Robust Adaptive Motion Planning in the Presence of Dynamic Obstacles

Nurali Virani and Minghui Zhu

Abstract— Usually in game theoretic formulations for robust motion planning, the model as well as the capabilities (input set) of all dynamic obstacles are assumed to be known. This paper aims to relax the assumption of known input set by proposing a unified framework for motion planning and admissible input set estimation. The proposed approach models every dynamic obstacle as an uncertain-constrained system and then uses the uncertainty estimation technique to estimate the bounds of those uncertainties. The RRT \(^*\) algorithm with uncertainty estimation for robust adaptive motion planning in presence of dynamic obstacles is presented in this paper. Simulation examples have been used to validate the proposed algorithm.

Keywords: Motion planning, adaptive control, pursuit-evasion games, uncertainty estimation.

I. INTRODUCTION

Motion planning for intelligent autonomous systems has been addressed in literature over the years. Since, it is known that even the most basic version of motion planning problem, called the piano movers problem, is PSPACE-hard [1], research efforts to explore sampling-based techniques have gained interest. The planning algorithms such as Rapidly-exploring Random Trees (RRT) [2] and Probabilistic Roadmaps (PRM) [3] successfully introduced sampling algorithms for motion planning. RRT \(^*\) algorithm [4] was developed from RRT to guarantee asymptotic optimality. Recently, a RRT\(^*\)-based robust motion planner was presented in [5] to address planning in presence of dynamic obstacles with known uncertainty sets. This paper aims to extend that work to address the problem in which the input set associated with each dynamic obstacle is unknown to the planner. The proposed planner estimates the uncertainties with observations while planning and adapts to the new information to deliver robust motion plans. Thus, a new robust adaptive motion planner is presented in this paper.

In scenarios having multiple agents [6]–[8], motion planning in the joint state space leads to an explosion in the number of states. Recently techniques have been developed to address decentralized multi-agent motion planning [9], [10], which naturally lead to game-theoretic formulations to address non-cooperative scenarios. In game theory, pursuit-evasion problems have been studied over many decades for motion planning with other dynamic agents [11]. In such a pursuit-evasion game, the evading agent is the first player, it has to come up with a open-loop strategy to reach a goal set. Since this strategy is known to the pursuing agents (second player) before they choose their own best strategy to catch the agent, the choice of initial strategy needs to consider the capabilities of each pursuing agent. In [5], the uncertainty in control inputs of all other agents is considered for obtaining the motion plan with incremental sampling, thus it is said to be a robust motion planning technique. It has asymptotic guarantees of probabilistic completeness (i.e., to find a path, if one exists to reach the goal), probabilistic soundness (i.e., to find a safe path, if one exists to avoid capture), and optimality (i.e., to find the minimum-time path, if feasible path set is non-empty).

Several game theoretic formulations, including [5], assume that the initial state, the model and the input set of all the dynamic agents is known. However, not having this assumed information can arise in several real scenarios, including intelligent vehicles in urban traffic as well as surveillance/tracking with mobile agents. Robust planning under uncertainty for linear Gaussian systems, due to localization error, process noise, and uncertain environmental constraints was addressed by developing Chance-constrained Rapidly-exploring Random Trees (CC-RRT and CC-RRT\(^*\)) [12]. Bry and Roy in [13] presented the Rapidly-exploring Random Belief Trees technique to plan in presence of state uncertainties. In [14], the additive uncertainty in the agent’s motion model is captured by statistical learning techniques for planning with safety guarantees. However, in this work, we aim to relax the crucial assumption of known input set from the framework of planning with other dynamic obstacles. We propose a technique by using ideas of fast uncertainty estimation [15]–[17] to address this motion planning problem.

In order to use existing techniques, one quick fix is to use a conservative input set for all other agents and robust solutions can still be obtained. However, if the considered input set is too large as compared to true set, the worst-case analysis from the game-theoretic approach might yield very conservative results leading to poor performance and in few cases, the feasibility can also be lost. On the other hand, if the agent has on-board sensors to observe the other dynamic agents, the proposed technique uses the observations to estimate their input/uncertainty sets and in parallel, uses these estimates to incrementally plan motion with respect to some initial state of all the dynamic agents. Thus, this technique in the current proposed form can be applied in scenarios where the initial states are known and the agents always start from that location. This would arise in several applications for military, such as infiltration and escape, as well as for autonomous driving in highway scenarios. The proposed algorithm inherits the guarantees of the robust
planning technique [5] of probabilistic completeness, probabilistic soundness, and asymptotic optimality, if the proposed uncertainty estimator converges in finite time.

