2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

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

Ishanu Chattopadhyay[‡] Asok Ray[‡]
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 based planners often get trapped in local minimum which can be shown to be a mathematical impossibility for $\nu^\star$.

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 $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$ be a trim (i.e., accessible and co-accessible) finite-state automaton model that represents the discrete-event dynamics of a physical plant, where $Q = \{q_k : k \in \mathcal{I}_Q\}$ is the set of states and $\mathcal{I}_Q \equiv \{1, 2, \cdots, n\}$ is the index set of states; the automaton starts with the initial state q_i; the alphabet of events is $\Sigma = \{\sigma_k : k \in \mathcal{I}_\Sigma\}$, having $\Sigma \bigcap \mathcal{I}_Q = \emptyset$ and $\mathcal{I}_\Sigma \equiv \{1, 2, \cdots, \ell\}$ is the index set of events; $\delta : Q \times \Sigma \to Q$ is the (possibly partial) function of state transitions; and $Q_m \equiv \{q_{m_1}, q_{m_2}, \cdots, q_{m_l}\} \subseteq Q$ is the set of marked (i.e., accepted) states with $q_{m_k} = q_j$ for some $j \in \mathcal{I}_Q$. Let Σ^* be the Kleene closure of Σ, i.e., the set of all finite-length strings made of the events belonging to Σ as well as the empty string ϵ that is viewed as the identity of the monoid Σ^* under the operation of string concatenation, i.e., $\epsilon s = s = s\epsilon$. The state transition map δ is recursively extended to its reflexive and transitive closure $\delta : Q \times \Sigma^* \to Q$ by defining $\forall q_j \in Q,\ \delta(q_j, \epsilon) = q_j$ and $\forall q_j \in Q, \sigma \in \Sigma, s \in \Sigma^\star,\ \delta(q_i, \sigma s) = \delta(\delta(q_i, \sigma), s)$.

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

Definition 2.2: *For every $q_j \in Q$, let $L(q_i, q_j)$ denote the set of all strings that, starting from the state q_i, terminate at the state q_j, i.e., $L_{i,j} = \{s \in \Sigma^* \mid \delta^*(q_i, s) = q_j \in Q\}$*

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 $\tilde{\pi} : \Sigma^\star \times Q \to [0, 1]$ such that $\forall q_j \in Q, \forall \sigma_k \in \Sigma, \forall s \in \Sigma^\star$,*
(1) $\tilde{\pi}(\sigma_k, q_j) \triangleq \tilde{\pi}_{jk} \in [0, 1)$; $\sum_k \tilde{\pi}_{jk} = 1 - \theta$, with $\theta \in (0, 1)$;

★This work has been supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant No. W911NF-07-1-0376 and by the Office of Naval Research under Grant No. N00014-08-1-380. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

‡The Pennsylvania State University, University Park, PA 16802, USA.

(2) $\tilde{\pi}(\sigma, q_j) = 0$ *if* $\delta(q_j, \sigma)$ *is undefined;* $\tilde{\pi}(\epsilon, q_j) = 1$*;*
(3) $\tilde{\pi}(\sigma_k s, q_j) = \tilde{\pi}(\sigma_k, q_j)\ \tilde{\pi}(s, \delta(q_j, \sigma_k))$.
The $n \times \ell$ *event cost matrix is defined as:* $\widetilde{\mathbf{\Pi}}|_{ij} = \tilde{\pi}(q_i, \sigma_j)$

Definition 2.4: *The state transition probability* $\pi : Q \times Q \to [0, 1)$*, of the DFSA* G_i *is defined as follows:* $\forall q_i, q_j \in Q, \pi_{ij} = \sum_{\sigma \in \Sigma \text{ s.t. } \delta(q_i,\sigma)=q_j} \tilde{\pi}(\sigma, q_i)$ *The* $n \times n$ *state transition probability matrix is defined as* $\mathbf{\Pi}|_{jk} = \pi(q_i, q_j)$.

The set Q_m of marked states is partitioned into Q_m^+ and Q_m^-, i.e., $Q_m = Q_m^+ \cup Q_m^-$ and $Q_m^+ \cap Q_m^- = \emptyset$, where Q_m^+ contains all *good* marked states that we desire to reach, and Q_m^- 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.

