Accelerating Motion Planning for Learned Mobile Manipulation Tasks using Task-Guided Gibbs Sampling

Chris Bowen and Ron Alterovitz

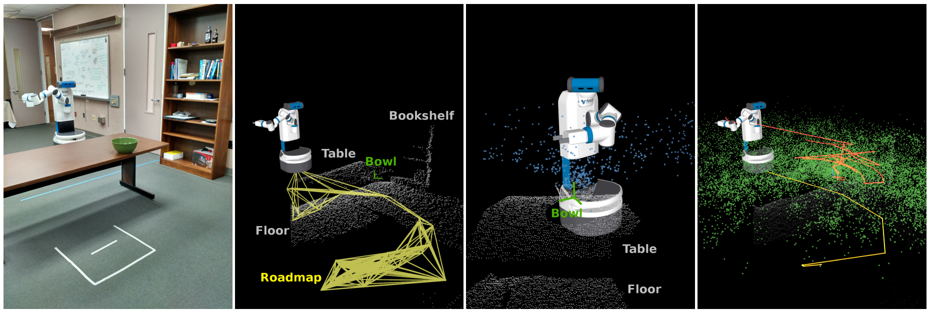
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175, USA
{cbbowen,ron}@cs.unc.edu

Abstract. We present Task-Guided Gibbs Sampling (TGGS), an approach to accelerating motion planning for mobile manipulation tasks learned from demonstrations. This method guides sampling toward configurations most likely to be useful for successful task execution while avoiding manual heuristics and preserving asymptotic optimality of the motion planner. We leverage the learned task model, which is already used by the motion planner to evaluate path quality, to also guide sampling, yielding plans with high rates of success faster than unbiased or goal-biased sampling. This is accomplished by tightly integrating sampling with a hybrid motion planner that builds separate base and arm roadmaps using Gibbs sampling. Such an approach allows the sampled arm configurations to depend on the reachable base configurations and vice-versa. We evaluate our method on two household tasks using the Fetch robot and achieve a 90% success rate on each, greatly improving upon a planner relying on unbiased sampling or either of two goal-biased planners when using the same cost metric.

1 Introduction

For a mobile manipulator performing a task, the problems of navigation and manipulation are inherently coupled. On the one hand, the manipulator may need to be repositioned to navigate through a narrow passage (e.g., moving between furniture in Fig. 1) while adhering to task constraints (e.g., levelness of the grasped pitcher to avoid spilling). On the other, the robot may need to reposition itself to enable the manipulator to accomplish a given task (e.g., pouring liquid from the pitcher into a bowl). This coupling presents multiple challenges, mostly stemming from the associated increase in dimensionality. While a traditional robot arm might have only six degrees of freedom, a mobile manipulator like the Fetch robot [32] might have eleven or more.

Sampling-based motion planners have previously been applied to the execution of tasks based on a learned model. We introduce Task-Guided Gibbs Sampling (TGGS) to accelerate such a motion planner in the context of mobile manipulation tasks by utilizing the same learned task model. TGGS enables the motion planner to more quickly sample configurations that are likely to be important for task success without relying on manual heuristics, thereby producing successful plans faster than sampling methods that do not explicitly consider the task and environment together, like goal-biasing or obstacle-biasing.



(a) (b) (c) (d)

Fig. 1. (a) The Fetch robot's task is to pour liquid from the grasped pitcher into the green bowl on the table. **(b)** Roadmap of base motions biased towards those most likely to be useful (yellow) while avoiding the obstacle point cloud (white). **(c)** End-effector positions for sampled arm configurations biased towards those most likely to be useful (blue). **(d)** End-effector positions for configurations in the hybrid roadmap (green) with the base (yellow) and end-effector (red) motion of the final plan.

Our method depends on a model of the task learned from demonstrations performed in varying environments. This same model, which is used by the motion planner to generate plans that successfully perform the task, also informs the sampling strategy. Specifically, the method we propose combines local optimization with a Markov Chain Monte Carlo (MCMC) approach to effectively project the task space distribution learned from the demonstrations into the robot's configuration space. This sampling strategy is applicable to both the bounded subspace of the manipulator as well as the unbounded subspace of the mobile base. While the task model we consider is based on prior work [6] and the Fetch robot is used in the experiments, we believe TGGS could also be adapted to a number of other models and robots.

