2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

$L^\star$: An Intelligent Path Planning Algorithm based on Renormalized Measure of Probabilistic Regular Languages*

Ishanu Chattopadhyay
ixc128@psu.edu

Goutham Mallapragada
grm150@psu.edu

Asok Ray
axr2@psu.edu

The Pennsylvania State University
University Park, PA 16802, USA

***Abstract*— A novel path planning algorithm $L^\star$ is introduced that reduces the problem to optimization of a probabilistic finite state machine and applies the rigorous theory of language-measure-theoretic optimal control to compute ν-optimal paths to the specified goal. It is shown that although the underlying navigation model is probabilistic, the proposed algorithm computes plans that can be executed in a deterministic sense with automated optimal trade-off between path length and robustness under dynamic uncertainty. The algorithm has been validated on mobile robotic platforms in a laboratory environment.**

***Index Terms*— Language Measure; Finite Markov Chains; Discrete Event Systems; Supervisory Control**

1. Introduction & Motivation

The field of trajectory and motion planning is enormous, with applications in such diverse areas as industrial robots, mobile robot navigation, spacecraft reentry, video games and even drug design. In the context of planning for mobile robots and manipulators much of the literature on path and motion planning is concerned with finding collision-free trajectories [7]. Graph-based techniques have been used very successfully in many wheeled ground robot path planning problems and have been used for some UAV planning problems, typically radar evasion [1]. These approaches suffer from exponential growth in complexity with both graph size and dimension. To circumvent the complexity associated with graph-based planning, sampling based planning methods [2] such as probabilistic roadmaps have become widespread. These approaches are probabilistically complete (i.e. if a feasible solution exists it will be found, given enough time) but there is no guarantee of finding a solution within a specified time. This paper introduces a novel approach to path-planning 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. The key advantages are as follows:

1) **$L^\star$ pre-processing is cheap:** Conventional cellular approaches decompose the set of free configurations into simple non-overlapping regions [7][8] with the adjacency relations and reachability constraints represented in a connectivity graph which is subsequently searched for a path. Construction of the connectivity graph that faithfully reflects the continuous C-Space constraints (*e.g.* obstacles) is expensive and often intractable for high dimensional problems. The decomposition required by $L^\star$ is simple and computationally cheap. The cells are mapped to PFSA states which are defined to have identical connectivity via symbolic inter-state transitions. Desirable or ”good” states assigned relatively positive weights to undesirable or ”bad” states. A change in the C-Space constraints therefore requires updating the 1D state weight vector only; recomputing the decomposition is unnecessary.
2) **$L^\star$ is fundamentally different from a search:** $L^\star$ replaces the ”search” problem to solution of a sequence of linear systems (i.e. matrix inversions). On completion of cellular decomposition, $L^\star$ optimizes the resultant PFSA via a iterative sequence of combinatorial operations which elementwise maximizes the language measure vector [4][5]. Note that although $L^\star$ involves probabilistic reasoning, the final waypoint sequence obtained is deterministic.
3) **Computational efficiency:** The intensive step in $L^\star$ is the inversion of certain specialized matrices to compute the language measure. 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 [2]. However, $L^\star$ 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 $L^\star$.

The paper is organized in five sections including the present one. Section 2 briefly explains the language-theoretic models considered in this paper and reviews the language-measure-theoretic optimal control of probabilistic finite state machines. Section 3 presents the basic problem formulation with Section 4 deriving a decision-theoretic solution to the path-planning problem. 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 [9]. 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; $\delta : Q \times \Sigma \rightarrow Q$

*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.

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 [6] 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 $\theta \in (0, 1)$;

(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 [9] 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 [10], 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 [9] 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 $\boldsymbol{\mu} = [\mu_1\ \mu_2\ \cdots\ \mu_n]^T$, is called the $\boldsymbol{\mu}$-vector. In vector form, we have $\boldsymbol{\mu} = \mathbf{\Pi}\mu + \boldsymbol{\chi}$ whose solution is given by $\boldsymbol{\mu} = (\mathbf{I} - \mathbf{\Pi})^{-1}\boldsymbol{\chi}$ The inverse exists for terminating plant models [6] because $\mathbf{\Pi}$ is a contraction operator [9] 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 [4] 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-parameterized 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 $\lim_{\theta \to o^+} \nu_\theta^i$.

