, specifically

$$0 \le h \le 2 \tag{13}$$

$$0 \le v_L \le 5. \tag{14}$$

Using sample points from this region, we construct a least-squares GPC using a *squared-exponential* kernel, that is we take

$$k(x_1, x_2) = \sigma \exp(-(x_2 - x_1)^T \Lambda(x_2 - x_1)),$$
(15)

where σ and the diagonal matrix $\Lambda = \text{diag}(\ell_1, \ell_2, \ell_3)$ are hyperparameters that are selected using maximum likelihood.

To demonstrate the effectiveness of the adaptive sampling method, we compare it to two other non-adaptive sampling strategies: sampling uniformly at random over the region, and sampling with a Latin hypercube over the region. To demonstrate how the number samples affects the quality of the predicted event set, we form two sample sets for each of the sampling methods, with m = 50 and m = 200 each.

For adaptive sample selection, we begin by selecting a pool of $m_{candidate} = 1000$ candidate samples from the region of interest using a Latin hypercube. Three samples are selected at random to serve as the initial set for the GPC model, and the remaining m - 3 are selected by sequentially minimizing the probability of misclassification.

The GPC-estimated event sets are shown in Figure 2. The true reachable set is also shown, to confirm that the estimated reachable sets are converging to the ground truth. For both sample sizes, the adaptive sampling method makes the most accurate event set estimate out of each of the three methods. By maximizing the probability of misclassification with each new sample, the adaptive method will either select a new sample with high prediction variance, which will be far away from the other samples, or one whose prediction mean is close to the threshold; that is, one close to the border. After minimizing the variance over most of the state space region of interest, the adaptive method begins to select samples that are likely to be near the border of the event set. Using this strategy, the adaptive GPC reduces the number of samples required to identify points near the boundary of the event set. By contrast, the uniform and Latin hypercube methods can only select states at random, so the likelihood of selecting a sample point near the boundary of the event set never increases.

We used a point-mass model here so that the true event set could be derived, as has been done in the Appendix. However, this same analysis could be carried out with a high-fidelity model with no change to the methodology.

5 Monte Carlo Interval Overapproximation

We now present a Monte Carlo Sampling (MCS) approach to produce *interval* overapproximations of epsilon-accurate reachable sets, that is overapproximations of the form

$$\hat{R} = [\underline{x}, \overline{x}] = \{ x | \underline{x} \le x \le \overline{x}, \underline{x} \in \mathbb{R}^n, \overline{x} \in \mathbb{R}^n \}$$
(16)

where \leq is the vector inequality corresponding to the positive orthant cone of \mathbb{R}^n . Geometrically, the set $[\underline{x}, \overline{x}]$ is an axis-aligned hyperrectangle of dimension n whose least point is \underline{x} and whose greatest point is \overline{x} .

In general, even the tightest interval overapproximation of the reachable set will be inaccurate, since intervals have such restricted geometry. Despite the inaccurate nature of their approximation of the true reachable set, interval approximations can be an appropriate design choice when quick computation and low memory requirements are preferable to reachable set accuracy.

An important example of when interval approximation is a suitable design choice is *symbolic control*, where controller synthesis is carried out on a finite-state machine *abstraction* that simulates the continuous-state dynamical system [9], [21]–[26]. The states of the abstraction represent the cells of a partition of \mathbb{R}^n , and the transitions are derived from the intersection of the forward reachable sets of each cell with the other cells. For high-dimensional state spaces, the number of reachable sets that must be computed and stored grows rapidly, so it is necessary to use a reachable set overapproximation that can be computed quickly and is memory-efficient.

A simple method to calculate the interval approximation is a Monte Carlo approach. For the forward reachable set case, this would consist of the following steps:

- 1. take a set of m samples each from the initial distribution and input distribution, $\{x_0^{(i)}\}_{i=1}^m$ and $\{u^{(i)}\}_{i=1}^m$;
- 2. Evaluate the sample successor states $x_1^{(i)} = \Phi(t; t_0, x_0^i, u^i);$
- 3. Take $\hat{R}^{(m)}$ as the smallest interval containing all of the $x_1^{(i)}$.

Despite its simplicity, the Monte Carlo Sampling (MCS) method described above is provably effective at overapproximating ϵ -accurate reachable sets with intervals. In particular, the inequality (17), adapted from an example in [27] serves as a lower bound on the number of sample points required to ensure that the method described above produces an overapproximation of a desired accuracy and confidence. Although similar in form, the sample complexity bound (17) is distinct from the Chernoff bounds commonly used in the analysis of Monte Carlo methods, in ways that will be discussed after the proof.