Like many low-cost mobile manipulators, the Fetch robot's mobile base uses differential drive, which presents the challenge of nonholonomy. Fast motion planning for nonholonomic robots with 10 or more degrees of freedom remains computationally difficult, especially when asymptotic optimality is desirable, as when soft task constraints are present. Fortunately, the configuration spaces of many mobile manipulators, including the Fetch robot, have a large subspace that *is* holonomic: that of the manipulator arm. TGGS leverages this property by partitioning the sampling across holonomic and nonholonomic degrees of freedom, which accelerates sampling-based motion planners for learned mobile manipulation tasks while preserving their asymptotic optimality guarantees.

We validate our new approach analytically and then empirically using the Fetch robot to perform two household tasks in randomly-generated scenarios.

2 Related Work

Motion planning for mobile manipulation is challenging because of the high number of degrees of freedom. Various methods that lack completeness address this problem, including potential fields [30, 12] and local optimization [20]. However, completeness guarantees are crucial in more complex environments and higher-dimensional spaces. Probabilistically complete sampling-based motion planners are effective in such spaces [25, 1] and can incorporate a class of task constraints [28, 17], even at reactive rates using elastic roadmaps [34].

Only some sampling-based motion planners consider plan cost, and asymptotic optimality guarantees become important when planning for tasks where feasibility is no longer sufficient for success [31]. Furthermore, in high-dimensional spaces, the convergence of asymptotically optimal motion planners may still be impractically slow. Even when only completeness is required, biased sampling may improve performance [2], particularly in narrow passages [24, 7, 29, 14] and other challenging motion planning problems. Similarly, cost-based task constraints [23] may induce narrow passages in cost space that make planning more challenging, and biased sampling is again effective [4, 16, 11, 5].

In this paper, we introduce a sampling strategy designed specifically for constructing roadmaps for a given environment and learned model of a mobile manipulation task that retains asymptotic optimality. Although we apply this approach to one task model [6] here, it could be readily adapted to planning tasks represented by other generative models that may be learned from demonstrations [8, 13].

Additionally, our approach relies on decomposing the configuration space of the robot. Related approaches have been considered in the contexts of hierarchical planning [10, 26] and hybrid planning [27], but without asymptotic optimality guarantees. We also note that high-level methods have considered this problem in a more traditional task planning framework [9, 33, 19], but these approaches depend on methods that can plan low-level actions, which is precisely the domain of our method.

3 Problem Definition

We consider a mobile manipulator that consists of a holonomic arm with d_{arm} degrees of freedom and a mobile base (which may be holonomic or nonholonomic) with d_{base} degrees of freedom. Given a set of arm configurations $Q_{\text{arm}} \subset \mathbb{R}^{d_{\text{arm}}}$ and a set of base configurations $Q_{\text{base}} \subseteq \mathbb{R}^{d_{\text{base}}}$, let $F_{\text{arm}} \subseteq Q_{\text{arm}}$ denote those arm configurations which are potentially free for some base configuration (i.e., those that are not in self-collision). Similarly, let $F_{\text{base}} \subseteq Q_{\text{base}}$ denote those base configurations which are potentially free for some arm configuration (i.e., those for which the base itself does not collide with the environment). Finally, let $Q = Q_{\text{arm}} \times Q_{\text{base}} \subset \mathbb{R}^{d_{\text{arm}}+d_{\text{base}}}$ denote the combined configuration space and $F \subseteq F_{\text{arm}} \times F_{\text{base}}$ denote the free configuration space.

In addition to the above sets, we assume as input to the method an initial configuration $\mathbf{q}_0 \in F$ and a task model. The task model depends on a feature function $f_{\mathbf{a}}(\mathbf{q}) \in Y \subseteq \mathbb{R}^b$ where $\mathbf{a}$ is an environment description that encodes the positions of salient objects (e.g. the bowl in Fig. 1). We call Y the feature space, and the task model consists of a finite sequence of feature space multivariate Gaussian distributions $\{\mu_1, \Sigma_1, \mu_2, \Sigma_2, \dots, \mu_T, \Sigma_T\}$. One can think of these feature space distributions as defining distributions over configurations when pulled back along f, conditional on a given environment described by $\mathbf{a}$. Task models of this form can be estimated from expert demonstrations for a variety of useful tasks [6, 8, 13].

The output of the motion planner is a feasible configuration space path π comprising a sequence of local plans which maximizes similarity to this task model. In this paper, we use a specific definition of similarity [6] which we summarize in Sec. 4.2.

4 Motion Planning using the Learned Model

The problem of planning motions for mobile manipulation tasks presents a number of challenges but also a degree of structure. Notably, any mobile base that uses differential drive (e.g. the Fetch robot) is nonholonomic and thus so is the robot as a whole. This precludes a large class of motion planners [21], including probabilistic roadmaps with linear edges. However, the subspace of configuration space that represents only the arm *is* holonomic.