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: $G = (Q, \Sigma, \delta, \widetilde{\Pi}, \chi, \mathscr{C})$

Definition 2.8: (Control Philosophy) *If $q_i \underset{\sigma}{\rightarrow} q_k$, 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 $\boldsymbol{\nu}^\star \geqq_{\texttt{(Elementwise)}} \boldsymbol{\nu}^\dagger \quad \forall \mathscr{D}^\dagger \subseteq \mathscr{C}$ where $\boldsymbol{\nu}^\star$ and $\boldsymbol{\nu}^\dagger$ 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 [5], [3] by designing an optimal policy for a terminating plant [6] 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 [5], [3] 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 [5], [3] 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: $\boldsymbol{\nu}_\# = \theta_{min}[I - (1-\theta_{min})\Pi^\#]^{-1}\boldsymbol{\chi}$ where $\Pi^\#$ is the transition probability matrix of the supervised plant $G^\star$, we note that $\boldsymbol{\nu}_\# = \nu^{[K]}$ where K is the total number of iterations required for convergence.*

3. Problem Formulation: A PFSA Model of Autonomous Navigation

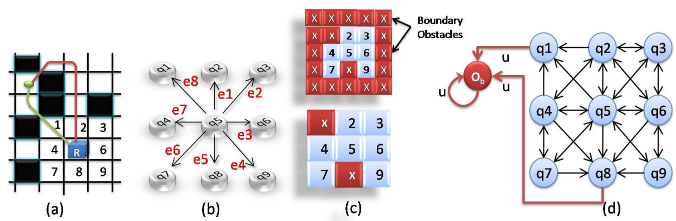

Fig. 1. **(a)** shows the vehicle (marked "R") with the obstacle positions shown as black squares. The green dot identifies the goal **(b)** shows the finite state representation of the possible one-step moves from the current position. **(d)** shows uncontrollable transitions "u" from states corresponding 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 $\mathbb{G}_{\mathrm{NAV}} = (Q, \Sigma, \delta, \widetilde{\Pi}, \chi)$ 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 $Q_{\text{OBSTACLE}} \subsetneqq Q$. 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 [5] assigns scalar weights to the PFSA states to capture the desirability of ending up in each state.

Definition 3.1: *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 3.2: *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 3.3: *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.*

4. 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 [5] described in Section 2. Keeping in line with nomeclature in the path-planning literature, we refer to the language-measure-theoretic algorithm as $\boldsymbol{L^\star}$ 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 elementwise 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 $\boldsymbol{L^\star}$ 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 $\mathbb{G}_{\text{NAV}}$ be the unsupervised navigation automaton and $\mathbb{G}^\star_{\text{NAV}}$ be the optimally supervised PFSA obtained by $\boldsymbol{L^\star}$. We note that $\nu^i_\#$ is the renormalized measure of the terminating plant $\mathbb{G}^\star_{\text{NAV}}(\theta_{min})$ 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 $\mathbb{G}^\star_{\text{NAV}}$ and $\mathbb{G}^\star_{\text{NAV}}(\theta_{min})$ as $\tilde{\pi} : Q \times \Sigma \rightarrow Q$ and $\tilde{\pi}^{\theta_{min}} : Q \times \Sigma \rightarrow Q$ respectively, we have

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

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

Notation 4.1: *For notational simplicity, we use*

$$\nu^i_\#(L(q_i)) = \nu_\#(q_i) = \boldsymbol{\nu_\#}|_i$$
$$\text{where } \boldsymbol{\nu_\#} = \theta_{min}[I - (1 - \theta_{min})\Pi^\#]^{-1}\boldsymbol{\chi}$$

Definition 4.1: ***($L^\star$-path:)*** *A $L^\star$-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 $q_{r_s} \in Q$, $\forall s \in \{1, \cdots, M\}, M \leqq \text{CARD}(Q)$ such that*

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

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

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

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

Lemma 4.1: *There exists an enabled sequence of transitions from state $q_i \in Q \setminus Q_{\text{OBSTACLE}}$ to $q_j \in Q \setminus \{q_\ominus\}$ in $\mathbb{G}^\star_{\text{NAV}}$ if and only if there exists a $\boldsymbol{L^\star}$-path $\rho(q_i, q_j)$ in $\mathbb{G}^\star_{\text{NAV}}$.*

