MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Data-Driven Anytime Algorithms for Motion Planning with Safety Guarantees

Jha, D.; Zhu, M.; Wang, Y.; Ray, A.

TR2016-041 July 2016

Abstract

This paper presents a learning-based (i.e., datadriven) approach to motion planning of robotic systems. This is motivated by controller synthesis problems for safety critical systems where an accurate estimate of the uncertainties (e.g., unmodeled dynamics, disturbance) can improve the performance of the system. The state-space of the system is built by sampling from the state-set as well as the input set of the underlying system. The robust adaptive motion planning problem is modeled as a learning-based approach evasion differential game, where a machine-learning algorithm is used to update the statistical estimates of the uncertainties from system observations. The system begins with a conservative estimate of the uncertainty set to ensure safety of the underlying system and we relax the robustness constraints as we get better estimates of the unmodeled uncertainty. The estimates from the machine learning algorithm are used to refine the estimates of the controller in an anytime fashion. We show that the values for the game converges to the optimal values with known disturbance given the statistical estimates on the uncertainty converges. Using confidence intervals for the unmodeled disturbance estimated by the machine learning estimator during the transient learning phase, we are able to guarantee safety of the robotic system with the proposed algorithms during transience.

2016 American Control Conference (ACC)

Data-Driven Anytime Algorithms for Motion Planning with Safety Guarantees

Devesh K. Jha† Minghui Zhu⋆ Yebin Wang‡ Asok Ray†

Abstract— **This paper presents a learning-based (i.e., data-driven) approach to motion planning of robotic systems. This is motivated by controller synthesis problems for safety critical systems where an accurate estimate of the uncertainties (e.g., unmodeled dynamics, disturbance) can improve the performance of the system. The state-space of the system is built by sampling from the state-set as well as the input set of the underlying system. The robust adaptive motion planning problem is modeled as a learning-based approach evasion differential game, where a machine-learning algorithm is used to update the statistical estimates of the uncertainties from system observations. The system begins with a conservative estimate of the uncertainty set to ensure safety of the underlying system and we relax the robustness constraints as we get better estimates of the unmodeled uncertainty. The estimates from the machine learning algorithm are used to refine the estimates of the controller in an anytime fashion. We show that the values for the game converges to the optimal values with known disturbance given the statistical estimates on the uncertainty converges. Using confidence intervals for the unmodeled disturbance estimated by the machine learning estimator during the transient learning phase, we are able to guarantee safety of the robotic system with the proposed algorithms during transience.**

I. Introduction

Motion planning is a classic problem in robotics and has received a lot of attention in robotics, computer science and control systems society. The basic problem of computing a collision-free trajectory connecting an initial configuration or state to a target region through a cluttered environment is well-understood and fairly well-solved [15], [16]. However, the shortcomings of the basic path planning algorithms are revealed when considering how these algorithms are used for controlling an autonomous robotic system using an auxiliary controller. A fundamental problem with control of autonomous robotic systems (e.g., self-driving cars) is safety control i.e., ensuring the system stays in the given safety sets while simultaneously achieving the given objectives. It becomes difficult to ensure safety for these systems in dynamically changing and uncertain environments while using the path planning algorithms with the decades-old *Sense Plan Act* paradigm as the reachable sets in the presence of dynamic uncertainties are not considered.

To date, the state-of-the-art motion planning algorithms are sampling-based (like RRT and PRM [16]) and can guarantee asymptotic optimality [13]. More recently, some algorithms can provide convergence rates to the optimal solution [11]. Various algorithms have been proposed to solve related stochastic and robust motion planning in the presence of external disturbances and uncertainties. Some approaches to solve robust planning can be found in [6], [17], [18] and stochastic control could be found in [5], [10], [12]. However, most of the proposed algorithms can not provide any performance guarantees for safety. Recently, a sampling-based approach-evasion game formulation of the motion planning problem was proposed in [20]. A robust-adaptive motion planning algorithm in the presence of moving obstacles has been proposed in [22]. However, it is based on open-loop control and makes use of a model-based estimator.

Contributions: In this paper, we propose data-driven anytime algorithms for motion planning with safety and performance guarantees for the underlying autonomous robotic system. In particular, we use incremental sampling to construct the state-space of the system and synthesize a robust controller by solving an approach-evasion differential game to reach the target set in the presence of unmodeled uncertainty in the system. The system uses a statistical estimator to estimate the unmodeled uncertainty based on observations of system trajectories. These estimates are then used by the controller to refine the estimates of the value functions for the underlying differential game to improve performance while preserving safety guarantees. This allows the system to learn from data and find better policies for performance enhancement while maintaining the desirable safety guarantees which are critical in modern autonomous robotic systems like self-driving cars etc..