In summary, the main contributions of the paper are:

- Development of a robust adaptive incremental sampling–based technique for motion planning in presence of dynamic obstacles.
- Formulation of a uncertainty set bound estimation scheme using a fast estimator from adaptive control.
- Validation of the proposed method by simulation experiments of a pursuit–evasion game.

The rest of the paper is organized into five more sections. Section II presents the mathematical formulation of the problem. In Section III, the uncertainty estimation framework and its relation to the motion planning approach is given. Section IV describes all the procedures, functions, and the proposed algorithms for the developed approach. Section V outlines the simulation setup and presents the simulation results. Lastly, in Section VI, some concluding remarks are made and future research directions are discussed.

II. PROBLEM FORMULATION

We consider an agent $A$ in a known static environment. Let, the state of the agent at time $t$ be denoted as $x_A(t) \in \mathcal{X}_A$ and the known dynamic model of the agent be given by:

$$ \dot{x}_A(t) = f_A(x_A(t), u_A(t)), $$

(1)

where $u_A(t)$ is the control input from the admissible set of controls $\mathcal{U}_A$. The environment also has a dynamic obstacle $O$ and the state of the obstacle at time $t$ is given by $x_O(t) \in \mathcal{X}_O$. Let the static obstacle free configuration space be denoted as $\mathcal{X}_A^{free}$ for the agent and as $\mathcal{X}_O^{free}$ for the obstacle. The initial state for agent $x_A(0) \in \mathcal{X}_A^{free}$, the initial state for obstacle $x_O(0) \in \mathcal{X}_O^{free}$, and a goal set for the agent $\mathcal{X}_{goal} \subset \mathcal{X}_A^{free}$ are known. The dynamic obstacle is modeled as an uncertain-constrained system and the corresponding dynamic model is given as:

$$ \dot{x}_O(t) = f_O(x_O(t)) + w(t), $$

(2)

where the uncertainty is captured by $w(t)$. The uncertainty function $w(t)$ and its first derivative are assumed to be uniformly bounded, i.e., there exists a compact set $\Omega$ and two positive constant $r_1, r_2 \in \mathbb{R}^+$, such that, $w(t)$ lies in the compact set $\Omega = \{w : \|w\| \leq r_1\}$ and its derivative is bounded by $r_2$, that is, $\|\dot{w}(t)\| \leq r_2$.

If the uncertainty function is considered as the unknown control input for the dynamic obstacle (assuming controllability holds for the obstacle model), the admissible control input set $\mathcal{U}_O$ is defined by the set of all inputs $w(t)$, that is, $\mathcal{U}_O = \Omega$. The RRT$^*$ framework [4] could have completely addressed this robust motion planning problem using the technique presented in [5], if the uncertainty set was perfectly known. In this problem setup, the bound of the uncertainty set, i.e., $r_1$ is unknown; however, a conservative estimate of the upper bound $\hat{r}_1 \geq r_1$ is available, thus we have $\overline{\Omega} = \{w : \|w\| \leq \hat{r}_1\}$ as the conservative estimate of the uncertainty set. Additional information to address the uncertainty estimation problem comes from an on-board sensor with the agent, which can observe the states of the obstacle. The sampling period of this sensor is given by $T_s$ and thus, the $k^{th}$ measurement is given as $y_k = x_O(kT_s)$. The proposed technique to address the uncertainty estimation problem is explained in the next section.

III. ADAPTIVE ESTIMATION FRAMEWORK

The goal of this presented work is to implement in parallel a framework which uses the concept of uncertainty estimation from the field of adaptive control [17]. The sensor observations would be used in order to improve the estimate of the uncertainty set $\Gamma(t) = \{w : \|w\| \leq \hat{r}_1(t)\}$. The updated admissible control input set will then be used within the incremental-sampling based planner to compute better paths. The proposed solution is presented ahead by first presenting the uncertainty estimation and the uncertainty set estimation process. These estimation processes are then used with the RRT$^*$-based pursuit–evasion algorithm to develop a motion planning with uncertainty estimation framework.