Proof: Let $q_i \xrightarrow{\sigma_1} q_{k1} \xrightarrow{\sigma_2} q_{k2} \xrightarrow{\sigma_3} \cdots q_{k\ell} \rightarrow \cdots \rightarrow q_j$ be an enabled sequence of transitions in $\mathbb{G}^\star_{\text{NAV}}$ where $\ell \in \{1, \cdots M\}$ for some $M \leqq \text{CARD}(Q)$. We note from the defined structure of the navigation automaton that there is no transition $q_s \xrightarrow{\sigma} q_r$ where $q_s \in Q_{\text{OBSTACLE}}, q_r \neq q_\ominus$. It follows that $\forall \ell \in \{1, \cdots, M\},\ q_{k\ell} \notin Q_{\text{OBSTACLE}}$. This in turn implies that each of the individual transitions $q_{k\ell 1} \xrightarrow{\sigma} q_{k\ell 2}$ in the above transition sequence is controllable. Monotonic convergence of the optimizaton algorithm [5], [3] then implies $\nu_\#(q_i) \leq \nu_\#(q_{k1}) \leq \nu_\#(q_{k2}) \leq \cdots \leq \nu_\#(q_j)$ implying that $\{q_i, q_{k1}, q_{k2}, \cdots, q_j\}$ is a $\boldsymbol{L^\star}$-path. The converse follows by a similar argument. ❐

Proposition 4.1: *For the optimally supervised navigation automaton $\mathbb{G}^\star_{\text{NAV}}$, we have*

$$\forall q_i \in Q \setminus Q_{\text{OBSTACLE}},\ L(q_i) \subsetneqq \Sigma^\star_C$$

Proof: We proceed by the method of contradiction. Assume $\exists s \in L(q_i)$ for some $q_i \in Q \setminus Q_{\text{OBSTACLE}}$ such that $s \notin \Sigma^\star_C$. Since $u \in \Sigma$ is the only uncontrollable event and

$$\delta^\#(q_j, u) = q_\ominus \text{ (if defined)} \tag{6}$$

$$\delta^\#(q_\ominus, u) = q_\ominus \text{ (Self-loop)} \tag{7}$$

it follows that $s = s_1 u^K$ where $s_1 \in \Sigma_C^\star$ and $K \geqq 1$. Since u is only defined from $q_j \in Q_{\text{OBSTACLE}}$, it follows that there exists an enabled sequence of controllable transitions from q_i to $q_j \in Q_{\text{OBSTACLE}} \setminus \{q_\ominus\}$ which in turn implies (See Lemma) that

$$\nu_\#(q_i) \leqq \nu_\#(q_j) \tag{8}$$

Since $L(q_j) = u^\star$, it follows from Definition that

$$\nu_\#(q_j) = \theta_{min} \sum_{k=2}^{\infty} (1-\theta_{min})^k \times -1 = -(\theta_{min} - 1)^2 < 0 \tag{9}$$

$$\begin{aligned} \nu_\#(q_i) &= \nu_\#(\omega \in L(q_i) s.t. \delta^\#(q_i,\omega) \notin Q_{\text{OBSTACLE}}) \\ &+ \nu_\#(\omega \in L(q_i) s.t. \delta^\#(q_i,\omega) \in Q_{\text{OBSTACLE}}) \end{aligned} \tag{10}$$

Now, since only $q_\ominus$ has a negative characteristic weight (See Definition), we have $\nu_\#(\omega \in L(q_i) s.t. \delta^\#(q_i,\omega) \notin Q_{\text{OBSTACLE}}) \geqq 0$. On the other hand, $q_i \notin Q_{\text{OBSTACLE}}$ implies

$$\begin{aligned} &\nu_\#(\omega \in L(q_i) : \delta^\#(q_i,\omega) \in Q_{\text{OBSTACLE}}) = \alpha \nu_\#(q_r), \\ &\qquad \text{where } q_r \in Q_{\text{OBSTACLE}} \setminus \{q_\ominus\} \\ &\Rightarrow \nu_\#(\omega \in L(q_i) : \delta^\#(q_i,\omega) \in Q_{\text{OBSTACLE}}) = \alpha \nu_\#(q_j) \end{aligned} \tag{11}$$

where $\alpha = \theta_{min} \sum_{\omega : \delta^\#(q_i,\omega) \in Q_{\text{OBSTACLE}} \setminus \{q_\ominus\}} \tilde{\pi}^{\theta_{min}}(q_i,\omega)$. We observe that $\alpha \in [0,1)$ with the strict upper bound $\alpha < 1$ following from the consideration of the worst case scenario where $\delta^\#(q_i,\sigma) \in Q_{\text{OBSTACLE}}$, $\forall \sigma \in \Sigma_C$, in which case we have

$$\alpha = \theta_{min} \frac{1}{\text{CARD}(\Sigma_C)} (1-\theta_{min}) \times \text{CARD}(\Sigma_C) = \theta_{min} - \theta_{min}^2 < 1$$