In Sec. 4.1, we first describe and analyze a hybrid roadmap that leverages this inherent structure by using different roadmap constructions for different subspaces of the problem as appropriate. Next, in Sec. 4.2, we apply a learned task model to this hybrid roadmap. Then, in Sec. 4.3, we describe how the nature of this construction lends itself to efficient sampling of high-quality configurations in both roadmaps using Gibbs sampling.

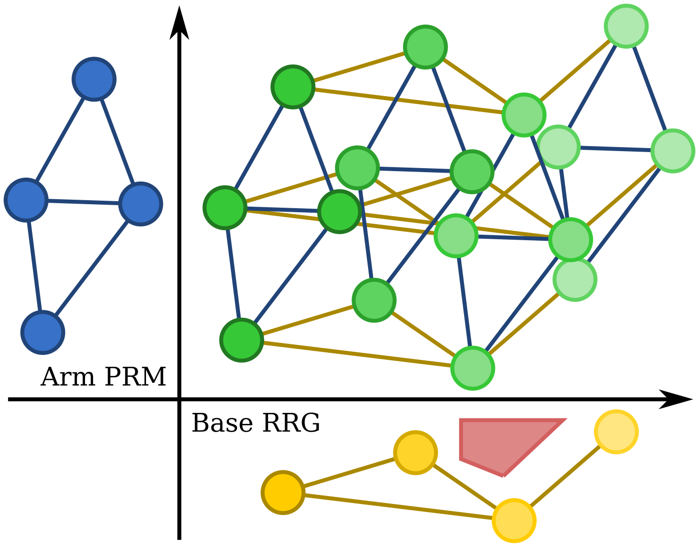


Fig. 2. Hybrid roadmap (green) formed by the Cartesian product of an arm PRM (blue) and a base RRG (yellow) while avoiding obstacles (red).

4.1 Mobile Manipulation Hybrid Roadmap

Mobile manipulators are generally characterized by low-dimensional base motion, usually $\mathrm{SE}(2)$, combined with a high-dimensional manipulator, $\mathbb{R}^5(\mathrm{S}^1)^3$ in the case of the Fetch robot. The combined configuration space is often 11-dimensional or greater. Additionally, we are not interested in merely feasible plans but ones which also have low costs, ideally approaching the optimal solution in the limit.

Asymptotically-optimal variants of rapidly-exploring random graphs, e.g RRG* [22], are applicable to such motion planning problems. However, in high-dimensional spaces, these planners can exhibit very slow convergence to optimality. On the other hand, less general methods like probabilistic roadmaps (e.g. PRM*) may exhibit faster convergence for multiple queries but often rely on additional structure, like an optimal local planner. Furthermore, applying PRM* to unbounded spaces, like $\mathrm{SE}(2)$, necessitates additional heuristics because new configurations will be added from the entire space unlike RRG* which only extends to configurations near those already explored.

We show that these planners can be combined to plan for mobile manipulation tasks in such a way that their relative strengths are exploited in the subspaces where they are most applicable. Specifically, we propose to build an RRG* of pure base motions and a separate PRM* of pure manipulator motions. We then search in the Cartesian product of the resulting graphs to find asymptotically-optimal motion plans. Additionally, this approach allows for eager collision checking on the base during roadmap construction but lazily checking for arm collisions during the shortest path search.

Let $C_{\text{base}}(e_{\text{base}}, \mathbf{q}_{\text{arm}})$ denote the cost of traversing edge e_{base} in the base roadmap with arm configuration $\mathbf{q}_{\text{arm}}$ and $C_{\text{arm}}(\mathbf{q}_{\text{base}}, e_{\text{arm}})$ denote the cost of traversing edge e_{arm} in the arm roadmap at base configuration $\mathbf{q}_{\text{base}}$. Every edge in the Cartesian product graph is composed of a vertex from one graph and an edge from the other. This implies that the arm and base are never actuated simultaneously. Let $C(e)$ be either C_{base} or C_{arm} as applicable. We assume C_{base} and C_{arm} are non-negative and Lipschitz continuous. We extend this cost to a path π in the obvious way:

$$C(\pi) = \sum_{e \in \pi} C(e).$$

Algorithm 1 BUILDHYBRIDROADMAP($\mathbf{q}_0$, n_{base}, n_{arm})

```
Input: Initial configuration $\mathbf{q}_0$, number of samples $n$
  $G_{\text{base}} \leftarrow$ BUILDBASEROADMAP($F_{\text{base}}$, $\mathbf{q}_0$, $n_{\text{base}}$)
  $G_{\text{arm}} \leftarrow$ BUILDARMROADMAP($F_{\text{arm}}$, $\mathbf{q}_0$, $n_{\text{arm}}$)
  return $G_{\text{base}} \times G_{\text{arm}}$ // graph Cartesian product
```

