Formalization of Methods

Alex Wallar

December 24, 2014

1 Temporal Probabilistic Roadmap

Construction A temporal probabilistic roadmap (tPRM) is similar to other probabilistic roadmaps but instead of just representing the configuration space of the robot, a tPRM also incorporates the time at which the robot would be at that configuration. For instance, in a two dimensional setting, the nodes of the tPRM would be of the form (x, y, t) where t is the time at which the robot would be at (x, y). The addition of time into the tradition approach allows more constraints to be set on the construction of the roadmap that include time. The tPRM is represented by a weighted directed graph, (V, E, W), such that an edge can only exist from n_i to n_j if $n_i^{(t)} < n_j^{(t)}$ along with other constraints set during the construction. The temporal probabilistic roadmap is constructed by sampling random points within the search space and connecting nodes that are feasible for the robot to move between. More formally, an edge exists between n_i and n_j if $F_R(n_i, n_j) = 1$ where the function, $F_R : V \times V \to \{0, 1\}$ represents whether or not it is feasible to move between two vertices in the roadmap for a given robot model, R. Other than the feasibility constraint for an edge between two nodes, there are also constraints that limit the size of edges. We assume that there is a maximum time $T_m > 0$ such that if $(n_i, n_j) \in E$, then $n_j^{(t)} - n_i^{(t)} \leq T_m$. Likewise, we assume that there is a maximum distance that can exist between nodes, $D_m > 0$, such that if $(n_i, n_j) \in E$, then $dist(n_i, n_j) \leq D_m$ where $dist(n_i, n_j)$ is the distance between n_i and n_j in Euclidean space. Algo. 1 shows pseudocode for the construction of the tPRM.

Search In order to get a set of sub-goals that can be used to lead the robot to the goal, a graph search algorithm is used. The current implementation uses Dijkstra's shortest path algorithm because it is provided in the `NetworkX` graph library by default. Given the constructed roadmap, the algorithm is used to find the shortest path in the directed graph from the robot's initial configuration to it's goal configuration.

2 Agent Position Probability Distribution

For a given set of agents, A, in the environment and their associated trajectories, $\{\zeta_a : a \in A\}$, where $\zeta_a : \mathbb{R} \to \mathbb{R}^k$, the probability of an agent being given position is represented between a given time interval is given by the function, $P_a^{(t_0,\, t_m)} : \mathbb{R}^k \to (0, 1)$, where k is the dimension of the search space. $P_a^{(t_0,\, t_m)}$ is defined as,

$$P_a^{(t_0,\, t_m)}(\mathbf{x}^k) = \int_{t_0}^{t_m} \mathcal{N}_k(\zeta_a(t), (t_m - t)^2 \cdot \alpha)(\mathbf{x}^k) dt$$

Algorithm 1 Pseudocode for tPRM construction

$G \leftarrow$ InitDirectedGraph()
$S \leftarrow \{start\}, i \leftarrow 0$
while $i < MaxPoints$ **do**
  $i \leftarrow i + 1$
  $\rho \leftarrow$ RandomChoice(S)
  $\sigma \leftarrow$ Sample(ρ, F_R)
  if WithinSearchSpace(σ) **then**
    $S \leftarrow S \cup \{\sigma\}$
  for all $s \in S$ **do**
    if $dist(s, \sigma) \leq D_m \wedge |\sigma^{(t)} - s^{(t)}| \leq T_m$ **then**
      if $s^{(t)} < \sigma^{(t)} \wedge F_R(s, \sigma) = 1$ **then**
        AddEdge($G, s, \sigma, P_A^{(s^{(t)},\, \sigma^{(t)})}(s, \sigma)$)
      if $s^{(t)} > \sigma^{(t)} \wedge F_R(\sigma, s) = 1$ **then**
        AddEdge($G, \sigma, s, P_A^{(\sigma^{(t)},\, \mathbf{s}^{(t)})}(\sigma, s)$)
return G

Where $\mathbf{x}^k$ represents the k-dimensional input vector, α is a scaling constant, and $\mathcal{N}_k(\mu, \sigma^2)$ is a k-dimensional Gaussian distribution centered at μ with a variance σ^2. For all of the agents, the probability density function becomes

$$P_A^{(t_0,\, t_m)}(\mathbf{x}^k) = \frac{\sum_{a \in A} P_a^{(t_0,\, t_m)}(\mathbf{x}^k)}{|A|}$$

3 Edge Weights

For two nodes, n_i and n_j, in the temporal probabilistic roadmap (tPRM), where $n_i^{(t)} < n_j^{(t)}$, where $n^{(t)}$ represents the time at which robot would be at node n, the weight of the edge between those two nodes is,

$$w_N(n_i, n_j) = \gamma \cdot \frac{\sum_{i=0}^{N} (P_A^{(n_i^{(t)},\, n_j^{(t)})} \circ E)(n_i^{(t)} + i \cdot \frac{n_j^{(t)} - n_i^{(t)}}{N})}{N}$$

Where $E(t)$ is a function, $E : \mathbb{R} \to \mathbb{R}^k$, that represents the position along the edge between n_i and n_j given a time, $t \in (n_i^{(t)},\ n_j^{(t)}]$ and k is the dimension of the search space. In the equation, N is the number of samples along the edge, $\circ$ represents the function composition, and γ is a scaling constant. As the number of samples tends towards infinity (i.e. $\lim_{N\to\infty} w_N(n_i, n_j)$), the weight becomes,

$$\begin{aligned} w(n_i, n_j) &= \lim_{N \to \infty} w_N(n_i, n_j) \\ &= \gamma \cdot \int_{n_i^{(t)}}^{n_j^{(t)}} (P_A^{(n_i^{(t)},\, n_j^{(t)})} \circ E)(t) dt \end{aligned}$$