Literature. Reinforcement learning has been widely used to design feedback controllers for systems without knowing full dynamics of the underlying systems [14]. A lot of work has been done in reinforcement learning using different approaches like Hidden Markov Models, Bayesian Methods, Q-learning, Temporal Difference (TD) learning [4] etc.. However, a big concern in these algorithms is the transient learning phase when a bad initial policy could be disastrous for the autonomous systems. More recently, an elegant Model Predictive Control (MPC) formulation of the problem with safety guarantees was presented in [2]

† Devesh K. Jha and Asok Ray are with Mechanical & Nuclear Engineering Department, Pennsylvania State University, University Park, PA 16802, USA (email: {dkj5042,axr2}@psu.edu).

⋆ Minghui Zhu is with Electrical Engineering Department, Pennsylvania State University, University Park, PA 16802, USA and is partially supported by NSF CNS-1505664. (email: muz16@psu.edu).

‡ Yebin Wang is with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA (email: yebinwang@ieee.org).

using reachability analysis of the dynamical system and Tube MPC for robust, learning-based control for linear systems with asymptotic performance guarantees. Some Hamilton-Jacobi-Issacs (HJI)-based approaches have been presented for control applications in [1], [9]. The basic idea in these papers is to first calculate the discriminating kernel and let the system explore and find better policies inside the discriminating kernel. However, there are no performance guarantees.

II. Problem Formulation

Consider a non-linear dynamical system governed by the following differential equation:

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

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^N$ is the system state, and $u(t) \in \mathcal{U}$ is the control input of the system. Furthermore, $g : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}^N$ is locally Lipschitz continuous functions which represent the known part of the dynamics of the underlying system. The function $d : \mathbb{R}^N \to \mathbb{R}^N$ represents the unknown dynamics of the system. We assume that $d(x) \in \mathcal{D}$, where $\mathcal{D} \in \mathbb{R}^N$. For system (1), the set of admissible (feedback) control strategies is defined as:

$$\mathcal{U} \triangleq \{u(\cdot) \ : \ [0,+\infty) \to U, \ \text{measurable}\},$$

where $U \subseteq \mathbb{R}^M$. Denote by $\phi(\cdot; x, u, d) \triangleq \{\phi(t; x, u, d)\}_{t\geq 0}$ the solution to system (1) given the initial state x, the controls of u and the unmodeled dynamics d.

Throughout this paper, we impose the following assumption on system (1):

Assumption 2.1: The following properties hold:

(A1) The sets $\mathcal{X}$, U and $\mathcal{D}$ are compact.

(A2) The disturbance function $d(x)$ is locally Lipschitz continuous.

(A3) The function f is continuous in (x,u,d) and Lipschitz continuous in x for any $(u,d) \in U \times \mathcal{D}$.

(A4) For any pair of $x \in \mathcal{X}$ and $u \in U$, $F(x,u)$ is convex where the set-valued map $F(x,u) \triangleq \cup_{d\in\mathcal{D}} f(x,u,d)$.

The problem of controller synthesis for the underlying system when $d(x)$ is unknown is formulated as an adaptive approach-evasion game with time-varying estimates on the disturbance. The objective of the controller is to maximize the performance of the dynamical system which is affected by some disturbance $d(x)$. Disturbance (which in our case is the unmodeled dynamics) wants to maximize the cost of the control input u. We formalize the aforementioned objectives as follows. Define by $t(x,u,d)$ the first time when the trajectory $\phi(\cdot; x, u, d)$ hits $\mathcal{X}_{\text{goal}}$ while staying in $\mathcal{X}_{\text{free}}$ before $t(x,u,d)$. More precisely, the functional $t(x,u,d)$ is defined as follows:

$$t(x,u,d) \triangleq \inf\{t \geq 0 \mid \phi(t;x,u,d) \in \mathcal{X}_{\text{goal}}, \\ \phi(s;x,u,d) \in \mathcal{X}_{\text{free}}, \ \forall \ s \in [0,t]\}.$$

If $\phi(\cdot; x, u, d)$ leaves $\mathcal{X}_{\text{free}}$ before reaching $\mathcal{X}_{\text{goal}}$ or never reaches $\mathcal{X}_{\text{goal}}$, then $t(x,u,d) = +\infty$. This formulation leads to a zero-sum differential game between two players i.e., the controller and the disturbance. We call it the time-optimal-approach-evasion (TO-AE) differential game.