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

$$\forall q \in Q, \quad \chi(q) \in \begin{cases} [-1, 0), & q \in Q_m^- \\ \{0\}, & q \notin Q_m \\ (0, 1], & q \in Q_m^+ \end{cases} \tag{1}$$

The state weighting vector, denoted by $\boldsymbol{\chi} = [\chi_1\ \chi_2\ \cdots\ \chi_n]^T$*, where* $\chi_j \equiv \chi(q_j)\ \forall j \in \mathcal{I}_Q$*, is called the* $\boldsymbol{\chi}$*-vector. The* j*-th element* χ_j *of* $\boldsymbol{\chi}$*-vector is the weight assigned to the corresponding terminal 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_m^+ and Q_m^- 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 $Q - Q_m$ and L^0 does not contain any one 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: $L(q_i) = \bigcup_{q_k \in Q} L_{i,k}$ and $L_m(q_i) = \bigcup_{q_k \in Q_m} L_{i,k} = L_m^+ \cup L_m^-$ where the sublanguage $L_{i,k} \subseteq G_i$ having the initial state q_i is uniquely labelled by the terminal state $q_k, k \in \mathcal{I}_Q$ and $L_{i,j} \cap L_{i,k} = \emptyset\ \forall j \neq k$; and $L_m^+ \equiv \bigcup_{q_k \in Q_m^+} L_{i,k}$ and $L_m^- \equiv \bigcup_{q_k \in Q_m^-} L_{i,k}$ are good and bad sublanguages of $L_m(q_i)$, respectively. Then, $L^0 = \bigcup_{q_k \notin Q_m} L_{i,k}$ and $L(q_i) = L^0 \cup L_m^+ \cup L_m^-$.

A signed real measure $\mu^i : 2^{L(q_i)} \to \mathbb{R} \equiv (-\infty, +\infty)$ is constructed on the σ-algebra $2^{L(q_i)}$ for any $i \in \mathcal{I}_Q$; interested readers are referred to [4] for the details of measure-theoretic definitions and results. With the choice of this σ-algebra, every singleton set made of an event string $s \in L(q_i)$ is a measurable set. By Hahn Decomposition Theorem [6], each of these measurable sets qualifies itself to have a numerical value based on the above state-based decomposition of $L(q_i)$ into L^0(null), L^+(positive), and L^-(negative) sublanguages.

Definition 2.6: *Let* $\omega \in L(q_i, q_j) \subseteq 2^{L(q_i)}$*. The signed real measure* μ^i *of every singleton string set* $\{\omega\}$ *is defined as:* $\mu^i(\{\omega\}) \equiv \tilde{\pi}(\omega, q_i)\chi(q_j)$*. The signed real measure of a sublanguage* $L_{i,j} \subseteq L(q_i)$ *is defined as:* $\mu_{i,j} \equiv \mu^i(L(q_i, q_j)) = \left(\sum_{\omega \in L(q_i,q_j)} \tilde{\pi}[\omega, q_i]\right)\chi_j$

Therefore, the signed real measure of the language of a DFSA G_i initialized at $q_i \in Q$, is defined as $\mu_i \equiv \mu^i(L(q_i)) = \sum_{j \in \mathcal{I}_Q} \mu^i(L_{i,j})$. It is shown in [4] that the language measure can be expressed as $\mu_i = \sum_{j \in \mathcal{I}_Q} \pi_{ij}\mu_j + \chi_i$. The language measure vector, denoted as $\mu = [\mu_1\ \mu_2\ \cdots\ \mu_n]^T$, is called the μ-vector. In vector form, we have $\mu = \mathbf{\Pi}\mu + \boldsymbol{\chi}$ whose solution is given by $\mu = (\mathbf{I} - \mathbf{\Pi})^{-1}\boldsymbol{\chi}$ The inverse exists for terminating plant models [5] because $\mathbf{\Pi}$ is a contraction operator [4] due to the strict inequality $\sum_j \Pi_{ij} < 1$. The residual $\theta_i = 1 - \sum_j \Pi_{ij}$ is referred to as the termination probability for state $q_i \in Q$. We extend the analysis to non-terminating plants with stochastic transition probability matrices (*i.e.* with $\theta_i = 0,\ \forall q_i \in Q$) by renormalizing the language measure [1] with respect to the uniform termination probability of a limiting terminating model as described next.

