Generalization of $\nu^\star$ Path Planning For
Accommodation of Amortized Dynamic Uncertainties in Plan Execution

Ishanu Chattopadhyay\textsuperscript{‡} Asok Ray\textsuperscript{‡}
ixc128@psu.edu axr2@psu.edu
The Pennsylvania State University
University Park, PA 16802, USA

Abstract—A significant generalization to the language-measure-theoretic path planning algorithm $\nu^\star$ is presented that accounts for average dynamic uncertainties in plan execution. The planning problem thus can be solved with parametric input from the dynamics of the robotic platform under consideration. Applicability of the algorithm is demonstrated in a simulated maze solution and by experimental validation on a mobile robotic platform in the laboratory environment.

Index Terms—Language Measure; Probabilistic Finite State Machines; Robotics; Path Planning; Supervisory Control

1. INTRODUCTION & MOTIVATION

Recently, a novel path planning algorithm $\nu^\star$ was reported that models the navigation problem in the framework of Probabilistic Finite State Machines and computes robust optimal plans via optimization of the PFSA from a strictly control-theoretic viewpoint. In this paper, we present a significant improvement; the average dynamic uncertainty in plan execution is integrated with the planning process, resulting in plans that are highly robust and take into account the average effect of physical dynamic limitations of individual robotic platforms and possibly different operating conditions and execution parameters. Thus we address the fact that robots have physical limitations on what commands can be executed and with what precision; the planning must take this into account to yield robust execution. It is important to note that we still consider a static known map in this paper; however the proposed approach allows for the possibility that planned command sequence may not executed perfectly. The key advantages are:

1) Pre-processing is cheap: The cellular decomposition required by $\nu^\star$ is simple and computationally cheap. The cells are mapped to PFSA states which are defined to have identical connectivity via symbolic inter-state transitions.

2) Fundamentally different from search: $\nu^\star$ optimizes the resultant PFSA via a iterative sequence of combinatorial operations which elementwise maximizes the language measure vector [1][2].

3) Computational efficiency: The time complexity of each iteration step can be shown to be linear in problem size implying significant numerical advantage over search-based methods for high-dimensional problems.

4) Global monotonicity: The solution iterations are globally monotonic. The final waypoint sequence is generated essentially by following the measure gradient which is maximized at the goal. The measure gradient is reminiscent of potential field methods [3]. However, $\nu^\star$ automatically generates the measure gradient; no potential function is necessary. Furthermore, the potential function

\begin{itemize}
    \item based planners often get trapped in local minimum which can be shown to be a mathematical impossibility for $\nu^\star$.
\end{itemize}

The paper is organized in five sections including the present one. Section 2 briefly explains the language-theoretic models considered in this paper, reviews the language-measure-theoretic optimal control of probabilistic finite state machines and presents the necessary details of the reported $\nu^\star$ algorithm. Section 3 presents the modifications to the navigation model to incorporate the effects of dynamic uncertainties within the framework of probabilistic automata. Pertinent theoretical results are presented that capture the key characteristics of the approach. The modified approach is validated in experiment on a SEGWAY RMP 200 two-wheeled robot. Section 4 derives a recursive formulation of $\nu^\star$ under dynamic uncertainty that is shown to be critically important for elimination of local maxima. A detailed simulated maze solution is presented as an example. The paper is summarized and concluded in Section 5 with recommendations for future work.

2. LANGUAGE-MEASURE-THEORETIC OPTIMIZATION

This section summarizes the signed real measure of regular languages; the details are reported in [4]. Let $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ be a trim (i.e., accessible and co-accessible) finite-state automaton model that represents the discrete-event dynamics of a physical plant, where $Q = \{q_k : k \in I_q\}$ is the set of states and $I_q = \{1,2,\ldots,n\}$ is the index set of states; the automaton starts with the initial state $q_i$; the alphabet of events is $\Sigma = \{\sigma_k : k \in I_{\Sigma}\}$, having $\Sigma \cap I_q = \emptyset$ and $I_{\Sigma} = \{1,2,\ldots,\ell\}$ is the index set of events; $\delta : Q \times \Sigma \rightarrow Q$ is the (possibly partial) function of state transitions; and $Q_m = \{q_{m_1}, q_{m_2}, \ldots, q_{m_l}\} \subseteq Q$ is the set of marked (i.e., accepted) states with $q_{m_k} = q_j$ for some $j \in I_q$. Let $\Sigma^*$ be the Kleene closure of $\Sigma$, i.e., the set of all finite-length strings made of the events belonging to $\Sigma$ as well as the empty string $\epsilon$ that is viewed as the identity of the monoid $\Sigma^*$ under the operation of string concatenation, i.e., $\epsilon s = s = s \epsilon$. The state transition map $\delta$ is recursively extended to its reflexive and transitive closure $\delta : Q \times \Sigma^* \rightarrow Q$ by defining $\forall q_j \in Q, \delta(q_j, \epsilon) = q_j$ and $\forall q_j \in Q, q_r \in \Sigma^*, \delta(q_j, qr) = \delta(\delta(q_j, q_r), r)$.