Theorem 1. Let ϵ , $\delta \in (0, 1)$. If

$$m \ge \frac{2n}{\epsilon} \log\left(\frac{2n}{\delta}\right),\tag{17}$$

then $\hat{R}^{(m)}$ overapproximates an ϵ -accurate reachable set with confidence δ , i.e. $P(R_{[t_0,t_1],\epsilon} \subset \hat{R}^{(m)}) \ge 1 - \delta$.

Proof. First, suppose that we have m samples of the successor $\{x_1^{(i)}\}_{i=1}^m$, and that the resulting $\hat{R}^{(m)}$ overapproximates the ϵ -accurate reachable set.



Figure 4: Diagram of the orthosis and its user, from [28]. State variables are labeled in red and green. Parameters are labeled in blue and black.

An *n*-dimensional interval has 2n "faces", which are all axis-aligned. To each "face" f_i of the interval we associate the half space \mathcal{H}_i whose boundary hyperplane coincides with that face, and faces "away from" the interval.

Now, let \mathcal{P}_i be a half space parallel to \mathcal{H}_i such that $p_{t_1}(\mathcal{P}_i) \geq \frac{\epsilon}{2n}$. Furthermore, let \mathcal{P}_i be the *smallest* such half space, in the sense that any other half space \mathcal{P} parallel to \mathcal{H}_i such that $p_{t_1}(\mathcal{P}_i) \geq \frac{\epsilon}{2n}$ contains \mathcal{P}_i . By the right-continuity of p_1 , this half space is unique. In the case that p_1 is continuous, this is just the unique hyperplane parallel to \mathcal{H}_i such that $p_{t_1}(\mathcal{P}_i) \geq \frac{\epsilon}{2n}$.

The probability that none of the *m* samples is in the half space \mathcal{P}_i is $(1 - \frac{\epsilon}{2n})^m$. This implies that the probability that at least one of the \mathcal{P}_i contains no samples less than $2n(1 - \frac{\epsilon}{2n})^m$, and that the probability that each \mathcal{P}_i contains at least one sample is greater than $1 - 2n(1 - \frac{\epsilon}{2n})^m$.

If one of the samples is in \mathcal{P}_i , then $\mathcal{H}_i \subset \mathcal{P}_i$, meaning that $p_{t_1}(\mathcal{H}_i) \leq p_{t_1}(\mathcal{P}_i) \leq \frac{\epsilon}{2n}$. If this is true for each \mathcal{P}_i , then $p_{t_1}(\bigcup_i \mathcal{H}_i) = p_{t_1}(\hat{R}^{(n)\complement}) \leq \epsilon$, and so $p_{t_1}(\hat{R}^{(i)}) \geq 1 - \epsilon$. This implies that

$$P(p_{t_1}(\hat{R}^{(i)}) \ge 1 - \epsilon) \ge 1 - 2n(1 - \frac{\epsilon}{2n})^m.$$
(18)

To ensure that $p_{t_1}(\hat{R}^{(i)}) \ge 1 - \epsilon$), i.e. that $\hat{R}^{(m)}$ overapproximates contains an ϵ -accurate reachable set, with probability $\ge 1 - \delta$, it suffices to ensure

$$\delta \ge 2n \left(1 - \frac{\epsilon}{2n}\right)^m. \tag{19}$$

Using the identity $\log(1-x) \leq -x$, valid for $0 \leq x \leq 1$, we can conclude that (17) guarantees (19).

For standard Monte Carlo methods, such as Monte Carlo Integration, probability inequalities such as *Chernoff bounds* can be used to determine the number of samples needed for a desired accuracy. However, Chernoff bounds are not applicable in this case. Chernoff bounds apply to the tail probability of a sum of iid random variables (e.g. a Monte Carlo integral) exceeding an arbitrary parameter, and are derived by applying Markov's inequality to the moment generating function of the random sum. In our case, however, we are interested in bounding the tail probability of the *maximum* of an iid sequence of random variables exceeding a specific observed value (the empirical maximum). The strategy for deriving a Chernoff bound does not work in this case, which is why a different strategy was needed to prove Theorem 1.

Example: Robustness Analysis For a Powered Lower Limb Orthosis Through Forward Reachable Sets

We now present an application of the MCS method to a reachable-set problem posed in [28] for robustness analysis of a sit-to-stand motion controller for powered lower limb orthoses.