We use the notion of non-anticipating or causal strategy in the sense of [21] to define the value of the TO-AE differential game. The set Γ^a of such strategies for the controller is such that $\gamma^a \ : \ \mathcal{D} \to \mathcal{U}$ satisfies for any $T \geq 0$, $\gamma^a(d(t)) = \gamma^a(d'(t))$ for $t \in [0,T]$ if $d(t) = d'(t)$ for $t \in [0,T]$. The lower value of the TO-AE differential game is given by:

$$T^*(x) = \inf_{\gamma^a(\cdot)\in\Gamma^a} \sup_{d(\cdot)\in\mathcal{D}} t(x, \gamma^a(d(\cdot)), d(\cdot)).$$

The function T^* is then referred to as the minimum time function. It is noted that $t(x,u,d)$ is potentially infinite and this may cause numerical issues. To deal with this, we normalize the hitting time by the Kružkov transform $\Psi(r) = 1 - e^{-r}$. With this nonlinear transform, we further define the discounted cost functional $J(x,u,d) = \Psi \circ t(x,u,d)$, and the discounted lower value v^* as follows:

$$v^*(x) = \inf_{\gamma^a(\cdot)\in\Gamma^a} \sup_{d(\cdot)\in\mathcal{D}} J(x, \gamma^a(d(\cdot)), d(\cdot)).$$

One can easily verify that $v^*(x) = \Psi \circ T^*(x)$ for $\forall x \in \mathcal{X}$. We will refer v^* to as the optimal value function.

Next under the TO-AE game setting, we allow the system to learn from observations and thus use new estimates on the disturbance function to make more accurate and better estimates on the minimum time function. To achieve this, we use a machine learning(ML)-based statistical estimator to estimate the function $d(x)$. Making use of observations for predicting the disturbance function results in inherent probabilistic estimates such that the point-wise functional estimates lie in some interval say $\tilde{\mathcal{D}}_k(x)$ at kth iteration with significance level $(1-\alpha)$ for $\alpha \in (0,1)$. A sufficient condition for the safety of the underlying system in the presence of the disturbance is that $d(x)$ lies in the estimated set $\tilde{\mathcal{D}}_k(x)$ for all $k \in \mathbb{N}$. While a possible solution to guarantee safety is to use the conservative bounds; our objective here is to estimate $d(x)$ or at least a tighter bound represented by the set $\tilde{\mathcal{D}}_k(x)$ so that we can improve system performance while maintaining provable-safety guarantees. The objective here is to optimize system performance by updating modeling uncertainties or exogenous disturbance interfering with the system while retaining safety guarantees. The difference from classical control theory-based robust adaptive control is that we use a machine learning-based statistical estimator (model-free) instead of a Kalman filter-type estimator.

Thus, our problem consists of two steps.

1) Making tighter estimates on the bounds of the disturbance function using a statistical method from observations on system trajectories.
2) Use the statistical estimates to make better estimate of minimum time function for the underlying approach-evasion game.

III. Data-Driven Anytime Robust-Adaptive Algorithm

In this section, we present the algorithms and ideas for solving the learning-based TO-AE differential game. We allow the controller and estimator to run in parallel, independent of each other using the separation principle. The

key idea is that we ensure the safety of the system by solving the TO-AE game which calculates the discriminating kernel [3], [20] as well the optimal controller simultaneously for the underlying system; we further refine the controller by getting new updates on the disturbances of the system using bootstrapping.

A. Machine Learning-based Estimator

We focus on the special case where we can measure all the states of our system and assume that there is no measurement noise. The key idea here is that to guarantee safety of the underlying system, we need to have deterministic convergence guarantees on the estimates provided by the statistical regression algorithm. While such guarantees might be elusive with finite amount of data, it is possible to estimate generative functions along with point-wise confidence intervals [8]. We try to estimate the unmodeled dynamics using non-parametric kernel-based regression technique.

To estimate the unmodeled dynamics of the underlying system, we first numerically calculate the system state derivatives using observations on the system state and then, we calculate the residuals using the known part of the system model. More formally, the residuals are calculated as

$$\bar{d}(x) = \bar{g}(x,u) - g(x,u) \tag{2}$$

where $\bar{g}(x,u)$ represents the numerically calculated gradient using the observations on system trajectories and $g(x,u)$ is the known part of the system dynamics. The term $\bar{d}(x)$ in equation (2) represents anomaly in the system observation which can't be explained using the known model and is thus considered to be present due to the unmodeled disturbance (dynamics). These residuals are then used to predict the unknown disturbance function using a least square support vector machine (LS-SVM) regression algorithm. Along-with estimating the underlying unknown function we also calculate the point-wise confidence bounds for the estimates. A least square support vector regression algorithm with confidence intervals was presented in [8]. We use the algorithms presented in [8] to estimate the unknown disturbance function. We use the following notation to describe the LS-SVM regression: the data set at any epoch n is the set $\{(X_1,Y_1),\ldots,(X_n,Y_n)\}$ where $Y_i = \bar{d}(x_i)$ and $X_i = x_i$. Thus, X_i's are the independent state variables and Y_i's represent the corresponding observations corresponding to X_i's. The goal is to find the underlying generative function, with probabilistic bounds on accuracy of the same. The motivation behind using the LS-SVM regression is to be able to find the disturbance function with probabilistic bounds using confidence intervals for the function.