Definition 2.1: The language $L(q_j)$ generated by a DFSA $G$ initialized at the state $q_j \in Q$ is defined as: $L(q_j) = \{ s \in \Sigma^* | \delta^*(q_j, s) \in Q\}$

The language $L_m(q_j)$ marked by the DFSA $G$ initialized at the state $q_j \in Q$ is defined as: $L_m(q_j) = \{ s \in \Sigma^* | \delta^*(q_j, s) \in Q_m\}$

Definition 2.2: For every $q_j \in Q$, let $L(q_j)$ denote the set of all strings that, starting from the state $q_j$, terminate at the state $q_j$, i.e.,

$L_{t,j} = \{ s \in \Sigma^* | \delta^*(q_j, s) = q_j \in Q\}$

The formal language measure is first defined for terminating plants [5] with sub-stochastic event generation probabilities i.e. the event generation probabilities at each state summing to strictly less than unity.

Definition 2.3: The event generation probabilities are specified by the function $P : \Sigma^* \times Q \rightarrow [0,1]$ such that $\forall q_j \in Q, \forall s \in \Sigma, \forall s \in \Sigma^*$,

$(1) \tilde{P}(\sigma_k, q_j) = \tilde{P}^{\star}_{jk} \in [0,1], \quad \sum_k \tilde{P}^{\star}_{jk} = 1 - \theta, \text{ with } \theta \in (0,1)$
Def. 2.4: The state transition probability \( \pi : Q \times Q \to [0,1] \) of the DFSA \( G_i \) is defined as follows: \( \forall q_i, q_j \in Q, \pi(q_j, q_i) = \sum_{\sigma \in \Sigma} \pi(q_i, \sigma) \delta(q_j, \sigma, q_i) \). The \( n \times n \) state transition probability matrix is defined as: \( \Pi_{q_j} = \pi(q_i, q_j) \).

The set \( Q_m \) of marked states is partitioned into \( Q^+ \) and \( Q^- \), i.e., \( Q_m = Q^+ \cup Q^- \) and \( Q^+ \cap Q^- = \emptyset \), where \( Q^+ \) contains all good marked states that we desire to reach, and \( Q^- \) contains all bad marked states that we want to avoid, although it may not always be possible to completely avoid the bad states while attempting to reach the good states. To characterize this, each marked state is assigned a real value based on the designer’s perception of its impact on the system performance.

Def. 2.5: The characteristic function \( \chi : Q \to [-1,1] \) that assigns a signed real weight to state-based sublanguages \( L(q_i, q_j) \) is defined as:

\[
L(q_i, q_j) = \begin{cases} 
-1, & q \in Q^+ \\
0, & q \notin Q_m \\
1, & q \in Q^- 
\end{cases}
\]

(1)

The state weighting vector, denoted by \( \chi = [\chi_1, \chi_2, \ldots, \chi_I]^T \), where \( \chi_j = \chi(q_j) \forall j \in I_q \), is called the \( \chi \)-vector. The \( j \)-th element \( \chi_j \) of the \( \chi \)-vector is the weight assigned to the corresponding state \( q_j \).