The Powered Lower Limb Orthosis and its user are modeled in [28] as a three-link planar robot with three joints. The model has six states (angles and angular velocities of the three joints, labeled in red and green in Figure 4) and twelve



Figure 5: Reachable set overapproximations (black boxes) of the center of mass trajectories at three time instants over sit-to-stand movement, calculated using the Monte Carlo interval overapproximation method with $\epsilon = 0.05$, $\delta = 0.001$. m = 7554 samples trajectories (also shown) were required to ensure the specified accuracy and confidence over the parameter-augmented 18-dimensional state space.

parameters (the lengths, masses, and moments of inertia of the three links, and the distances between each joint and the center of moment of its corresponding link, labeled in blue and black in Figure 4). The twelve parameters are *uncertain*, since they all depend on the weight of the user. The sit-to-stand reference trajectory is defined in terms of the position in the x- and y- (horizontal and vertical in Figure 4) directions of the center of mass (CoM) of the user and orthosis, which can be computed from the state variables and parameters.

The authors of [28] designed a finite time horizon LQR controller to track a reference trajectory that brings the user and orthosis from a sitting position to a standing position. To analyze the robustness of the motion to parameter variations, the authors used sensitivity-based reachability methods to compute an interval overapproximation of the CoM trajectories subject under parameter changes induced by a 5% variation in user body weight. In this example, we approach this robustness analysis problem by using the MCS method to compute an interval overapproximation of the CoM trajectory with accuracy $\epsilon = 0.05$ and confidence $\delta = 0.001$.

To recast the problem from a robustness analysis problem into a reachability problem, we add a new state variable p_i for each parameter with constant dynamics, i.e. $\dot{p}_i = 0$. The parameters may be selected by choosing the initial conditions of the p_i . This way, we can perform the robustness verification by computing reachable set overapproximations of the parameter-augmented 18-state system.

The parameter intervals are derived from human biometric data, and the initial state is assumed to be fixed. We choose an initial distribution to be the uniform distribution over the initial set. In this case, the initial distribution has a physical significance, namely that the parameters are subject to change from user to user. If we had more detailed statistical information about the parameters (e.g. that they are Gaussian-distributed with some mean and variance), that could be used to define the initial set instead.

Using (17) with values $\epsilon = 0.05$, $\delta = 0.001$, and n = 18, we know that m = 7554 sample trajectories will suffice to compute an $\hat{R}^{(m)}$ that has at the desired levels of accuracy and confidence.

The resulting interval overapproximations of the position of the CoM and is shown in Figure 5. Specifically, we show the overapproximation for three points in the sit-to-stand movement; at the beginning (t = 0), in the middle (t = 1.75), and at the end (t = 3.5) of the movement. In addition, the volumes of the intervals in Figure 5 are reported in Table 1, and compared to the volumes of the intervals computed using sensitivity-based methods in [28]. The table also reports the volumes of the interval overapproximations to the velocity of the CoM, which come from the same calculation as the position overapproximations. At each time, the volumes of the MCS rectangles is lower than those of the sensitivity method volumes, meaning that the MCS method gives less conservative estimates of the reachable sets.

interval	Volume from [28]	Volume from MCS
$(x_{CoM}, y_{CoM}), t = 0$	2.4×10^{-3}	1.7×10^{-3}
$(x_{CoM}, y_{CoM}), t = 1.75$	4.0×10^{-3}	2.5×10^{-3}
$(x_{CoM}, y_{CoM}), t = 3.5$	3.2×10^{-3}	5.0×10^{-4}
$(\dot{x}_{CoM}, \dot{y}_{CoM}), t = 0$	0	0
$(\dot{x}_{CoM}, \dot{y}_{CoM}), t = 1.75$	1.0×10^{-3}	7.4×10^{-4}
$(\dot{x}_{CoM}, \dot{y}_{CoM}), t = 3.5$	4.8×10^{-6}	1.1×10^{-6}

Table 1: Comparison of Interval Overpproximation Volumes

6 Conclusion

In this paper, we showed the utility of a probabilistic formulation of the problem of reachable set approximation by developing data-driven reachable set computation methods, and by verifying them with numerical examples. Between the two methods, the analyst can make a reachable set approximation to any degree of accuracy. Sets computed with the MCS method also come with a rigorous probabilistic guarantee of accuracy and confidence. Since the problem formulation makes few assumptions, these two methods can be used on a wide range of systems, many of which are too complex for standard reachability methods.