Let $\widetilde{\Pi}$ and Π be the stochastic event generation and transition probability matrices for a non-terminating plant $G_i = \langle Q, \Sigma, \delta, q_i, Q_m \rangle$. We consider the terminating plant $G_i(\theta)$ with the same DFSA structure $\langle Q, \Sigma, \delta, q_i, Q_m \rangle$ such that the event generation probability matrix is given by $(1-\theta)\widetilde{\Pi}$ with $\theta \in (0, 1)$ implying that the state transition probability matrix is $(1-\theta)\Pi$.

Definition 2.7: (Renormalized Measure:) *The renormalized measure* $\nu_\theta^i : 2^{L(q_i(\theta))} \to [-1, 1]$ *for the* θ*-parametrized terminating plant* $G_i(\theta)$ *is defined as:*

$$\forall \omega \in L(q_i(\theta)),\ \nu_\theta^i(\{\omega\}) = \theta\mu^i(\{\omega\}) \tag{2}$$

The corresponding matrix form is given by $\boldsymbol{\nu_\theta} = \theta\ \boldsymbol{\mu} = \theta\ [I - (1-\theta)\Pi]^{-1}\boldsymbol{\chi}$ with $\theta \in (0, 1)$*. We note that the vector representation allows for the following notational simplification* $\nu_\theta^i(L(q_i(\theta))) = \boldsymbol{\nu_\theta}\big|_i$ *The renormalized measure for the non-terminating plant* G_i *is defined to be* $\lim_{\theta \to 0^+} \nu_\theta^i$*.*

A. Event-driven Supervision of PFSA

Plant models considered in this paper are *deterministic* finite state automata (plant) with well-defined event occurrence *probabilities*. In other words, the occurrence of events is probabilistic, but the state at which the plant ends up, *given a particular event has occurred*, is deterministic. Since no emphasis is placed on the initial state and marked states are completely determined by χ, the models can be completely specified by a sextuple as: $G = (Q, \Sigma, \delta, \widetilde{\mathbf{\Pi}}, \chi, \mathscr{C})$

Definition 2.8: (Control Philosophy) *If* $q_i \underset{\sigma}{\longrightarrow} q_k$*, and the event* σ *is disabled at state* q_i*, then the supervisory action is to prevent the plant from making a transition to the state* q_k*, by forcing it to stay at the original state* q_i*. Thus disabling any transition* σ *at a given state* q *results in deletion of the original transition and appearance of the self-loop* $\delta(q, \sigma) = q$ *with the occurrence probability of* σ *from the state* q *remaining unchanged in the supervised and unsupervised plants. For a given plant, transitions that can be disabled in the sense of Definition 2.8 are defined to be controllable transitions. The set of controllable transitions in a plant is denoted* $\mathscr{C}$*. Note controllability is state-based.*

B. The Optimal Supervision Problem: Formulation & Solution

A supervisor disables a subset of the set $\mathscr{C}$ of controllable transitions and hence there is a bijection between the set of all possible supervision policies and the power set $2^{\mathscr{C}}$. That is, there exists $2^{|\mathscr{C}|}$ possible supervisors and each supervisor is uniquely identifiable with a subset of $\mathscr{C}$ and the language measure ν allows a quantitative comparison of different policies.

Definition 2.9: *For an unsupervised plant* $G = (Q, \Sigma, \delta, \widetilde{\Pi}, \chi, \mathscr{C})$*, let* $G^\dagger$ *and* $G^\ddagger$ *be the supervised plants with sets of disabled transitions,* $\mathscr{D}^\dagger \subseteq \mathscr{C}$ *and* $\mathscr{D}^\ddagger \subseteq \mathscr{C}$*, respectively, whose measures are* $\boldsymbol{\nu}^\dagger$ *and* $\boldsymbol{\nu}^\ddagger$*. Then, the supervisor that disables* $\mathscr{D}^\dagger$ *is defined to be superior to the supervisor that disables* $\mathscr{D}^\ddagger$ *if* $\boldsymbol{\nu}^\dagger \geqq_{\texttt{(Elementwise)}} \boldsymbol{\nu}^\ddagger$ *and strictly superior if* $\boldsymbol{\nu}^\dagger >_{\texttt{(Elementwise)}} \boldsymbol{\nu}^\ddagger$*.*