In general, the marked language \( L_m(q_i) \) consists of both good and bad event strings that, starting from the initial state \( q_i \), lead to \( Q^+ \) and \( Q^- \), respectively. Any event string belonging to the language \( L_0 = L(q_i) - L_m(q_i) \) leads to one of the non-marked states belonging to \( Q^+ \) and \( Q^- \) and does not contain any of the good or bad strings. Based on the equivalence classes defined in the Myhill-Nerode Theorem, the regular languages \( L(q_i) \) and \( L_m(q_i) \) can be expressed as:

\[ L(q_i) = \bigcup_{q \in Q^+} L_q \cup L(q_i) = \bigcup_{q \in Q^-} L_q \cup L(q_i) = L_0 \cup L_m \]

where the sublanguage \( L_{i,k} \subseteq G_i \) has the initial state \( q_i \) uniquely labelled by the terminal state \( q_i, k \in I_q \) and \( L_{i,j} \cap L_{i,k} = \emptyset \forall j \neq k \); and \( L_{i,m} \equiv \bigcup_{q \in Q^-} L_q \cup L(q_i) \) are good and bad sublanguages of \( L_{i,k} \), respectively. Then, \( L_0 = \bigcup_{q \in Q^+} L_q \cup L(q_i) \) and \( L(q_i) = L_0 \cup L_{i,m} \cup L_{i,m} \).

A signed real measure \( \mu : 2^{L(q_i)} \to \mathbb{R} \) is defined as:

\[ \mu \equiv \chi(q_i) \chi(q_j) \]

(2)
The solution to the optimal supervision problem is obtained in [2], [7] by designing an optimal policy for a terminating plant [5] with a sub-stochastic transition probability matrix \((1 - \theta)\Pi\) with \(\theta \in (0, 1)\). To ensure that the computed optimal policy coincides with the one for \(\theta = 0\), the suggested algorithm chooses a small value for \(\theta\) in each iteration step of the design algorithm. However, choosing \(\theta\) too small may cause numerical problems in convergence. 

**Remark 2.1:** The underlying theory does not require the grid to be regular; the numbering scheme chosen is irrelevant. In the absence of dynamic constraints and state estimation uncertainties, the alphabet contains one uncontrollable event \(i.e., \Sigma = \Sigma_C \cup \{u\}\) such that \(\Sigma_C\) is the set of uncontrollable events corresponding to the possible moves of the robot. The uncontrollable event \(u\) is defined from each of the blocked states and leads to \(q_0\) which is a deadlock state. All other transitions (i.e., moves) are removed from the blocked states. Thus, if a robot moves into a blocked state, it uncontrollably transitions to the deadlock state \(q_0\) which is physically interpreted to be a collision. We further assume that the robot fails to recover from collisions which is reflected by making \(q_0\) a deadlock state. We note that \(q_0\) does not correspond to any physical grid location. The set of blocked grid locations along with the obstacle state \(q_0\) is denoted as \(Q_{OBSTACLE} \subseteq Q\). Figure 1 illustrates the navigation automaton for a nine state discretized workspace with two blocked squares. Note that the only outgoing transition from the blocked states \(q_1\) and \(q_3\) is \(u\). Next we augment the navigation FSA by specifying event generation probabilities defined by the map \(\tilde{\pi} : Q \times \Sigma \rightarrow [0, 1]\) and the characteristic state-weight vector specified as \(\chi : Q \rightarrow [-1, 1]\). The characteristic state-weight vector \([2]\) assigns scalar weights to the PFSA states to capture the desirability of ending up in each state.

**Definition 2.13:** The characteristic weights are specified for the navigation automaton as follows:

\[
\chi(q_i) = \begin{cases} 
-1 & \text{if } q_i = q_0 \\
1 & \text{if } q_i \text{ is the goal} \\
0 & \text{otherwise}
\end{cases}
\]

(3)

In the absence of dynamic constraints and state estimation uncertainties, the robot can "choose" the particular controllable transition to execute at any grid location. Hence we assume that the probability of generation of controllable events is uniform over the set of moves defined at any particular state.

**Definition 2.14:** Since there is no uncontrollable events defined at any of the unblocked states and no controllable events defined at any of the blocked states, we have the following consistent specification of event generation probabilities: 

\[
\tilde{\pi}(q_i, \sigma_j) = \begin{cases} 
\text{No. of controllable events at } q_i & \text{if } \sigma_j \in \Sigma_C \\
1 & \text{otherwise}
\end{cases}
\]

The boundaries are handled by "surrounding" the workspace with blocked position states shown as "boundary obstacles" in the upper part of Figure 1(c).
Definition 2.15: The navigation model id defined to have identical connectivity as far as controllable transitions are concerned implying that every controllable transition or move (i.e. every element of $\Sigma_C$) is defined from each of the unblocked states.