Both of the methods we present are useful in their present form, as shown by the examples. However, they can both be improved and extended with future work. For the MCS interval overapproximation method, the proof of (17) makes no assumptions on the successor distribution p_1 , other than that $p_1(x)$ be well-defined for all $x \in \mathcal{X}_0$. On one hand, having a distribution-free result is useful, since we don't in general have any knowledge of p_1 . On the other hand, it suggests that if we did have some knowledge about p_1 , the bound could perhaps be improved by using importance sampling or a related technique. Additionally, the proof as given depends on the geometry of hyperrectangles, and does not generalize to other classes of sets that we may want to use for interval reachable set overapproximation. For the GPC method, an useful extension would be a probabilistic guarantee, of a similar kind to the guarantee that the MCS method enjoys. It may also be useful to consider other classification methods than least-squares, which may be more accurate and amenable to analysis.

Acknowledgments

This work was supported in part by the grants ONR N00014-18-1-2209, AFOSR FA9550-18-1-0253, NSF ECCS-1906164.

References

- I. M. Mitchell and C. J. Tomlin, "Overapproximating reachable sets by Hamilton-Jacobi projections," *Journal of Scientific Computing*, vol. 19, no. 1, pp. 323–346, 2003, ISSN: 1573-7691. DOI: 10.1023/A:1025364227563. [Online]. Available: https://doi.org/10.1023/A:1025364227563.
- [2] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005, ISSN: 0018-9286. DOI: 10.1109/TAC.2005.851439.
- D. Bertsekas and I. Rhodes, "On the minimax reachability of target sets and target tubes," Automatica, vol. 7, no. 2, pp. 233 –247, 1971, ISSN: 0005-1098. DOI: https://doi.org/10.1016/0005-1098(71)90066-5. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005109871900665.
- [4] I. M. Mitchell and J. A. Templeton, "A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems," in *International Workshop on Hybrid Systems: Computation and Control*, Springer, 2005, pp. 480–494.
- [5] M. Althoff, O. Stursberg, and M. Buss, "Verification of uncertain embedded systems by computing reachable sets based on zonotopes," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 5125–5130, 2008.
- [6] M. Althoff, "An introduction to CORA 2015," in *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015.
- [7] P.-J. Meyer, A. Devonport, and M. Arcak, "TIRA: Toolbox for interval reachability analysis," *ArXiv preprint arXiv:1902.05204*, 2019.
- [8] P.-J. Meyer and D. V. Dimarogonas, "Hierarchical decomposition of LTL synthesis problem for mixed-monotone control systems," *ArXiv preprint arXiv:1712.06014*, 2017.

- [9] T. Moor and J. Raisch, "Abstraction based supervisory controller synthesis for high order monotone continuous systems," in *Modelling, Analysis, and Design of Hybrid Systems*, Springer, 2002, pp. 247–265.
- [10] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724 –2734, 2008, ISSN: 0005-1098. DOI: https://doi.org/10.1016/ j.automatica.2008.03.027. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ S0005109808002677.
- [11] M. Althoff, O. Stursberg, and M. Buss, "Stochastic reachable sets of interacting traffic participants," in 2008 IEEE Intelligent Vehicles Symposium, IEEE, 2008, pp. 1086–1092.
- [12] K. Margellos, P. Goulart, and J. Lygeros, "On the road between robust optimization and the scenario approach for chance constrained optimization problems," *IEEE Transactions on Automatic Control*, vol. 59, no. 8, pp. 2258–2263, 2014.
- [13] Y. Yang, J. Zhang, K.-Q. Cai, and M. Prandini, "Multi-aircraft conflict detection and resolution based on probabilistic reach sets," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 309–316, 2016.
- [14] K. Margellos and J. Lygeros, "Toward 4-d trajectory management in air traffic control: A study based on Monte Carlo simulation and reachability analysis," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1820–1833, 2013, ISSN: 1063-6536. DOI: 10.1109/TCST.2012.2220773.
- [15] R. Tempo, G. Calafiore, and F. Dabbene, Randomized algorithms for analysis and control of uncertain systems: With applications. Springer Science & Business Media, 2012.
- B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.
- [17] C. E. Rasmussen and C. K. Williams, Gaussian processes for machine learning. MIT Press, 2006.
- [18] B. Echard, N. Gayton, and M. Lemaire, "AK-MCS: An active learning reliability method combining Kriging and Monte Carlo simulation," *Structural Safety*, vol. 33, no. 2, pp. 145 –154, 2011, ISSN: 0167-4730. DOI: https://doi.org/10. 1016/j.strusafe.2011.01.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ S0167473011000038.
- [19] R. Schbi, B. Sudret, and S. Marelli, "Rare event estimation using polynomial-chaos Kriging," ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering, vol. 3, no. 2, p. D4016002, 2017. DOI: 10.1061/ AJRUA6.0000870.
- [20] J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez, "Sequential design of computer experiments for the estimation of a probability of failure," *Statistics and Computing*, vol. 22, no. 3, pp. 773–793, 2012.
- [21] J. Lunze, "Qualitative modelling of linear dynamical systems with quantized state measurements," *Automatica*, vol. 30, no. 3, pp. 417–431, 1994.
- [22] R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas, "Discrete abstractions of hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 971–984, 2000.
- [23] P. Tabuada, Verification and control of hybrid systems: A symbolic approach. Springer, 2009.
- [24] H. Kress-Gazit, T. Wongpiromsarn, and U. Topcu, "Correct, reactive, high-level robot control," *IEEE Robotics Automation Magazine*, vol. 18, no. 3, pp. 65–74, 2011.
- [25] C. Belta, B. Yordanov, and E. Gol, Formal methods for discrete-time dynamical systems. Springer, 2017.
- [26] G. Reissig, A. Weber, and M. Rungger, "Feedback refinement relations for the synthesis of symbolic controllers," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1781–1796, 2017, ISSN: 0018-9286. DOI: 10.1109/TAC.2016. 2593947.
- [27] M. Vidyasagar, *Learning and generalisation: With applications to neural networks, second edition.* Springer Science & Business Media, 2003.
- [28] O. Narvaez-Aroche, P.-J. Meyer, M. Arcak, and A. Packard, "Reachability analysis for robustness evaluation of the sit-tostand movement for powered lower limb orthoses," in ASME 2018 Dynamic Systems and Control Conference, Atlanta, GA, USA, 2018.