A. Fast Uncertainty Estimation

Consider the obstacle dynamic model in (2), where the uncertainty $w(t)$ is unknown. A fast estimator [15], [16] is included to estimate the uncertainty. This estimator consists of a state estimator and a piecewise constant adaptation law. The estimator is defined as follows:

$$ \dot{z}(t) = -a \dot{x}_O(t) + f_O(x_O(t)) + \hat{w}(t), $$

(3a)

$$ z(0) = x_O(0), $$

(3b)

where $a \in \mathbb{R}^+$ is an arbitrarily chosen positive constant and $\dot{x}_O(t) = z(t) - x_O(t)$ is the state estimation error. The signal $\hat{w}(t)$ is the estimate of the uncertainty and it is updated using a piecewise constant adaptation law, given as follows:

$$ \hat{w}(t) = -\Phi(T_s)\hat{x}(kT_s), $$

(4)

for $t \in [kT_s, (k + 1)T_s)$, where $\Phi(T_s) = \frac{a}{s^2 + 1}$ and $T_s$ is the sampling time of the sensor. This is a fast estimator, because the uncertainty estimation error can be made to be arbitrarily small by sampling fast enough. From [15] (See Proposition 4.1 in [15]), the performance of this estimator is given in the Remark 1. The proof of this result was shown in [16] (See Lemma 1 in [16]).

**Remark 1: Performance of Uncertainty Estimator:**

If the constrained uncertainty signal satisfies $\|w(t)\| \leq \hat{r}_1$, $\|\dot{w}(t)\| \leq r_2$, and the uncertainty estimator is given by (3) and (4), then the inequality:

$$ \|w(t) - \hat{w}(t)\| \leq \gamma(T_s), $$

(5)

holds for any $t \geq T_s$, where,

$$ \gamma(T_s) = 2r_2T_s + \|1 - e^{-aT_s}\|\hat{r}_1. $$

(6)

Note that, $\gamma(T_s) \to 0$, as sampling time $T_s \to 0$.

The uncertainty estimator is used in this work to estimate the uncertainty for the most recent sampling interval in the model of all dynamic obstacles.
B. Uncertainty Set Estimation

The uncertainty set is given as $\Omega = \{w : \|w\| \leq r_1\}$. Thus, the bounds of the set $\Omega$ needs to be obtained using the uncertainty estimates from the estimator discussed in Section III-A. Let, $\hat{r}_1^k$ denote estimate of the bound $r_1$ of uncertainty after incorporating $k^{th}$ measurement, then the update law for this estimate is proposed to be the following:

$$\hat{r}_1^k = \max(\hat{r}_1^{k-1}, \min(\|\hat{w}(kT_s)\| + \gamma(T_s), \hat{r}_1)),$$

$$\hat{r}_1^0 = 0,$$

for $k \in \mathbb{N}$, where $\gamma(T_s)$ is as given in (6), $\hat{w}(kT_s)$ is output of the estimator given in (4) at $t = kT_s$, and $\hat{r}_1$ is the known conservative bound of the uncertainty. The update rule ensures that the sequence of estimates are bounded above by $\hat{r}_1$ and monotonically increasing, thus by using the Monotone Convergence Theorem, the sequence converges. The limit point of the sequence is characterized by Theorem 1.

**Theorem 1:** Let $\hat{r}_1^*$ be the limit of the convergent sequence given by $\{\hat{r}_1^0, \hat{r}_1^1, \hat{r}_1^2, \ldots\}$, that is, $\hat{r}_1^* = \lim_{k \to \infty} \hat{r}_1^k$. If the true bound is denoted by $r_1$, then given the conservative bound $\hat{r}_1$, the limit point satisfies the following relations:

1) $r_1 \leq \hat{r}_1^* \leq \hat{r}_1$

2) $0 \leq \hat{r}_1^* - r_1 \leq 2\gamma(T_s)$.

**Proof of Theorem 1:** See Appendix.

**Remark 2:** Implication of the relation 1 of Theorem 1

The estimate obtained from the proposed approach is less conservative than the known bound $\hat{r}_1$, in general. However, if the conservative bound is the true bound, i.e., $r_1 = \hat{r}_1$, then the estimator also converges to the same value.

**Remark 3:** Implication of the relation 2 of Theorem 1

Irrespective of the value of the conservative bound $\hat{r}_1$ used in (7), the error in estimate of $r_1$ can be made arbitrarily small by sampling fast enough. However, once a sampling time $T_s$ has been fixed, the bound $r_1$ is at most over-estimated by $2\gamma(T_s)$. It is noted that, in order to obtain safe plans, the update rule has been designed to ensure that the limit point does not underestimate the bound.