For the completeness of the paper, we very briefly describe the key idea behind the regression-based estimation process. In particular, we model our data is being generated by $Y = d(X) + \sigma(X)\varepsilon$, where $\mathbf{E}[\varepsilon|X] = 0$, $\mathbf{Var}[\varepsilon|X] = 1$ and X & ε are independent (Y is the observation and X represents the independent variable in the domain of the unknown function). What we are interested is an estimate of the function $d(x)$ (we denote it by $\hat{d}_n(x)$ where n denotes the number of observations) and the corresponding confidence interval for $d(x)$ i.e., given $\alpha \in (0,1)$ we want to find a bound η_α such that $\mathbf{P}(\sup_{x\in X} |\hat{d}_n(x) - d(x)| \leq \eta_\alpha) \leq 1-\alpha$, where X is the domain of the function $d(\cdot)$. As such, we estimate the confidence interval for the unknown function $d(x)$ point-wise as well as the interval over the domain of the function. The LS-SVM problem is formulated as an optimization problem as follows

$$\min_{w,b,e} \mathcal{J}(w,b,e) = \frac{1}{2}w^T w + \frac{\gamma}{2}\sum_{i=1}^{n} e_i^2 \tag{3}$$

such that $Y_i = w^T\psi(X_i) + b + e_i\,, i = 1,2,\ldots,n$ where $e_i \in \mathbb{R}$ are assumed to be i.i.d. random variables with $\mathbf{E}[e|X] = 0$ and $\mathbf{Var}[e|X] < \infty$. We assume that the d is a smooth function and $\mathbf{E}[Y|X] = d(X)$, ψ is a feature map or kernel used in standard SVM. Based on the observations on the system trajectories, an estimate of the unknown function (denoted as $\hat{d}$) using LS-SVM regression is obtained as

$$\hat{d}(x) = \sum_{i=1}^{n} \hat{\alpha}_i K(x, X_i) + \hat{b} \tag{4}$$

where, $K : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ represents a kernel function (e.g., Gaussian or radial basis functions), m is the dimension. The terms $\hat{\alpha}$ are the dual Lagrange multiplier obtained by solving the LS-SVM problem. In the next step we calculate the linear smoother matrix L for the LS-SVM regression such that we can estimate the conditional mean and variance for the estimated unmodeled function. The conditional mean and variance of the estimated function is given by the following expressions.

$$\mathbf{E}[\hat{d}(x)|X = x] = \sum_{i=1}^{n} l_i(x) m(x_i) \tag{5}$$

$$\mathbf{Var}[\hat{d}(x)|X = x] = \sum_{i=1}^{n} l_i(x)^2 \sigma^2(x_i) \tag{6}$$

Then, the expected value or the bias is approximately calculated by the following equation (see Theorem 2 in [8] for a proof).

$$\widehat{\texttt{bias}}[\hat{d}(x)|X = x] = L(x)^T\hat{d} - \hat{d}(x) \tag{7}$$

where, $\hat{d} = (\hat{d}(X_1),\ldots,\hat{d}(X_n))$ and L is the smoother matrix. The variance of the estimates is then calculated using the following equation (for a proof see Theorem 3 in [8]).

$$\mathbf{Var}[\hat{d}(x)|X = x] = L(x)^T\hat{\Sigma}^2 L(x) \tag{8}$$

where, $\hat{\Sigma} = \texttt{diag}(\hat{\sigma}^2(X_1),\ldots,\hat{\sigma}^2(X_n))$ and L is the smoother matrix. The term $\hat{\sigma}^2(x)$ is calculated using the following equation.

$$\hat{\sigma}^2(x) = \frac{S(x)^T\texttt{diag}(\hat{\varepsilon}\hat{\varepsilon}^T)}{1 + S(x)^T\texttt{diag}(LL^T - L - L^T)}$$

where, $S(x)$ is smoother vector at an arbitrary point X such that $S : \mathbb{R}^m \to \mathbb{R}^n$ and $S1_n = 1_n$, $\hat{\varepsilon}$ represents the residuals (i.e., deviation from the bias term) and $\texttt{diag}(A)$ represents the diagonal terms of A expressed as a column vector. The LS-SVM guarantees that under some regularity conditions, the central limit theorem is valid and the following is

true asymptotically. More formally we state the following theorem.