D. Problem Solution as a Decision-theoretic Optimization of PFSA

The above-described probabilistic finite state automaton (PFSA) based navigation model allows us to compute optimally feasible path plans via the language-measure-theoretic optimization algorithm [2] described in Section 2. Keeping in line with no nomenclature in the path-planning literature, we refer to the language-measure-theoretic algorithm as $v^*$ in the sequel. For the unsupervised model, the robot is free to execute any one of the defined controllable events from any given grid location (See Figure 1(b)). The optimization algorithm selectively disables controllable transitions to ensure that the formal measure vector of the navigation automaton is element-wise maximized. Physically, this implies that the supervised robot is constrained to choose among only the enabled moves at each state such that the probability of collision is minimized with the probability of reaching the goal simultaneously maximized. Although $v^*$ is based on optimization of probabilistic finite state machines, it is shown that an optimal and feasible path plan can be obtained that is executable in a purely deterministic sense. Mobile robotic platforms however suffer from varying degrees of dynamic and parametric uncertainties, implying that path length minimization is of lesser practical importance to computing plans that are robust under sensor noise, imperfect actuation and possibly accumulating odometry errors. Even with sophisticated signal processing techniques such errors cannot be eliminated. The $v^*$ algorithm addresses this issue by an optimal trade-off between path lengths and availability of feasible alternate routes in the event of unforeseen dynamic uncertainties. If $\omega$ is the shortest path to goal from state $q_i$, then the shortest path from state $q_j$ (with $q_i \xrightarrow{c_i} q_k$) is given by $\sigma_i \omega$. However, a larger number of feasible paths may be available from state $q_j$ (with $q_i \xrightarrow{c_j} q_k$) which may result in the optimal $v^*$ plan to be $\sigma_j \omega_i$. Mathematically, each feasible path from state $q_i$ has a positive measure which may sum to be greater than the measure of the single path $\omega$ from state $q_i$. The condition $v_i(q_j) > v_i(q_k)$ would then imply that the next state from $q_i$ would be computed to be $q_j$ and not $q_k$. Physically it can be interpreted that the mobile gent is better off going to $q_j$ since the goal remains reachable even if one or more paths become unavailable. The key results [8] are as follows:

Lemma 3.1: For the optimally supervised navigation automaton $G^*_\text{Nav}$, we have $\forall q_i \in Q \setminus Q_\text{OBSTACLE}$,

$$v_i((\omega_i)) = \min_{\sigma_i \in \text{CARD}(\Sigma_C)} (1 - \theta_{\text{min}}(G_{\text{Nav}}(q_i)))$$

Proposition 3.1: For $q_i \in Q \setminus Q_\text{OBSTACLE}$, let $q_i \xrightarrow{c_i} q_j \rightarrow \cdots \rightarrow q_{G_{\text{GOAL}}}$ be the shortest path to the goal. If there exists $q_k \in Q \setminus Q_\text{OBSTACLE}$ with $q_i \xrightarrow{c_i} q_k$ for some $q_k \in \Sigma_C$ such that $v_i(q_k) > v_i(q_j)$, then the number of distinct paths to goal from state $q_k$ is at least $\text{CARD}(\Sigma_C) + 1$. The lower bound computed in Proposition 3.1 is not tight and if the alternate paths are longer or if there are multiple ‘shortest’ paths then the number of alternate routes required is significantly higher. Detailed examples can be easily presented to illustrate situation where $v^*$ opts for a longer but more robust plan.

B. Robustness to Dynamic Uncertainty

In this paper, we modify the PFSA-based navigation model to explicitly reflect dynamic uncertainties in plan execution.