**Remark 4:** Time to reach convergence

The convergence time for the estimator does not depend on any parameter chosen in (3), (4), and (7), but on how the obstacle chooses to execute the different available actions. If it executes the motion with some time period $T_O$, then number of measurements required to reach convergence is less than $\kappa_O$, which is the minimum value satisfying $T_O < \kappa_O T_s$. In other cases, eventually we would see every action that obstacle can do, thus get the convergence asymptotically.

**Remark 5:** Monotonicity of estimated uncertainty set

Let the estimated uncertainty sets after $k^{th}$ update be given as $\hat{\Omega}^k = \{w : \|w\| \leq \hat{r}_1^k\}$, then as a direct consequence of monotonicity and boundedness of the estimates $\hat{r}_1^k$, we have:

$$\{\phi\} \subset \hat{\Omega}^{k-1} \subset \hat{\Omega}^k \subset \hat{\Omega}.$$  

C. Motion Planning with Uncertainty Estimation

Consider a na"ive attempt to use the presented uncertainty estimation with motion planning, in which the estimator given in Section III-A and Section III-B is used for some finite time and the resulting estimate of uncertainty set is used in the motion planning framework for pursuit-evasion as presented in [5]. This would lead to optimal paths from RRT*, which are not guaranteed to be safe as the uncertainty set might have been under-estimated in that finite time window which was dedicated for uncertainty estimation (See Remark 4). Safe and optimal paths can only be guaranteed, if both the number of measurements and number of iterations of the motion planner go to infinity. Thus, instead of this na"ive sequential approach, an approach which unifies the two techniques for pursuit evasion with uncertainty estimation is presented. This section aims to present the core idea of this developed approach which enables to unify uncertainty estimation and incremental sampling-based pursuit evasion.

The uncertainty bound update rule in (7) gives a non-decreasing estimate of the bound. Thus, the uncertainty set, which in this case is also the set of admissible inputs for the obstacle, grows monotonically (Remark 5). In incremental sampling for pursuit-evasion [5], as the estimated set of states of pursuers grows monotonically, the safely reachable region of the evader is pruned down. Similarly, in the proposed approach, the safely reachable region of evader is pruned as the time-constrained reachability set grows for the pursuers; however, this growth is also attributed to the monotonic update of the admissible input set for the pursuers. Thus, uncertainty estimation can seamlessly be accommodated in the incremental sampling approach for motion planning. The algorithms for unifying motion planning and uncertainty estimation are presented ahead in the next section.

IV. Algorithm

This section presents and explains the proposed incremental sampling algorithm for pursuit-evasion with uncertainty estimation. This algorithm banks on the RRT* algorithm [4] and is closely related to the pursuit evasion algorithm [5]. The existing primitive procedures used in the algorithm are explained in Section IV-A for completeness of the paper and then, proposed modifications as well as new procedures are explained in Section IV-B.

A. Existing Primitive Procedures

The procedures mentioned in this section have been previously formulated in literature and detailed explanation for these procedures can be found in [4], [5]. Note that, the procedures with subscript $\alpha \in \{A, O\}$ perform differently for agent A, i.e., evader, and for obstacle O, i.e., pursuer.

**Sampling:** The function $\text{Sample}_\alpha : \mathbb{N} \to X_\alpha$ returns independent identically distributed (i.i.d.) samples from the obstacle-free space $X^f_\alpha$.

**Collision Checking:** The boolean output function $\text{CollisionFree}_\alpha(x_\alpha)$ returns true, if and only if the state trajectory $x_\alpha$ lies completely in the obstacle-free region of the static environment.

**Nearest Neighbor:** Given a tree $G_\alpha = (V_\alpha, E_\alpha)$ on $X_\alpha$ and a state $z \in X_\alpha$, the procedure $\text{Nearest}_\alpha(G_\alpha, z)$ returns the vertex $z^\text{nr} \in V_\alpha$ closest to $z$ using the Euclidean metric.
Algorithm 1: Extendα(Gα,z, r1)
1 V′ ← Vα, E′ ← Eα
2 znearest ← Nearestα(Gα,z)
3 (xnew,unew,Tnew) ← Steerα(znearest,z, r1)
4 znear ← znearest
5 if CollisionFreeα(znear) then
6 V′ ← V′ ∪ {znear}
7 znear ← znear
8 for all zn ∈ znear do
9 if CollisionFreeα(zn) and znear = zn then
10 E′ ← E′ ∪ {(znear, zn)}
11 znear = znear
12 if Timeα(znear) + c(znear) < Timeα(znew) then
13 znear = znear
14 znear = znear
15 znear = znear
16 znear = znear
17 znear = znear
18 znear = znear
19 znear = znear
20 znear = znear
21 znear = znear
22 else
23 znear = znear
24 return G′ = (V′, E′, znear)