Theorem 3.1: For finite variance of the residuals $\hat{\varepsilon}$ (calculated using equation (8)), the following is true.

$$\lim_{n\to\infty} \frac{\hat{d}_n(x) - \mathbf{E}[\hat{d}_n(x)|X=x]}{\sqrt{\mathbf{Var}[\hat{d}_n(x)|X=x]}} \xrightarrow{D} \mathcal{N}(0,1)$$

where $\xrightarrow{D}$ implies convergence in distribution.

Proof: Follows from Theorem 7.4 in [7]. ■

Then, the point-wise $(1-\alpha)$, where $\alpha \in (0,1)$, confidence interval for the unknown function $d(x)$ is given by the following expression

$$\hat{d}_n(x) - \widehat{\texttt{bias}}[\hat{d}_n(x)|X=x] \pm z_{1-\alpha/2}\sqrt{\mathbf{Var}[\hat{d}_n(x)|X=x]} \tag{9}$$

where, the $\widehat{\texttt{bias}}$ is the correction term given by equation (7) and $z_{1-\alpha/2}$ denotes the $(1-\alpha/2)$ quantile of the standard Gaussian distribution. For prediction at a new state $\tilde{x}$ the confidence interval is given by the following equation.

$$\hat{d}_n(\tilde{x}) - \widehat{\texttt{bias}}[\hat{d}_n(\tilde{x})|X=\tilde{x}] \pm z_{1-\alpha/2}\sqrt{\hat{\sigma}^2(\tilde{x}) + \mathbf{Var}[\hat{d}_n(\tilde{x})|X=\tilde{x}]} \tag{10}$$

With point-wise confidence interval for the statistical estimates, we can define the expected bound for disturbance with probabilistic significance level point-wise, i.e., the set $\tilde{\mathcal{D}}_k(x)$ which was introduced earlier in section II. Using the sets $\tilde{\mathcal{D}}_k(x)$, we can construct the function D_k which contains the point-wise bounds for disturbance functions corresponding to a chosen significance level. This function is then used for synthesizing the control law. The original disturbance function then lies in this interval with significance level of $(1-\alpha)$.

The statistical estimator can thus maintain an estimate of the unknown disturbance function with expected deviation from the actual function. Using the central limit theorem, the LS-SVM regression guarantees convergence of the expected deviation of the function to the standard Gaussian distribution. This allows us to calculate the disturbance function bounds with probabilistic confidence intervals. Then, the idea is that we use these bounds corresponding to significantly high confidence intervals to relax the safety constraints on the system for improvement of performance.

B. Controller Synthesis using Iterative Incremental Game Algorithm

In the proposed approach, we decouple the controller and estimator such that sampling of estimates by the controller is independent of sampling rate of data used by the sensors and the estimator. Let the supremum of the disturbance vector norm in the set D_k be denoted by r_k, i.e., $r_k = \sup_{x\in\mathcal{X}} |D_k(x)|$. The estimate r_k is allowed to vary with time and it is not assumed to be monotonic. However, to establish the results the results presented in the paper we make the following assumption.

Assumption 3.1: The norm of the disturbance function is upper bounded by R which gives a conservative estimate of the disturbance i.e., $|d(x)| < R$ for all $x \in R$.

The above assumption is not very restrictive in the sense that, in general, models with very high accuracy are available for engineered systems and a conservative bound for uncertainties involved with the environment could be approximated using statistical estimates of past experiences. However, this is important to ensure the safety of the system as otherwise the system might end up using policies which might result in very high penalties. With this structure on the estimates provided by the statistical estimator, a new TO-AE game is defined for a new estimate of the disturbance. This leads to a family of parametric differential game [19] in the sense that $v^\star_{r_k} \leq v^\star_{r_n}$ for any pair of $r_k \leq r_n$ and for all $x \in \mathcal{X}$ (where $v^\star_{r_i}$ denotes the estimates of optimal value functions parameterized by r_i). Thus when we get a new estimate of the disturbance, it can be used to solve a new TO-AE game parameterized by a new disturbance bound estimate using bootstrapping on the sampled graph, i.e., the values for the new game could be initialized by already existing estimates for the same parameterized by a different bound.

To begin with, we solve the approach-evasion game using the conservative bound on the unmodeled disturbance; thus we recover discriminating kernel and the corresponding optimal control inputs corresponding to the expected conservative disturbance bound. We estimate the unmodeled disturbance, based on the observations on system trajectory, using the the statistical estimator described in section III-A. With a new estimate on the bounds of the unmodeled dynamics, we can initialize a new approach-evasion game and solve to find new control law. However, the statistical estimates of the bounds of deviation from the actual disturbance function may not monotonically converge to normal distribution (equation (9) implies asymptotic convergence; however no convergence rates are provided). Here, we would like to point out that if the estimator can provide monotonic, deterministic estimates on the bounds, we can use them to refine the estimates on the value functions for the system while ensuring safety. With probabilistic bounds, a risk-averse strategy is to maintain a bank of controllers for safer control during transience.