3. Tradeoff between Computed Path Length & Plan Robustness

A. Robustness to Map Uncertainty

Majority of reported path planning algorithms consider minimization of the computed feasible path length as the sole optimization objective. Mobile robotic platforms however suffer from varying degrees of dynamic and parametric uncertainties, implying that path length minimization is of lesser practical importance to computing plans that are robust under sensor noise, imperfect actuation and possibly accumulating odometry errors. Even with sophisticated signal processing techniques such errors cannot be eliminated. The $v^*$ algorithm provides a trade-off between path lengths and availability of feasible alternate routes in the event of unforeseen dynamic uncertainties. If $\omega$ is the shortest path to goal from state $q_i$, then the shortest path from state $q_j$ (with $q_i \xrightarrow{c_i} q_k$) is given by $\sigma_i \omega$. However, a larger number of feasible paths may be available from state $q_j$ (with $q_i \xrightarrow{c_j} q_k$) which may result in the optimal $v^*$ plan to be $\sigma_j \omega_i$. Mathematically, each feasible path from state $q_i$ has a positive measure which may sum to be greater than the measure of the single path $\omega$ from state $q_i$. The condition $v_i(q_j) > v_i(q_k)$ would then imply that the next state from $q_i$ would be computed to be $q_j$ and not $q_k$. Physically it can be interpreted that the mobile gent is better off going to $q_j$ since the goal remains reachable even if one or more paths become unavailable. The key results [8] are as follows:

Lemma 3.1: For the optimally supervised navigation automaton $G^*_\text{Nav}$, we have $\forall q_i \in Q \setminus Q_\text{OBSTACLE}$,

$$v_i((\omega_i)) = \min_{\sigma_i \in \text{CARD}(\Sigma_C)} (1 - \theta_{\text{min}}(G_{\text{Nav}}(q_i)))$$

Proposition 3.1: For $q_i \in Q \setminus Q_\text{OBSTACLE}$, let $q_i \xrightarrow{c_i} q_j \rightarrow \cdots \rightarrow q_{G_{\text{GOAL}}}$ be the shortest path to the goal. If there exists $q_k \in Q \setminus Q_\text{OBSTACLE}$ with $q_i \xrightarrow{c_i} q_k$ for some $q_k \in \Sigma_C$ such that $v_i(q_k) > v_i(q_j)$, then the number of distinct paths to goal from state $q_k$ is at least $\text{CARD}(\Sigma_C) + 1$. The lower bound computed in Proposition 3.1 is not tight and if the alternate paths are longer or if there are multiple ‘shortest’ paths then the number of alternate routes required is significantly higher. Detailed examples can be easily presented to illustrate situation where $v^*$ opts for a longer but more robust plan.

B. Robustness to Dynamic Uncertainty

In this paper, we modify the PFSA-based navigation model to explicitly reflect dynamic uncertainties in plan execution.
**Definition 3.1:** The modified navigation automaton \( = (Q, \Sigma, \delta, \tilde{\Pi}, \chi, ) \) is defined similar to the formulation in Section 2-C with the additional parameter \( \in [0, 1] \) quantifying the expected dynamic uncertainty in terms of transition uncontrollability as follows: Each event \( \sigma \in \Sigma \) is assumed to be decomposable as

\[
\sigma = (\sigma_{\text{controllable}}, \sigma_{\text{uncontrollable}})
\]

with the transition probabilities distributing as

\[
\forall q_i \in Q, \quad \tilde{\Pi}(q_i, \sigma_{\text{controllable}}) = \tilde{\Pi}(q_i, \sigma)
\]

\[
\forall q_i \in Q, \quad \tilde{\Pi}(q_i, \sigma_{\text{uncontrollable}}) = (1-\tilde{\Pi}(q_i, \sigma)
\]

The effect of dynamic uncertainty is illustrated in Figure 2. Note, while in absence of uncertainty, one can disable transitions perfectly, in the modified model, such disabling is only partial. The model incorporates physical movement errors and sensing noise in an amortized fashion. For example, it can be expected, that the probability of erroneously moving right when the robot is asked to go left, would be smaller than, maybe, moving forward. Thus, in reality, the factor should be different for each event \( \sigma \in \Sigma \); and would also vary with the current continuous dynamic state of the overall system. However, one can estimate a constant factor for the specific robotic platform under consideration, by averaging over the observed errors from sufficiently long experimental runs. As a specific example, the uncertainty parameter for a two-wheeled robot such as the SEGWAY RMP 200 will be significantly higher compared to a more stable four wheeled SEGWAY RMP 400.