Definition 2.10: (Optimal Supervision Problem) *Given a (non-terminating) plant* $G = (Q, \Sigma, \delta, \widetilde{\Pi}, \chi, \mathscr{C})$*, the problem is to compute a supervisor that disables a subset* $\mathscr{D}^\star \subseteq \mathscr{C}$*, such that*

$\nu^\star \geqq_{(\texttt{Elementwise})} \nu^\dagger \quad \forall \mathscr{D}^\dagger \subseteq \mathscr{C}$ *where* $\nu^\star$ *and* $\nu^\dagger$ *are the measure vectors of the supervised plants* $G^\star$ *and* $G^\dagger$ *under* $\mathscr{D}^\star$ *and* $\mathscr{D}^\dagger$*, respectively.*

Remark 2.1: *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* $(1-\theta)\widetilde{\Pi}$ *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* θ *in each iteration step of the design algorithm. However, choosing* θ *too small may cause numerical problems in convergence. Algorithms reported in [2], [7] computes how small a* θ *is actually required, i.e., computes the critical lower bound* $\theta_\star$*, thus solving the optimal supervision problem for a generic PFSA. It is further shown that the solution obtained is optimal and unique and can be computed by an effective algorithm.*

Definition 2.11: *Following Remark 2.1, we note that algorithms reported in [2], [7] compute a lower bound for the critical termination probability for each iteration of such that the disabling/enabling decisions for the terminating plant coincide with the given non-terminating model. We define* $\theta_{min} = \min_k \theta_\star^{[k]}$ *where* $\theta_\star^{[k]}$ *is the termination probability computed in the* k^{th} *iteration.*

Definition 2.12: *If G and* $G^\star$ *are the unsupervised and supervised PFSA respectively then we denote the renormalized measure of the terminating plant* $G^\star(\theta_{min})$ *as* $\nu_\#^i : 2^{L(q_i)} \rightarrow [-1,1]$ *(See Definition 2.7). Hence, in vector notation we have:* $\nu_\# = \theta_{min}[I-(1-\theta_{min})\Pi^\#]^{-1}\chi$ *where* $\Pi^\#$ *is the transition probability matrix of the supervised plant* $G^\star$*, we note that* $\nu_\# = \nu^{[K]}$ *where K is the total number of iterations required for convergence.*

C. Problem Formulation: A PFSA Model of Autonomous Navigation

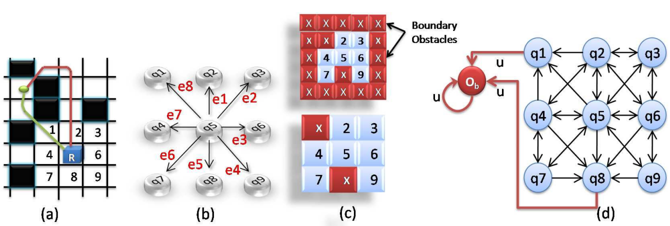


Fig. 1. **(a)** shows the vehicle (marked "R") with the obstacle positions shown as black squares. The green dot identifies the goal **(b)** shows the finite state representation of the possible one-step moves from the current position. **(d)** shows uncontrollable transitions "u" from states corresponding to blocked grid locations to "$q_\ominus$"

We consider a 2D workspace for the mobile agents. This restriction on workspace dimensionality serves to simplify the exposition and can be easily relaxed. To set up the problem, the workspace is first discretized into a finite grid and hence the approach developed in this paper falls under the generic category of discrete planning. The underlying theory does not require the grid to be regular; however for the sake of clarity we shall present the formulation under the assumption of a regular grid. The obstacles are represented as blocked-off grid locations in the discretized workspace. We specify a particular location as the fixed goal and consider the problem of finding optimal and feasible paths from arbitrary initial grid locations in the workspace. Figure 1(a) illustrates the basic problem setup. We further assume that at any given time instant the robot occupies one particular location (*i.e.* a particular square in Figure 1(a)). As shown in Figure 1, the robot has eight possible moves from any interior location. The boundaries are handled by removing the moves that take the robot out of the workspace. The possible moves are modeled as controllable transitions between grid locations since the robot can "choose" to execute a particular move from the available set. We note that the number of possible moves (8 in this case) depends on the chosen fidelity of discretization of the robot motion and also on the intrinsic vehicle dynamics. The complexity results presented in this paper only assumes that the number of available moves is significantly smaller compared to the number of grid squares, *i.e.*, the discretized position states. Specification of inter-grid transitions in this manner allows us to generate a finite state automaton (FSA) description of the navigation problem. Each square in the discretized workspace is modeled as a FSA state with the controllable transitions defining the corresponding state transition map. The formal description of the model is as follows:

Let $\mathcal{G}_{\mathsf{NAV}} = (Q,\Sigma,\delta,\widetilde{\Pi},\chi)$ be a Probabilistic Finite State Automaton (PFSA). The state set Q consists of states that correspond to grid locations and one extra state denoted by $q_\ominus$. The necessity of this special state $q_\ominus$ is explained in the sequel. The grid squares are numbered in a pre-determined scheme such that each $q_i \in Q \setminus \{q_\ominus\}$ denotes a specific square in the discretized workspace. The particular numbering scheme chosen is irrelevant. In the absence of dynamic uncertainties and state estimation errors, the alphabet contains one uncontrollable event *i.e.* $\Sigma = \Sigma_C \bigcup \{u\}$ such that Σ_C is the set of controllable events corresponding to the possible moves of the robot. The uncontrollable event u is defined from each of the blocked states and leads to $q_\ominus$ 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 $q_\ominus$ which is physically interpreted to be a collision. We further assume that the robot fails to recover from collisions which is reflected by making $q_\ominus$ a deadlock state. We note that $q_\ominus$ does not correspond to any physical grid location. The set of blocked grid locations along with the obstacle state $q_\ominus$ is denoted as $Q_{\mathsf{OBSTACLE}} \subsetneqq Q$. 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_8 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:*

$$\chi(q_i) = \begin{cases} -1 & \text{if } q_i \equiv q_\ominus \\ 1 & \text{if } q_i \text{ is the goal} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

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:* $\forall q_i \in Q, \sigma_j \in \Sigma$,

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

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 Σ_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 nomenclature in the path-planning literature, we refer to the language-measure-theoretic algorithm as $\nu^\star$ 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 $\nu^\star$ 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.* Let $\mathcal{G}_{\textsf{NAV}}$ be the unsupervised navigation automaton and $\mathcal{G}^\star_{\textsf{NAV}}$ be the optimally supervised PFSA obtained by $\nu^\star$. We note that $\nu^i_\#$ is the renormalized measure of the terminating plant $\mathcal{G}^\star_{\textsf{NAV}}(\theta_{min})$ with substochastic event generation probability matrix $\widetilde{\Pi}^{\theta_{min}} = (1-\theta_{min})\widetilde{\Pi}$. Denoting the event generating function (See Definition 2.3) for $\mathcal{G}^\star_{\textsf{NAV}}$ and $\mathcal{G}^\star_{\textsf{NAV}}(\theta_{min})$ as $\tilde{\pi} : Q \times \Sigma \to Q$ and $\tilde{\pi}^{\theta_{min}} : Q \times \Sigma \to Q$ respectively:

$$\tilde{\pi}^{\theta_{min}}(q_i, \epsilon) = 1 \tag{4a}$$

$$\forall q_i \in Q, \sigma_j \in \Sigma, \ \tilde{\pi}^{\theta_{min}}(q_i, \sigma_j) = (1-\theta_{min})\tilde{\pi}(q_i, \sigma_j) \tag{4b}$$

Notation 2.1: *For notational simplicity, we use*

$$\nu^i_\#(L(q_i)) = \nu_\#(q_i) = \boldsymbol{\nu}_\#|_i$$

$$\text{where } \boldsymbol{\nu}_\# = \theta_{min}[I - (1-\theta_{min})\Pi^\#]^{-1}\chi$$