Hence we conclude $\nu_\#(q_i) \leqq \nu_\#(q_j)$ which contradicts Eq. (8). ❐

Corollary 4.1: (Obstacle Avoidance:) *There exists no $L^\star$-path from any unblocked state to any blocked state in the optimally supervised navigation automaton $\mathbb{G}_{\text{NAV}}^\star$.*

Proof: Let $q_i \in Q \setminus Q_{\text{OBSTACLE}}$ and assume that there exists a $L^\star$-path from q_i to some $q_j \in Q_{\text{OBSTACLE}} \setminus \{q_\ominus\}$. It follows from Lemma 4.1 that there exists an enabled sequence of transitions from q_i to q_j. Since the uncontrollable transition u is defined from $q_j \in Q_{\text{OBSTACLE}}$ (See Definition), it follows that $L(q_i) \nsubseteq \Sigma_C^\star$ which contradicts Proposition 4.1. ❐

Proposition 4.2: (Existence of $L^\star$-paths:) *There exists a $L^\star$-path $\rho(q_i, q_{\text{GOAL}})$ from any state $q_i \in Q$ to the goal $q_{\text{GOAL}} \in Q$ if and only if $\nu_\#(q_i) > 0$.*

Proof: Partitioning $L(q_i)$ based on terminating states,

$$\begin{aligned} \nu_\#(q_i) &= \nu_\#(\{\omega : \delta^\#(q_i,\omega) = q_{\text{GOAL}}\}) \\ &+ \nu_\#(\{\omega : \delta^\#(q_i,\omega) \in Q \setminus (Q_{\text{OBSTACLE}} \textstyle\bigcup \{q_{\text{GOAL}}\})\}) \\ &+ \nu_\#(\{\omega : \delta^\#(q_i,\omega) \in Q_{\text{OBSTACLE}}\}) \end{aligned} \tag{12}$$

Now, Corollary 4.1 implies $\{\omega : \delta^\#(q_i,\omega) \in Q_{\text{OBSTACLE}}\} = \varnothing \Rightarrow \nu_\#(\{\omega : \delta^\#(q_i,\omega) \in Q_{\text{OBSTACLE}}\}) = 0$. Also, since $\forall q_j \in Q \setminus (Q_{\text{OBSTACLE}} \bigcup \{q_{\text{GOAL}}\})$, $\chi(q_j) = 0$, we have $\nu_\#(\{\omega : \delta^\#(q_i,\omega) \in Q \setminus (Q_{\text{OBSTACLE}} \bigcup \{q_{\text{GOAL}}\})\}) = 0$. Hence we conclude

$$\nu_\#(q_i) = \nu_\#(\{\omega : \delta^\#(q_i,\omega) = q_{\text{GOAL}}\}) \tag{13}$$

It follows from $\chi(q_{\text{GOAL}}) = 1$ that $\delta^\#(q_i,\omega) = q_{\text{GOAL}} \Rightarrow \nu_\#(\{\omega\}) > 0$ implying there exists an enabled sequence of controllable transitions from q_i to q_{GOAL} in $\mathbb{G}_{\text{NAV}}^\star$ if and only if $\nu_\#(q_i) > 0$. The result then follows from Lemma 4.1. ❐

Corollary 4.2: (Absence of Local Maxima:) *If there exists a $L^\star$-path from $q_i \in Q$ to $q_j \in Q$ and a $L^\star$-path from q_i to q_{GOAL} then there exists a $L^\star$-path from q_j to q_{GOAL}, i.e.,*

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

Proof: Let $q_i, q_j \in Q$. We note

$$\exists \rho_1(q_i, q_{\text{GOAL}}) \Rightarrow \nu_\#(q_i) > 0 \quad \text{(Proposition 4.2)} \tag{14a}$$

$$\exists \rho_2(q_i, q_j) \Rightarrow \nu_\#(q_i) \leq \nu_\#(q_j) \quad \text{(Definition 4.1)} \tag{14b}$$