**Remark 3.1:** The state transition matrix \( \Pi \) for decomposes as

\[
\Pi = \Pi + (1-\Pi), \quad \Pi \text{ corresponds to controllable transitions and the residual to the uncontrollable transitions arising from dynamic uncertainties.}
\]

For small models, the modified model can be optimized via the measure-theoretic technique in a straightforward manner, using the \( v^* \)-algorithm reported in [8]. However, due to the presence of uncontrollable transitions, some of the results obtained in [8] need to be modified. Furthermore, large problem sizes give rise to critical issues due to partial controllability of transitions in presence of dynamic uncertainty, which would be addressed in the next section.

**Proposition 3.2:** (Weaker Version of Proposition 2.2) There exists a \( v^* \)-path \( \nu(q_i, q_{\text{Goal}}) \) from any state \( q_i \in Q \) to the goal \( q_{\text{Goal}} \in Q \) if \( \nu(q_i) > 0 \).

**Proof:** We note that \( \nu(q_i) > 0 \) implies that there necessarily exists at least one string \( \omega \) of positive measure initiating from \( q_i \); and hence there exists at least one string that terminates on \( q_{\text{Goal}} \). The proof then follows from the definition of \( v^* \)-paths (See Definition 2.16).

**Proposition 3.3:** Let \( [1,2] \) be two navigation automata differing only in the value of the uncertainty parameter, with \( 1 > 2 \). If for some \( q_i \in Q, \nu(q_i) > 0 \) and \( v^2(q_i) > 0 \), then the shortest \( v^* \)-path from \( q_i \) to \( q_{\text{Goal}} \in 2 \) is at least as long as the corresponding shortest \( v^* \)-path from \( q_i \) to \( q_{\text{Goal}} \in 1 \).

**Proof:** We use induction on the length of the shortest \( v^* \)-path from \( q_i \) to \( q_{\text{Goal}} \) in \( 2 \), which we denote as \( \ell_1 \). First we note that the result is trivially true if \( \ell_1 = 0 \) or \( \ell_1 = 1 \). As our induction hypothesis, we assume that the result is true for \( \ell_1 = k \). Then, for \( \ell_1 = k + 1 \), we note that if \( q_i, q_{r_1}, \ldots, q_{r_k} \) is the shortest \( v^* \)-path in \( 2 \), then the shortest \( v^* \)-path from \( q_i \) to \( q_{r(k-1)} \) cannot be longer than \( k \) (as per our induction hypothesis). The proof is then completed by noting that \( q_{r_k} \) is actually \( q_{\text{Goal}} \), and hence the path from \( q_{r(k-1)} \) to \( q_{r_k} \) is a single hop in \( 1 \).

**Remark 3.2:** Proposition 3.3 implies that higher dynamic uncertainty leads to longer \( v^* \)-paths in general.

Unfortunately, the critical result pertaining to absence of local maxima (Corollary 2.2) is no longer valid and we will discuss how to remedy this in the sequel. However, we have the following result:

**Proposition 3.4:** The solution of the modified planning problem solves the following optimization problem: Maximize \( p_1 - p_2 \) under the model constraints, where \( p_1 \) and \( p_2 \) are the stationary probabilities of reaching the goal and hitting an obstacle respectively.

**Proof:** We recall that the language-measure-theoretic optimization of PFSA accomplishes the maximization of \( \omega^T H \) where \( \omega \) is the stationary probability vector on the automaton states [2]. Since \( \chi(q_{\text{Goal}}) \approx 1 \) and \( \chi(q_{\text{Obstacle}}) \approx -1 \) and all other states have zero characteristic, it follows that \( p_1 - p_2 \) gets maximized in the optimization.

**Remark 3.3:** We note that under the modified model, \( \nu(q_i) < 0 \) needs to be interpreted somewhat differently. In absence of any dynamic uncertainty, \( \nu(q_i) < 0 \) implies that no path to goal exists. However, due to weakening of Proposition 2.1 (See Proposition 3.2), and in the light of Proposition 3.4, \( \nu(q_i) < 0 \) implies that the probability of reaching goal is smaller to that of hitting an obstacle from the state \( q_i \).