Nearby Vertices: Given a tree Gα = (Vα, Eα) on Xα, a state z ∈ Xα ⊂ Rnα, and a number n ∈ N, the Nearby(G,v) returns all vertices in Vα that are in the ball Bγ(r, z), where r = n(log n/n)1/nα, for some predetermined constant γ > 0.

Time taken: Given a vertex v ∈ Vα, the function Timeα(v) returns the time to reach that vertex from the root along the unique path in the tree Gα. Given a state trajectory xα : [0, t] → Xα, the function c(xα) returns the end time t.

Near-Capture Vertices: The Nearbyα works very similar to Nearby. The function Nearbyα(G,v) returns all vertices v ∈ Vα that are at most distance rα away from being captured by from z.

Remove Vertex: The Remove(G,v) function removes the vertex v from the graph G along with all descendants of that vertex in the tree. This is useful for pruning the agent tree G as the safely reachable set is being estimated by avoiding regions where capture is imminent.

B. New Procedures

In this paper, we have slightly modified two procedures from the existing literature to include an extra input argument, which is the bound of the admissible input set of the pursuer. The modified procedures are the Steerα function and Extendα function (See Algorithm 1)

Steering: The Steerα(z1,z2, r1) returns a time-optimal state trajectory xα : [0, t] → Xα, input sequence uα : [0, t] → Uα(r1), and end time t, such that, xα(t) = fα(xα(0)) + w(t), xα(0) = z1, xα(t) is near z2, and w(t) ∈ Uα(r1) = {w : ∥w∥ ≤ r1}. However, the Steerα(z1,z2, r1) function neglects the third argument, as the admissible control set Uα is known.

Let UEflag be a boolean variable which is set to be true if the uncertainty estimation framework must be used in the planning algorithm. If we set the boolean variable UEflag to be false in Algorithm 2, then the above procedures and the Extend procedure would be used to address the pursuit evasion problem, as developed in [5], by using the conservative bound r1 to define the admissible input set Uα. If this variable is set to be true, then we need to newly define three more procedures to accomplish pursuit evasion with uncertainty estimation. The theory related to these procedures has been explained in Section III-A and Section III-B. These new procedures are as follows:

Measurement Collection: The function Measure : N → Xα, given k ∈ N collects a new measurement from the sensor to observe the full state of the pursuer at time kT, where T is the sampling time of the sensor. Thus, the output is yk = xα(kT).

Uncertainty Estimation: Given measurement sample number k ∈ N, previous measurement yk−1, new measurement yk, and previous state estimate sk−1, the function UE simulates the fast uncertainty estimator for t ∈ [(k−1)T, kT], using the following equations:

\[ \dot{z}(t) = -a z(t) + a y_{k-1} + f_0(y_k) + \dot{w}(t), \]
\[ z((k-1)T) = y_{k-1}, \]
\[ \dot{w}(t) = -\Phi(T_s)(s_{k-1} - y_{k-1}). \]

The output of the function is the final value of the state estimate sk = z(kT) and the updated uncertainty estimate.
\[ \hat{w}_k = -\Phi(T_s) (s_k - y_k). \] Here, \( \Phi(T_s) = a e^{-aTs} \) and \( a \) is a positive constant. This step estimates uncertainty in the pursuer’s system in the most recent sampling interval.

**Update Uncertainty Bound:** In the function UpdateBound, the uncertainty bound is updated by using the previous estimate of the bound \( \hat{r}_k \) and new estimate of uncertainty \( \hat{w}_k \) with the update law in (7).

The uncertainty estimation block (Line 19 – 23 in Algorithm 2) is executed only when a new measurement is available after \( T_s \) seconds. The update in uncertainty bound in the \( i \)th iteration also updates the admissible input set for the dynamic obstacles for the current and all future iterations. Considering the Remark 4 on time for convergence of the uncertainty bound, if the estimator reaches convergence after \( i = N_0 \) finite iterations in Algorithm 2, then the bound \( \hat{r}_k \) does not change ahead and the execution is identical to the usual pursuit evasion algorithm. Thus in such cases, the guarantees of probabilistic completeness and asymptotic optimality of the usual pursuit evasion algorithm [5] can be directly inherited by this algorithm too.