However, the problem becomes more complex with stochastic time-varying estimates on the bounds of disturbance function deviation and it requires a better switching law to retain system safety. Nevertheless, if we can estimate the confidence intervals with high significance level, we can still ensure safety with very high confidence during transience. This motivates the use of confidence intervals with the statistical estimates. Since we have a conservative bound on the expected disturbance to the system, the actual discriminating kernel of the system corresponding to the actual disturbance function is thus a superset of the discriminating kernel calculated corresponding to the initial conservative bound. Thus the new estimates can still be used with low-confidence intervals to refine the value functions inside of the discriminating kernel and guarantee safety. At the boundaries of the discriminating kernel, using a bound with low confidence interval might have low safety

guarantees. This observation can be used to further improve the performance of the system in some regions of the state-space. However, we use the estimates with confidence intervals corresponding to high significance levels for deciding new control laws to avoid unnecessary switching. While a deterministic estimate would lead to maximum performance, probabilistic estimates leads to a trade-off in performance for safety. However, this is a restriction imposed by the statistical model-free estimator.

To decide the control law, a user-defined significance level for statistical estimates is used to find the confidence interval for the unmoded disturbance function. Then, suppose that the controller samples new estimate of the unmodeled disturbance at an epoch n. Then, the uncertainty with a significance level $(1-\alpha_{\text{CI}})$ (α_{CI} is user input) over the state-space is bounded by the following term, $r_n = \sup_{x\in\mathcal{X}} |D_n(x)|$.

The term r_n provides the bound on the expected deviation of the estimated function from the original disturbance function over its domain with a significance level $(1-\alpha_{\text{CI}})$. Then, a new game could be initiated which is parameterized by r_n. Then, we get a series of such estimates and the corresponding value function which we arrange in an ordered fashion. We represent the sets by $E_k = \{R, r_1, r_2, \ldots, r_k\}$ and $V_k = \{v^\star_R, v^\star_{r_1}, v^\star_{r_2}, \ldots, v^\star_{r_k}\}$ for the estimates and the corresponding value functions respectively where, $R \geq r_1 \geq r_2 \geq \cdots \geq r_j \geq \ldots$ and it follows that $v^\star_R \geq v^\star_{r_1} \geq v^\star_{r_2} \geq \cdots \geq v^\star_{r_j} \geq \ldots$. Then, when a new estimate r_n is sampled by the controller with confidence level α_{CI}, it is compared to the elements of the existing elements of E_{n-1} and a controller corresponding to $v^\star_{r_k}$ is used where $r_k \geq r_n \geq r_{k-1}$. Consequently, the new estimate to the set E_k and a new game parameterized by r_n is initiated with $v^\star_{r_k}$ (technically we recover the optimal costs asymptotically; with some abuse of notation we use the $v^\star_{r_k}$ to denote the converged numerical estimates). As the estimates for v_{r_n} converge, we switch to the control given by the same. With this we incrementally populate the set E_k and the set V_k. This process is repeated till the elements of set E_k converge. The controller is always decided by the latest estimates on the bounds of the disturbance function estimated by the ML-based estimator with significance level $(1-\alpha_{\text{CI}})$. The adaptive controller synthesis is presented in algorithms 1 through 4. It is noted that since the controller and estimator are decoupled, the iteration number for them are different. The estimator is collecting data and making estimates based on the sampling rate of the sensors.

The controller synthesis is based on the iGame algorithm earlier presented in [20] where the states-space for the robot is incrementally built by sampling from the free configuration space of the robot. The input set for the robot is also incrementally built by sampling from its input set. At every iteration of the algorithm, a single update of the value iteration is solved on the sampled graph for the corresponding pursuit-evasion game. Thus, by incremental sampling from the state-space a better refined discrete state-space for the robot is created and the estimates for the

Algorithm 1: Data-Driven Anytime Control

Require: Initially, use the conservative bound R to solve iGame with values v_R and $E_0 \leftarrow \{R\}$. Pick a sampling interval T_s to get a new estimate for $d(x)$ using Algorithm 4