C. Experimental Validation with SEGWAY RMP

The proposed modification is validated on a SEGWAY RMP 200 which is a two-wheeled robot with significant dynamic uncertainty. In particular, the inverted-pendulum dynamics prevents the platform from halting instantaneously. The experimental runs were conducted at the Networked Robotics & Systems Laboratory (NRSL), Pennstate, with the workspace discretized into a \( 53 \times 29 \) grid. Each grid location is about 4 sq. ft. allowing the SEGWAY to fit complete inside each such discretized positional state which justifies the simplified circular robot modeling. The runs are illustrated in Figure 4. The robots were run at three different average speeds; leading to three different values of the uncertainty parameter \( \nu \). In the top plate, \( \nu = 0.98 \) with average robot speed \( v = 0.3m/sec \). The middle plate illustrates the case with \( \nu = 0.9,v = 0.5m/sec \) and for the bottom plate the values are \( \nu = 0.85,v = 1m/sec \). The plates on the left hand side illustrate the measure gradients; the ones on the right illustrate the executed plan. The results show that the approach presented in this paper successfully integrates amortized dynamics with autonomous planning.

4. Recursive Decomposition for Maxima Elimination

Weakening of Proposition 2.1 (See Proposition 3.2) has the crucial consequence that Corollary 2.2 is no longer valid. Local maximac occur under the modified model. This is a serious problem for autonomous planning and must be remedied. Local Maxima elimination is notoriously difficult for potential based planning approaches. The problem becomes critically important when applied to solution of mazes; larger the number of obstacles, higher is the chance of ending up in a local maxima. However, \( v^* \) can be modified with ease into a recursive scheme that eliminates local maxima occurring in models with non-zero dynamic uncertainty. The correctness of the proposed is established in the next proposition.

**Proposition 4.1:** 1. The planning loop terminates in finite number of steps.

2. The execution loop is free from local maxima.

**Proof:** Statement 1 immediately follows from the finiteness of the state set \( Q \) and the fact that \( H_k, H_j \) are mutually disjoint for \( k \neq j \).
For Statement 2, we argue by the method of induction. First, we note that if the initial state $q_i$ is in $H_1$, then $\nu_k(q_i) > 0$ w.r.t. the plan saved in $M_1$, implying that there is a $\nu^*$-path to the goal. For our induction hypothesis, we assume the result is true if $q_i$ is in $H_k$. Next, let $q_i \in H_{k+1}$. Let $M_k = \nu_k$. Then, since $M_{k+1}$ was obtained by solving the planning problem after setting every state in $H_k$ as goal, we conclude that there exists a $\nu^*$-path to some state $q_j \in H_k$, which in turn implies the existence of a succession of $\nu$-star-paths to the goal by our induction hypothesis. This completes the proof. Figure 3(a) illustrates the sequential execution.

**Remark 4.1:** The recursive version of the $\nu^*$ can be interpreted as accomplishing the following: Simultaneously minimize the probability of hitting any obstacle and maximize the probability of reaching the goal, under the constraint that the robot executes the planned local moves only with $\times 100\%$ probability at any instant.

A. Simulation Example

Recursive $\nu^*$ is validated with a detailed simulation example as illustrated in Figures 3(b). The crucial problem that the recursive procedure addresses is clear from plate (a). Note that the number of states with positive measure is very small; implying that from the remaining states, the robot is more probable to hit a obstacle than reach the goal. The final plan is constructed by piecing together the plans obtained within $H_1$ to $H_6$. The result is shown in Figure 3(b). The dotted lines are the plans computed under dynamic uncertainty; the solid lines are plans that assume perfect execution. Note the plans that assume uncertainty are significantly longer; but go around narrow spaces, whereas, the solid lines goes through them. The color coding on the dotted lines illustrate the different planning zones $H_1$ to $H_6$.

5. **SUMMARY & FUTURE RESEARCH**

A novel path planning algorithm $\nu^*$ is introduced that models the autonomous navigation as an optimization problem for probabilistic finite state machines and applies the rigorous theory of language-measure-theoretic optimal control to compute $\nu$-optimal plan to the specified goal, with automated trade-off between path length and robustness of the plan under dynamic uncertainty. Future work will extend the language-measure theoretic planning algorithm to address the following problems:

1) **Multi-robot coordinated planning:** Future work will address multi-robot scenarios, with each robot treating the remaining group as moving obstacles.

2) **Hierarchical implementation to handle very large workspaces:** Large workspaces can be solved more efficiently if planning is done when needed rather than solving the whole problem at once.

3) **Handling partially observable dynamic events:** Physical errors and onboard sensor failures may need to be handled as unobservable transitions and will be addressed in future publications.

**REFERENCES**