Appendix: Derivation of the True ACC Safe Set

For the system

$$\dot{x} = v, \tag{20}$$

$$\dot{v} = -bv^2 - a, \tag{21}$$

the distance traveled before coming to a full stop is (from the derivation below):

$$\frac{1}{2b}\ln\left(1+\frac{b}{a}v(0)^2\right).$$
(22)

Since h is a monotonic function of t (see below), to check for a collision it suffices to check the sign of h after both cars have come to a stop. if $h \ge 0$ when both cars have stopped, then $h(t) \ge 0$ for all prior t, and no collision occurred.

On the other hand, if h < 0 after the cars stopped, at some point h changed sign, at which time there was a collision. With this in mind, from (22) the safe set in the $(h(0), v_L(0))$ space is:

$$E(v_F(0)) = \left\{ (h, v) : h + \frac{1}{2b} \ln \left(1 + \frac{b}{a} v^2 \right) - \frac{1}{2b} \ln \left(1 + \frac{b}{a} v_F(0)^2 \right) \ge 0 \right\}.$$
 (23)

Derivation of (22): the solution of (21) is

$$v(t) = \sqrt{\frac{a}{b}} \tan\left(\tan^{-1}\left(\sqrt{\frac{b}{a}}v(0)\right) - \sqrt{ab}t\right)$$
(24)

and integration gives

$$x(t) - x(0) = \frac{1}{b} \ln \left(\frac{\cos\left(\tan^{-1}\left(\sqrt{\frac{b}{a}}v(0)\right) - \sqrt{ab}t\right)}{\cos\left(\tan^{-1}\left(\sqrt{\frac{b}{a}}v(0)\right)\right)} \right).$$
(25)

Note that v(t) = 0 when t is such that the argument of the tangent term in (24) is zero. Since the cosine in the numerator of (25) has the same argument, the numerator equals one when v(t) = 0. When, in addition, the denominator is simplified as

$$\cos\left(\tan^{-1}\left(\sqrt{\frac{b}{a}}v(0)\right)\right) = \left(1 + \frac{b}{a}v(0)^2\right)^{-1/2},$$

the right-hand side of (25) becomes (22).

Monotonicity of h(t): introducing some abbreviations, we can write (25) as

$$x_i(t) - x_i(0) = \frac{1}{b} \ln\left(\frac{\cos(\alpha_i - \beta t)}{\cos(\alpha_i)}\right)$$
(26)

where i = L or F. This This expression gives us \dot{h} as

$$\dot{h} = \dot{x}_L - \dot{x}_F = \frac{\beta}{b} \left(\tan(\alpha_L - \beta t) - \tan(\alpha_F - \beta t) \right).$$
(27)

Now we can consider two cases: $\alpha_L > \alpha_F$, and $\alpha_L \le \alpha_F$. In the first case, we will have $\alpha_L - \beta t > \alpha_F - \beta t$, and $\tan(\alpha_L - \beta t) > \tan(\alpha_F - \beta t)$. This makes $\dot{h}(t) > 0$ for all t, meaning that h(t) is monotonically increasing. By analogous reasoning, in the second case h(t) is monotonically decreasing.