Ensure: Repeat the following steps at every iteration

if $n \mod T_s = 0$ **then**
  $\texttt{flag} = 1$
else
  $\texttt{flag} = 0$
if $\texttt{flag} == 1$ **then**
  $k = n/T$;
  $E_k = E_{k-1} \cup \{r_k = \sup_{x\in\mathcal{X}} |D_m(x)|\}$;
  Find m s.t. $r_m \geq r_k \geq r_{m+1}$ and order the set $E_k = E_{k-1} \cup \{r_k\}$;
  Initialize v_{r_k} with values from v_{r_m} on S_{n-1};
  $V_k \leftarrow V_{k-1} \cup \{v_{r_k}\}$;
  $u_n(x) \leftarrow$ solution to u from v_{r_m};
  go to Algorithm 2;
else
  go to Algorithm 2

Algorithm 2: The iGame Algorithm

$y_n \leftarrow \texttt{Sample}(\mathcal{X}, 1)$;
$S_n \leftarrow S_{n-1} \cup \{y_n\}$;
$h_n \leftarrow \zeta_n^{\frac{1}{1+\gamma}}$;
$\gamma_n \leftarrow 2\zeta_n + \ell h_n \zeta_n + M\ell h_n^2$;
$v_{n-1} = v_{r_m}$;
for $x \in S_{n-1}$ **do**
  $\tilde{v}_{n-1}(x) = v_{n-1}(x)$;
$\tilde{v}_{n-1}(y_n) = 1$;
for $x \in K_n \subseteq S_n \setminus \mathcal{B}(\mathcal{X}_{\text{goal}}, Mh_n + \zeta_n)$ **do**
  $(v_n(x), u_n(x)) \leftarrow \texttt{VI}(S_n, \tilde{v}_{n-1})$;
for $x \in S_n \setminus \big(K_n \cup \mathcal{B}(\mathcal{X}_{\text{goal}}, Mh_n + \zeta_n)\big)$ **do**
  $v_n(x) = \min_{y\in\mathcal{B}(x,\gamma_{n-1})\cap S_{n-1}} v_{n-1}(y)$;
for $x \in S_n \cap \mathcal{B}(\mathcal{X}_{\text{goal}}, Mh_n + \zeta_n)$ **do**
  $v_n(x) = \tilde{v}_{n-1}(x)$;

underlying pursuit-evasion game is further refined by solving value iteration once after a new sample and input is added to the corresponding state-space graph and input sets. Interested readers are referred to [20] for more details of the iGame algorithm. The iGame algorithm is anytime and guarantees asymptotic optimality for the robust motion planning with known disturbance bounds. Some of the notations in the iGame algorithm are defined as follows. The state dispersion ζ_n is the quantity such that for any $x \in \mathcal{X}$, there exists $x' \in S_n$ such that $\|x - x'\| \leq \zeta_n$. The other quantities for time discretization are defined as $h_n = \zeta_n^{\frac{1}{1+\gamma}}$ and $\kappa_n = h_n - \zeta_n$.

IV. Performance Analysis

In this section, we discuss the performance of the data-driven anytime algorithms. The controller is anytime as the iGame algorithm which is used to calculate the system policy is anytime and thus can be terminated anytime after the sampled graph reaches the goal set of the robot. As the robust approach evasion game is solved with a conservative bound on the disturbance function, the existence of optimal

Algorithm 3: $\texttt{VI}(S_n, \tilde{v}_{n-1})$

$U_n \leftarrow U_{n-1} \cup \texttt{Sample}(U, 1)$;

$v_n(x) \leftarrow 1 - e^{-\kappa_n} + e^{-\kappa_n} \max_{c_{m,\alpha_{\mathrm{CI}}} \in D_m} \min_{u \in U_n} \min_{y \in \mathcal{B}(x+h_n f(x,u,d),\gamma_n) \cap S_n} \tilde{v}_{n-1}(y)$;

$u_n(x) \leftarrow$ the solution to u in the above step;

Algorithm 4: Confidence Intervals

Input: The observation set $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ and significance level α_{CI}

Output: $\hat{d}_n(x)$ and the expected confidence intervals with significance level $(1 - \alpha)$

Given the data $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, calculate $\hat{d}$ using the equation (4);

Calculate the expectation or bias using equation (7);

Calculate the residuals as $\hat{\varepsilon}_k = Y_k - \hat{d}_k$, $k = 1, \ldots, n$;

Calculate the variance using equation (8);

Set the significance level $\alpha = \alpha_{\mathrm{CI}}$;

Calculate the point-wise confidence intervals using equations (9) and (10), say $c_{n,\alpha_{\mathrm{CI}}}(x)$;

Calculate the set D_n where $D_n(x) = c_{n,\alpha_{\mathrm{CI}}}(x)$;

policies leading to the goal set is guaranteed from the discriminating kernel of the dynamical system parameterized by the conservative bound. The asymptotic performance of the anytime algorithm is guaranteed under the assumption that the statistical estimates on the disturbance function converge (these estimates converge in probability; thus, we need the convergence of the variance for the estimates described earlier). Suppose that the error bounds on the statistical estimates converge to asymptotic bound denoted by $r^\star$. Then, the the values corresponding to the game parameterized by the sequence of bounds $\{r_k\}_{k \in \mathbb{N}}$ converges to the game with parameter $r^\star$, i.e., the following is true point-wise $\lim_{k \to \infty} \|v^\star_{r_k} - v^\star_{r^\star}\|_{\mathcal{X}} = 0$ (where the norm is defined as $\|v\|_S = \sup_{v \in S} \|v\|$). This follows from Theorem 3.1 in [20]. This shows that our controller is asymptotically optimal and safe, as the solutions are optimal for the underlying approach-evasion games.