Definition 2.16: *($\nu^\star$-path:) A $\nu^\star$-path $\rho(q_i, q_j)$ from state $q_i \in Q$ to state $q_j \in Q$ is defined to be an ordered set of PFSA states $\rho = \{q_{r_1}, \cdots, q_{r_M}\}$ with $q_{r_s} \in Q, \ \forall s \in \{1, \cdots, M\}, M \leq \textsf{Card}(Q)$ such that*

$$q_{r_1} = q_i \tag{5a}$$

$$q_{r_M} = q_j \tag{5b}$$

$$\forall i, j \in \{1, \cdots, M\}, \ q_{r_i} \neq q_{r_j} \tag{5c}$$

$$\forall s \in \{1, \cdots, M\}, \forall t \leqq s, \ \nu_\#(q_{r_t}) \leqq \nu_\#(q_{r_s}) \tag{5d}$$

We reproduce without proof the following key results pertaining to $\nu^\star$- planning as reported in [8].

Lemma 2.1: *There exists an enabled sequence of transitions from state $q_i \in Q \setminus Q_{\textsf{Obstacle}}$ to $q_j \in Q \setminus \{q_\ominus\}$ in $\mathcal{G}^\star_{\textsf{NAV}}$ if and only if there exists a $\nu^\star$-path $\rho(q_i, q_j)$ in $\mathcal{G}^\star_{\textsf{NAV}}$.*

Proposition 2.1: *For the optimally supervised navigation automaton $\mathcal{G}^\star_{\textsf{NAV}}$, we have*

$$\forall q_i \in Q \setminus Q_{\textsf{Obstacle}}, \ L(q_i) \subseteqq \Sigma^\star_C$$

Corollary 2.1: *(Obstacle Avoidance:) There exists no $\nu^\star$-path from any unblocked state to any blocked state in the optimally supervised navigation automaton $\mathcal{G}^\star_{\textsf{NAV}}$.*

Proposition 2.2: *(Existence of $\nu^\star$-paths:) There exists a $\nu^\star$-path $\rho(q_i, q_{\textsf{Goal}})$ from any state $q_i \in Q$ to the goal $q_{\textsf{Goal}} \in Q$ if and only if $\nu_\#(q_i) > 0$.*

Corollary 2.2: *(Absence of Local Maxima:) If there exists a $\nu^\star$-path from $q_i \in Q$ to $q_j \in Q$ and a $\nu^\star$-path from q_i to $q_{\textsf{Goal}}$ then there exists a $\nu^\star$-path from q_j to $q_{\textsf{Goal}}$, i.e.,*

$$\forall q_i, q_j \in Q\Big(\exists \rho_1(q_i, q_{\textsf{Goal}}) \bigwedge \exists \rho_2(q_i, q_j) \Rightarrow \exists \rho(q_j, q_{\textsf{Goal}})\Big)$$

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 $\nu^\star$ 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 ω is the shortest path to goal from state q_k, then the shortest path from state q_i (with $q_i \xrightarrow{\sigma_2} q_k$) is given by $\sigma_2\omega$. However, a larger number of feasible paths may be available from state q_j (with $q_i \xrightarrow{\sigma_1} q_j$) which may result in the optimal $\nu^\star$ plan to be $\sigma_1\omega_1$. Mathematically, each feasible path from state q_j has a positive measure which may sum to be greater than the measure of the single path ω from state q_k. The condition $\nu_\#(q_j) > \nu_\#(q_k)$ 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 $\mathcal{G}^\star_{\textsf{NAV}}$, we have $\forall q_i \in Q \setminus Q_{\textsf{Obstacle}}$,*

$$\forall \omega \in L(q_i), \ \nu^i_\#(\{\omega\}) = \theta_{min}\left(\frac{1-\theta_{min}}{\textsf{Card}(\Sigma_C)}\right)^{|\omega|}\chi(\delta^\#(q_i, \omega))$$

Proposition 3.1: *For $q_i \in Q \setminus Q_{\textsf{Obstacle}}$, let $q_i \xrightarrow{\sigma_1} q_j \to \cdots \to q_{\textsf{Goal}}$ be the shortest path to the goal. If there exists $q_k \in Q \setminus Q_{\textsf{Obstacle}}$ with $q_i \xrightarrow{\sigma_2} q_k$ for some $\sigma_2 \in \Sigma_C$ such that $\nu_\#(q_k) > \nu_\#(q_j)$, then the number of distinct paths to goal from state q_k is at least $\textsf{Card}(\Sigma_C) + 1$.*

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 $\nu^\star$ 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.