V. SIMULATION RESULTS

The robust adaptive motion planning method proposed in this paper was validated using a simulation experiment. We have simulated an agent which is planning to escape and reach a goal set, whereas the dynamic obstacle is patrolling at an outpost. The agent observes the motion and in parallel plans the optimal route to escape. The agent and dynamic obstacle are modeled as a single integrator with velocity bounds. The equation for agent and dynamic obstacle are:

\[
\begin{align*}
\dot{x}_A(t) &= \left[ \begin{array}{c} \dot{x}_{A,1}(t) \\
\dot{x}_{A,2}(t) \end{array} \right] = u_A(t) = \left[ \begin{array}{c} u_{A,1}(t) \\
 u_{A,2}(t) \end{array} \right], \\
\dot{x}_O(t) &= \left[ \begin{array}{c} \dot{x}_{O,1}(t) \\
\dot{x}_{O,2}(t) \end{array} \right] = w(t) = \left[ \begin{array}{c} w_1(t) \\
w_2(t) \end{array} \right],
\end{align*}
\]

where \( \|u_A(t)\|_2 \leq 2 \) and \( \|w(t)\|_2 \leq 1.3 \). The velocity bound for obstacle is not known to the agent and the conservative bound \( \tilde{r}_1 \) is considered to be 2. A 2D environment with 7 different static obstacles was considered. The initial location of the dynamic obstacle was fixed at \( (2, 16) \) and that of agent was at \( (10, 18) \). The goal set was defined to be a single point \( (4, 2) \). During the motion planning phase, the dynamic obstacle is converging to an elliptical path around an outpost with the following motion model:

\[
\begin{align*}
\dot{x}_{O,1}(t) &= -0.6(1 - e^{-t/2}) \sin(0.6t), \\
\dot{x}_{O,2}(t) &= 1.3(1 - e^{-t/3}) \cos(0.6t).
\end{align*}
\]

This input can be shown to satisfy the differentiability, bounded value, and bounded derivative criterion. Also, this input slowly reveals the capability of the dynamic obstacle. The bound \( r_1 = 1.300 \) and \( r_2 = 0.859 \). The sampling time of the sensor, i.e., \( T_s \), is set to be 0.01. In the state estimator (9) used for uncertainty estimation, the parameter \( a \) is chosen to be 0.05. The proposed technique is compared with the robust motion planning algorithm [5]. The conservative bound \( \tilde{r}_1 \), which is used to define the uncertainty set for this technique, is chosen to be 1.45 and 1.65 in the two different runs. Note that, these values are smaller than the one used in the proposed framework. The simulation results are presented ahead in this section.

The simulation results have been shown in Fig. 1. The figures 1a and 1b show the results of a typical execution of the proposed technique. The algorithm could find a path to reach the goal and the uncertainty bound estimate also converged very close above the true value. Considering the maximum error from the proposed uncertainty bound estimator from Theorem 1, the error \( \tilde{r}_1 - r_1 \) must be less than 0.0364, since \( \gamma(0.01) = 0.0182 \). The error from simulation was 0.0180, since the estimated value of \( r_1 \) was 1.3180. This error is less than the bound given in Theorem 1.

The distance traveled on the path to reach goal in the proposed technique was 17.23 (See Fig. 1a), which is lesser than 17.93 in the simulation run with robust planning using \( \tilde{r}_1 = 1.45 \) (See Fig. 1c). Another run with \( \tilde{r}_1 = 1.65 \) for robust planning could not find a path to the goal as the bound was too conservative, as shown in Fig. 1d. Additionally, it is noted that the trajectory obtained by using the correct uncertainty estimate bound (\( \tilde{r}_1 = 1.30 \)) with robust planning is identical to that obtained with uncertainty estimation (hence, it was not shown separately). It is noted that when the generated motion plan will be executed, the obstacle is assumed to be located at the initial state (shown as the red hexagram in Fig. 1). Thus, we show that uncertainty estimation can be used within the incremental sampling framework to aid in robust adaptive motion planning.