During the transient behavior, i.e., when the estimates of the value functions and thus the discriminating kernel estimates are also based on the statistical estimates and the corresponding confidence intervals for deviation from the original function, the guarantees are based on the confidence interval and we provide probabilistic optimality with a high degree of confidence. During the transient phase, our guarantees for safety are based on the probability that the actual disturbance function lies inside the confidence interval provided by the statistical estimator.

V. Conclusions and Future Work

In this paper, we presented some initial results for data-driven anytime motion planning of robotic systems where we use observations on system trajectories to statistically estimate the unknown system dynamics which is then used to improve the controller performance while retaining system safety guarantee. The safety guarantees for the system during the transient phase in this paper are based on the confidence intervals provided by the statistical regression algorithm.

Use of recursive techniques for regression-based estimation of the unmodeled disturbance is a topic of future research.

References

[1] A. K. Akametalu, S. Kaynama, J. F. Fisac, M. N. Zeilinger, J. H. Gillula, and C. J. Tomlin, "Reachability-based safe learning with Gaussian processes," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 1424–1431.

[2] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[3] J.-P. Aubin and H. Frankowska, *Set-valued analysis*. Springer Science & Business Media, 2009.

[4] D. P. Bertsekas and J. N. Tsitsiklis, "Neuro-dynamic programming: an overview," in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 1. IEEE, 1995, pp. 560–564.

[5] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *Robotics, IEEE Transactions on*, vol. 26, no. 3, pp. 502–517, 2010.

[6] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 723–730.

[7] K. De Brabanter, "Least squares support vector regression with applications to large-scale data: a statistical approach," 2011.

[8] K. De Brabanter, J. De Brabanter, J. A. Suykens, and B. De Moor, "Approximate confidence and prediction intervals for least squares support vector regression," *Neural Networks, IEEE Transactions on*, vol. 22, no. 1, pp. 110–120, 2011.

[9] J. H. Gillula and C. J. Tomlin, "Reducing conservativeness in safety guarantees by learning disturbances online: iterated guaranteed safe online learning," *Robotics: Science and Systems*, 2012.

[10] V. A. Huynh, S. Karaman, and E. Frazzoli, "An incremental sampling-based algorithm for stochastic optimal control," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2865–2872.

[11] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International Journal of Robotics Research*, p. 0278364915577958, 2015.

[12] D. Jha, Y. Li, T. Wettergren, and A. Ray, "Robot path planning in uncertain environments: A language measure-theoretic approach," *ASME Journal of Dyn. Sys., Meas., Control*, vol. 137, p. 0345003, 2015.

[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[14] J. Z. Kolter and A. Y. Ng, "Policy search via the signed derivative." in *Robotics: science and systems*, 2009.

[15] S. LaValle, "Motion planning," *Robotics Automation Magazine, IEEE*, vol. 18, no. 1, pp. 79–89, March 2011.

[16] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[17] B. Luders, M. Kothari, and J. P. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *AIAA guidance, navigation, and control conference (GNC), Toronto, Canada*, 2010.

[18] A. Majumdar and R. Tedrake, "Robust online motion planning with regions of finite time invariance," in *Algorithmic foundations of Robotics X, Springer Berlin Heidelberg*, 2013, pp. 543–558.

[19] E. Mueller, S. Z. Yong, M. Zhu, and E. Frazzoli, "Anytime computation algorithms for stochastically parametric approach-evasion differential games," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 3816–3821.

[20] E. Mueller, M. Zhu, S. Karaman, and E. Frazzoli, "Anytime computation algorithms for approach-evasion differential games," *arXiv preprints*, 2013. [Online]. Available: http://arxiv.org/abs/1308.1174

[21] P. Varaiya, R. Elliott, E. Roxin, and N. Kalton, "The existence of value in differential games," *American Mathematical Society*, no. 126, 1972.

[22] N. Virani and M. Zhu, "Robust adaptive motion planning in the presence of dynamic obstacles," in *2016 American Control Conference*, 2016, To Appear.