It follows that $\nu_\#(q_j) > 0$ which implies that there exists a $L^\star$-path from q_j to q_{GOAL} (Proposition 4.2). ❐

```
input  : Current State q_CURRENT, ν#, δ
output: Next State q_NEXT
begin
    Set M = -Inf;
    for j = 1 to CARD(Σ_C) do
        q_j = δ(q_CURRENT, σ_j);
        if ν#(q_j) > M then
            q_NEXT = q_j;
            M = ν#(q_j);


end
```

Algorithm 1: Computation of Next State

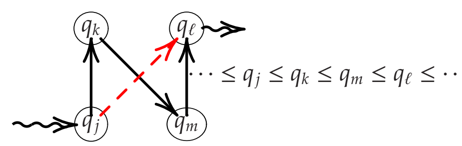

Fig. 2. States q_j, q_k, q_m, q_ℓ with $q_j \leq q_k \leq q_m \leq q_\ell$ implying $\{q_j, q_k, q_m, q_\ell\}$ is a $L^\star$-path from q_j to q_ℓ; smoothed path is $\{q_j, q_\ell\}$

Remark 4.1: (Smoothing computed $L^\star$-paths:) *We note that given an arbitrary current state q_i, Algorithm 1 computes the next state q_{NEXT} as the one which has maximal measure among all possible next states q_j satisfying $\nu_\#(q_j) \geqq \nu_\#(q_i)$. This precludes the possibility illustrated in Figure 2 implying that the computed plan is relatively smoother. However, such smoothing may result in a computed plan that is not necessarily the shortest path from the initial state to the goal. This is expected since $L^\star$ is not meant to determine the shortest path, but compute a ν-optimal path to the goal. In the sequel, we expound this optimal tradeoff that $L^\star$ makes between path length and robust planning.*

Notation 4.2: *We denote the $L^\star$-path computed by Algorithm 1 as a $L^\star$-plan in the sequel.*

A. Optimal Tradeoff between Computed Path Length & Robustness to Dynamic 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 $L^\star$ 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 $\boldsymbol{L}^\star$ 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 by Algorithm 1. 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. Next we estimate the *minimum number of alternate routes required for the* $\boldsymbol{L}^\star$ *plan to be different from the shortest path*.

Lemma 4.2: *For the optimally supervised navigation automaton* $\mathbb{G}^\star_{\mathsf{NAV}}$*, we have* $\forall q_i \in Q \setminus Q_{\mathsf{OBSTACLE}}$,

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

Proof: We claim that $\forall q_i \in Q \setminus Q_{\mathsf{OBSTACLE}}$,

$$\forall \omega \in L(q_i),\ \tilde{\pi}^{\theta_{min}}(q_i,\omega) = \left(\frac{1-\theta_{min}}{\mathsf{CARD}(\Sigma_C)}\right)^{|\omega|} \quad (15)$$

We note that on account of Proposition 4.1 we need to consider only $\omega \in \Sigma_C^\star$. Eq. (15) is trivially satisfied for $|\omega| = 0$. For $|\omega| \geqq 1$, we proceed by the method of induction. First we note that since the control philosophy (See Definition 2.8) does not alter the event generation probabilities, we have from Definition 3.2 that $\forall q_i \in Q \setminus Q_{\mathsf{OBSTACLE}}, \sigma \in \Sigma_C$, $\tilde{\pi}^{\theta_{min}}(q_i,\sigma) = \frac{1}{\mathsf{CARD}(\Sigma_C)}$ implying Eqn.(15) is satisfied for $|\omega| = 1$. Assume that the result holds for all $\omega \in \Sigma_C^\star$ and for all $q_i \in Q \setminus Q_{\mathsf{OBSTACLE}}$ such that $|\omega| = K$. Then the claim follows from noting

$$\tilde{\pi}^{\theta_{min}}(q_i,\sigma\omega) = \tilde{\pi}^{\theta_{min}}(q_i,\sigma)\tilde{\pi}^{\theta_{min}}(\delta^\#(q_i,\sigma),\omega) = \left(\frac{1-\theta_{min}}{\mathsf{CARD}(\Sigma_C)}\right)^{K+1}$$