VI. CONCLUSIONS

This paper has presented an extension of the existing robust motion planning technique by building up an uncertainty estimation framework to enable robust adaptive planning. The developed framework uses the fast adaptive estimation techniques from adaptive control for estimating uncertainties in the models of other moving agents. These estimates of uncertainty were then used with an easy-to-implement update rule proposed for uncertainty bound estimation. The convergence analysis of this estimator and update rule was shown and the limit point of the sequence of estimates was characterized. It was shown that the limit point of this estimate sequence can be made arbitrarily close to the true uncertainty bound by sampling fast enough. The detailed algorithm for solving pursuit evasion games with uncertainty estimation was developed and presented in this paper. The algorithm was validated by simulation experiments and it was compared to an existing robust motion planning algorithm. The results and analysis demonstrate that systems which have access to sensing capabilities can leverage the benefit of adaptive planning and improve performance.

The robust adaptive planning research needs more exploration to enable on-line implementation with finite horizon planning, tolerance to noisy measurements, and partial state feedback. These will be the focus of the work in near future. Rigorous validation with experiments on ground robots will also be pursued to enhance the applicability of this work.
(b) Uncertainty bound estimator

(c) Robust planning: Run 1

(d) Robust planning: Run 2

Fig. 1: The results of proposed robust adaptive planning and existing robust planning technique is shown in an environment with multiple obstacles in three different runs. The trajectory is shown in blue, the obstacles are shown in gray, magenta circle denotes the goal, yellow pentagram denotes initial location of the agent, red hexagram denotes initial location of the obstacle, black line denotes the patrolling trajectory of the dynamic obstacle, and green tree branches denote all safe paths for the agent computed in 10000 iterations. Figures (a) and (b) are generated with proposed approach, showing the evader trajectory and uncertainty bound estimation result, respectively. Figures (c) and (d) are generated without using uncertainty estimation by considering the conservative bound for defining the uncertainty set.

REFERENCES


APPENDIX

Proof of Theorem 1: From (5), (4), using reverse triangular inequality, we get:
\[ ||w(t)|| - ||\hat{w}_k|| \leq ||w(t) - \hat{w}_k|| \leq \gamma(T_s) \]
for \( t \in [kT_s, (k+1)T_s) \) for some \( k \in \mathbb{N} \). By definition, \( r_1 = \max_{t \in [0, \infty)} ||w(t)|| \). Let \( t' \in [0, \infty) \) be a time instant when \( w(t') = r_1 \). There exists a \( k' \in \mathbb{N} \), such that, \( t' \) belongs to the interval \([k'T_s, (k'+1)T_s)\), then from (5), we get:
\[ r_1 \leq ||\hat{w}_k|| + \gamma(T_s), \]
\[ r_1 \leq \min(||\hat{w}_k|| + \gamma(T_s), \tilde{r}_1), \cdots, r_1 \leq \tilde{r}_1, \]
\[ r_1 \leq \max(\min(||\hat{w}_k|| + \gamma(T_s), \tilde{r}_1)) = \lim_{k \to \infty} \tilde{r}_1 = \hat{r}_1. \]

using monotonicity of the bound estimate sequence and definition of the limit point. Using boundedness of this sequence, we get the desired result: \( r_1 \leq \hat{r}_1 \leq \tilde{r}_1 \).

Now we prove 0 \( \leq \tilde{r}_1 \leq r_1 \leq 2\gamma(T_s) \). Using (6), we get:
\[ ||\hat{w}_k|| \leq ||w(t)|| + \gamma(T_s), \]
where \( k^* \) is the index for which the uncertainty estimate is maximum and \( t^* \) is some time instant in \([k^*T_s, (k^*+1)T_s)\).

Using definition of \( r_1 \), we get:
\[ ||\hat{w}_k|| \leq r_1 + \gamma(T_s), \]
\[ ||\hat{w}_k|| + \gamma(T_s) \leq r_1 + 2\gamma(T_s), \]
\[ \min(||\hat{w}_k|| + \gamma(T_s), \tilde{r}_1) \leq r_1 + 2\gamma(T_s), \]
\[ \hat{r}_1 \leq r_1 + 2\gamma(T_s), \]
using monotonicity of the estimate sequence. Since, \( k^* \) is the index corresponding to maximum uncertainty estimate over the complete sequence, \( \hat{r}_1 \) is same as \( \hat{r}_1 \). Combining this result with relation 1, we get:
\[ r_1 \leq \hat{r}_1 \leq \tilde{r}_1 \leq r_1 + 2\gamma(T_s). \]

\[ \blacksquare \]