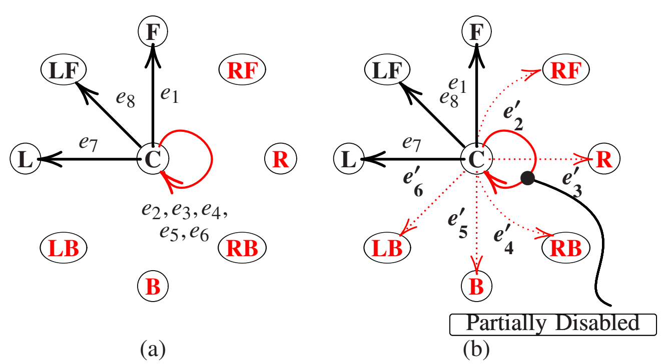


Fig. 2. (a) shows the enabled moves in the optimally supervised PFSA with no dynamic uncertainty, (b) illustrates the case with dynamic uncertainty, so that the robot can still uncontrollably (and hence unwillingly) make the disabled transitions, albeit with a small probability, *i.e.*, probability of transitions e'_2, e'_3, e'_4 etc. is small.

Definition 3.1: *The modified navigation automaton $= (Q, \Sigma, \delta, \widetilde{\Pi}, \chi,)$ 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*

$$\sigma = \{\sigma_{controllable}, \sigma_{uncontrollable}\} \tag{6}$$

with the transition probabilities distributing as

$$\forall q_i \in Q,\ \widetilde{\Pi}(q_i, \sigma_{controllable}) = \widetilde{\Pi}(q_i, \sigma) \tag{7}$$

$$\forall q_i \in Q,\ \widetilde{\Pi}(q_i, \sigma_{uncontrollable}) = (1-)\widetilde{\Pi}(q_i, \sigma) \tag{8}$$

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 Π for decomposes as $\Pi = \Pi + (1-)\Pi$, where Π 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 $\nu^\star$-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 $\nu^\star$-path $\rho(q_i, q_{\text{GOAL}})$ 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 ω of positive measure initiating from q_i and hence there exists at least one string that terminates on q_{GOAL}. The proof then follows from the definition of $\nu^\star$-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 $q_i \in Q$, $\nu^1_\#(q_i) > 0$ and $\nu^2_\#(q_i) > 0$, then the shortest $\nu^\star$-path from q_i to q_{GOAL} in $|_2$ is at least as long as the corresponding shortest $\nu^\star$-path from q_i to q_{GOAL} in $|_1$.*

Proof: We use induction on the length of the shortest $\nu^\star$-path from q_i to q_{GOAL} in $|_2$, which we denote as ℓ_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 fro $\ell_1 = k$. Then, for $\ell_1 = k+1$, we note that if $q_i, q_{r_1}, \cdots, q_{r_k}$ is the shortest $\nu^\star$-path in $|_2$, then the shortest $\nu^\star$-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_{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 $\nu^\star$-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 $\wp^T \chi$ where $\wp$ is the stationary probability vector on the automaton states [2]. Since $\chi(q_{\text{GOAL}}) = 1$ and $\chi(q_{\text{OBSTACLE}}) = -1$ 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

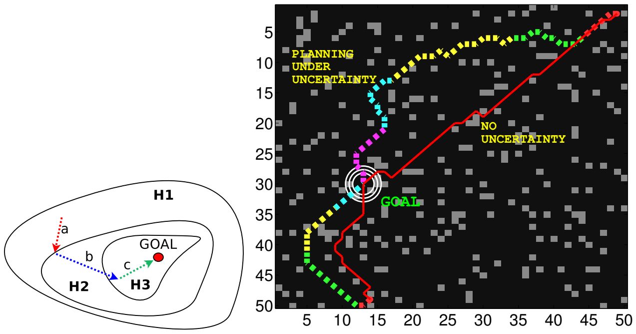
The proposed modification is validated on a SEGWAY RMP 200 which is a two-wheeled robot with significant dynamic uncertainty. In particular, the inverted-pendulum dynamics prevents the platform from halting instantaneously. The experimental runs were conducted at the Networked Robotics & Systems Laboratory (NRSL), Pennstate, with the workspace discretized into a 53×29 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 . In the top plate, $= 0.98$ with average robot speed $v = 0.3m/sec$. The middle plate illustrates the case with $= 0.9, v = 0.5m/sec$ and for the bottom plate the values are $= 0.85, v = 1m/sec$. The plates on the lefthand 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 maximacan occur under the modified model. This is a serious problem for autonomous planning and must be remedied. Local Maxima elation is notoriously difficult for potential based planning approaches. The problem becomes critically important when applied to solution of mazes; larger the number of obstables, higher is the chance of ending up in a local maxima. However, $\nu^\star$ 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 $k \neq j$.