Finally we recall Definition 2.12 to note $\nu^i_\#(\{\omega\}) = \theta_{min}\tilde{\pi}^{\theta_{min}}(q_i,\omega)\chi(q_j)$ where $q_j = \delta^\#(q_i,\omega)$. This completes the proof. ❐

Proposition 4.3: *For* $q_i \in Q \setminus Q_{\mathsf{OBSTACLE}}$*, let* $q_i \xrightarrow{\sigma_1} q_j \to \cdots \to q_{\mathsf{GOAL}}$ *be the shortest path to the goal. If there exists* $q_k \in Q \setminus Q_{\mathsf{OBSTACLE}}$ *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* $\mathsf{CARD}(\Sigma_C) + 1$.

Proof: Let the length of the shortest path $q_i \xrightarrow{\sigma_1} q_j \to \cdots \to q_{\mathsf{GOAL}}$ be m. To compute a lower bound on the number of alternate routes, we assume that there is only one path from state q_j to the goal. Further we assume that every alternate path from state q_i through q_k has length $m+1$. If there are r such paths, then for the condition $\nu_\#(q_k) > \nu_\#(q_j)$ to be true, we need

$$\sum_{\omega_1 \in L(q_j)} \nu_\#(\{\omega_1\}) > \sum_{\omega_2 \in L(q_k)} \nu_\#(\{\omega_2\})$$
$$\Rightarrow r \times \theta_{min}\left(\frac{1c-\theta_{min}}{\mathsf{CARD}(\Sigma_C)}\right)^m > \theta_{min}\left(\frac{1-\theta_{min}}{\mathsf{CARD}(\Sigma_C)}\right)^{m-1} \text{ (Lemma 4.2)}$$
$$\Rightarrow r > \mathsf{CARD}(\Sigma_C) \Rightarrow r \geqq \mathsf{CARD}(\Sigma_C) + 1$$

❐

The lower bound computed in Proposition 4.3 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 $\boldsymbol{L}^\star$ opts for a longer but more robust plan.

5. Summary & Future Research

A novel path planning algorithm $\boldsymbol{L}^\star$ 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.

A. Future Work

Future work will extend the language-measure theoretic planning algorithm to address the following problems:

1) **Multi-robot coordinated planning:** Run-time complexity grows exponentially with the number of agents if one attempts to solve the full Cartesian product problem. However $\boldsymbol{L}^\star$ can be potentially used to plan individually followed by an intelligent assembly of the plans to take interaction into account.
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; however care must be taken to ensure that the computed solution is not too far from the optimal one.
3) **Handling uncontrollable dynamic events:** In this paper the only uncontrollable event is the one modeling a collision when the robot attempts to move into a blocked state. Physical dynamic response of the robot may need to be modeled as uncontrolled transitions in highly uncertain environments and will be addressed in future publications.

References

[1] D. A. Anisi, J. Hamberg, and X. Hu, *Nearly time-optimal paths for a ground vehicle*, Journal of Control Theory and Applications, (2003).

[2] J. Barraquand, B. Langlois, and J.-C. Latombe, *Robot motion planning with many degrees of freedom and dynamic constraints*, MIT Press, Cambridge, MA, USA, 1990.

[3] 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).

[4] I. Chattopadhyay and A. Ray, *Renormalized measure of regular languages*, Int. J. Control, 79 (2006), pp. 1107–1117.

[5] ———, *Language-measure-theoretic optimal control of probabilistic finite-state systems*, Int. J. Control, (August,2007).

[6] V. Garg, *An algebraic approach to modeling probabilistic discrete event systems*, Proceedings of 1992 IEEE Conference on Decision and Control, (Tucson, AZ, December 1992), pp. 2348–2353.

[7] K. Kondo, *Motion planning with six degrees of freedom by multistrategicbidirectional heuristic free-space enumeration*, IEEE Transactions on Robotics and Automation, 7 (1991), pp. 267–277.

[8] T. Lozano-Perez, *A simple motion-planning algorithm for general robot manipulators*, IEEE Transactions on Robotics and Automation, 3 (1987), pp. 224–238.

[9] A. Ray, *Signed real measure of regular languages for discrete-event supervisory control*, Int. J. Control, 78 (2005), pp. 949–967.

[10] W. Rudin, *Real and Complex Analysis, 3rd ed.*, McGraw Hill, New York, 1988.