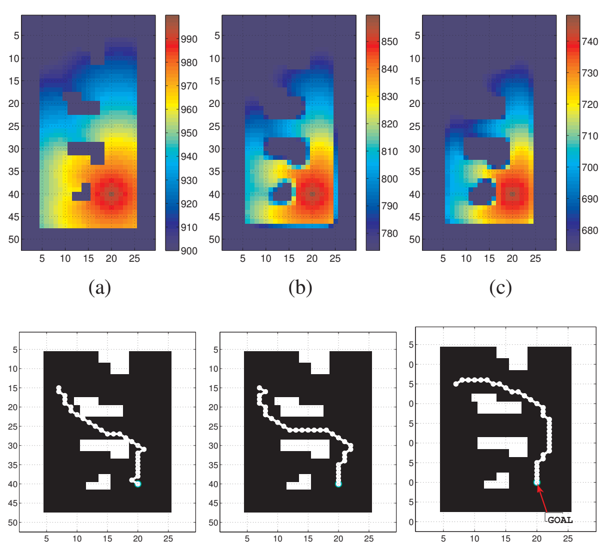


(a) (b)

Fig. 3. (a)Illustration for Statement 2 of Proposition 4.1 (b)Simulated maze solution under dynamic uncertainties in plan execution. The broken line corresponds to the plan considering dynamic uncertainty; while the solid line is the one without. The color coding on the broken line switches when the plan moves from H_i to H_{i-1}.

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 $\nu_{\#}(q_i) > 0$ w.r.t. the plan saved in M_1, implying that there is a $\nu^\star$-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 $M_k = \boldsymbol{\nu}_{\#}$. 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 $\nu^\star$-path to some state $q_j \in H_k$, which in turn implies the existence of a succession of *nustar*-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* $\nu^\star$ *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) (b) (c)

(d) (e) (f)

Fig. 4. Experimental Validation at NRSL, Pennstate: Navigable area shrinks as uncertainty parameter is increased from top to bottom with consequent change in the plan

A. Simulation Example

Recursive $\nu^\star$ 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 $\nu^\star$ 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 ν-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 modeled as unobservable transitions and will be addressed in future publications.

References

[1] I. Chattopadhyay and A. Ray, "Renormalized measure of regular languages," *Int. J. Control*, vol. 79, no. 9, pp. 1107–1117, 2006.
[2] I. Chattopadhyay and A. Ray, "Language-measure-theoretic optimal control of probabilistic finite-state systems," *Int. J. Control*, August,2007.
[3] J. Barraquand, B. Langlois, and J.-C. Latombe, *Robot motion planning with many degrees of freedom and dynamic constraints*. Cambridge, MA, USA: MIT Press, 1990.
[4] A. Ray, "Signed real measure of regular languages for discrete-event supervisory control," *Int. J. Control*, vol. 78, no. 12, pp. 949–967, 2005.
[5] V. Garg, "An algebraic approach to modeling probabilistic discrete event systems," *Proceedings of 1992 IEEE Conference on Decision and Control*, pp. 2348–2353, Tucson, AZ, December 1992.
[6] W. Rudin, *Real and Complex Analysis, 3rd ed.* McGraw Hill, New York, 1988.
[7] I. Chattopadhyay, "Quantitative control of probabilistic discrete event systems," *PhD Dissertation, Dept. of Mech. Engg. Pennsylvania State University, http:// etda.libraries.psu.edu / theses / approved / WorldWideIndex / ETD-1443*, 2006.
[8] I. Chattopadhyay, G. Mallapragada, and A. Ray, "$\nu^\star$: a robot path planning algorithm based on renormalized measure of probabilistic regular languages," *International Journal of Control*, in press.