Theorem 1. *If both* BUILDBASEROADMAP *and* BUILDARMROADMAP *are asymptotically-optimal with respect to C_{base} and C_{arm} for all $\mathbf{q}_{\text{arm}}$ and $\mathbf{q}_{\text{base}}$ respectively, then* BUILDHYBRIDROADMAP *(see Alg. 1) is asymptotically-optimal with respect to C with the restriction that only the base or arm are actuated at any given time.*

Proof. For any $\varepsilon > 0$, consider a feasible path π such that $C(\pi) < C^* + \frac{\varepsilon}{2}$ where C^* is the infimum of costs among feasible paths. Such a path must exist by definition. Let π_{base}

denote the set of edges in π which correspond to base motions and π_{arm} those which correspond to arm motions. Observe that, by associativity, $C(\pi) = C(\pi_{\text{base}}) + C(\pi_{\text{arm}})$.

We next note that because the base and arm do not actuate simultaneously, π_{base} can be divided up into no more than $|\pi|$ sub-paths, each with $\mathbf{q}_{\text{arm}}$ constant in each sub-path. Let K_{base} denote the Lipschitz constant of C_{base}. By assumption, for all $\delta > 0$ there exists m_{arm} such that BUILDARMROADMAP(F_{arm}, $\mathbf{q}_0$, m_{arm}) includes a reachable configuration $\mathbf{q}'_{i,\text{arm}}$ that is $\frac{\varepsilon}{8K_{\text{base}}|\pi|}$-close to a given $\mathbf{q}_{i,\text{arm}}$ with probability $1 - \frac{\delta}{4|\pi|}$. Thus, with probability at least $(1 - \frac{\delta}{4|\pi|})^{|\pi|} \geq 1 - \frac{\delta}{4}$ this holds for each such $\mathbf{q}_{\text{arm}}$.

Now, consider one of these sub-paths $\pi_{i,\text{base}}$ with shared $\mathbf{q}_{i,\text{arm}}$. Again by assumption, there exists n_{base} such that with probability at least $1 - \frac{\delta}{4|\pi|}$ there is a path $\pi'_{i,\text{base}}$ in BUILDBASEROADMAP(F_{base}, $\mathbf{q}_0$, n_{base}) with $C(\pi'_{i,\text{base}}, \mathbf{q}_{i,\text{arm}}) < C(\pi_{i,\text{base}}, \mathbf{q}_{i,\text{arm}}) + \frac{\varepsilon}{8|\pi|}$. By Lipschitz continuity of C_{base}, we have $C(\pi'_{i,\text{base}}, \mathbf{q}'_{i,\text{arm}}) < C(\pi'_{i,\text{base}}, \mathbf{q}_{i,\text{arm}}) + \frac{\varepsilon}{8|\pi|}$. Together, these sub-paths form π'_{base} with $C(\pi'_{\text{base}}) < C(\pi_{\text{base}}) + \frac{\varepsilon}{4}$ with probability at least $(1 - \frac{\delta}{4})^2 \geq 1 - \frac{\delta}{2}$.

The above argument applies similarly to π_{arm}, with π'_{arm}, m_{base}, and n_{arm}. So, for $n = \max(m_{\text{base}}, n_{\text{base}}, m_{\text{arm}}, n_{\text{arm}})$, with probability at least $(1 - \frac{\delta}{2})^2 \geq 1 - \delta$ there exists a path π' in BUILDHYBRIDROADMAP($\mathbf{q}_0, n, n$) such that $C(\pi') = C(\pi'_{\text{base}}) + C(\pi'_{\text{arm}}) < C(\pi) + \frac{\varepsilon}{2} < C^* + \varepsilon$. ■

The condition that BUILDBASEROADMAP be asymptotically optimal with respect to C_{base} for all $\mathbf{q}_{\text{arm}}$ and the corresponding condition for BUILDARMROADMAP may initially seem too strong. However, these conditions are satisfied with many useful metrics for mobile manipulators. In the experiments we performed, we let C_{base} be a scalar multiple of execution time, which is independent of $\mathbf{q}_{\text{arm}}$ and thus the assumption holds using an RRG* with a closed-form local planner to optimally satisfy the non-holonomic constraints [3]. Because the manipulator arm is holonomic, asymptotic optimality of PRM* holds for all non-singular C_{arm} by local metric equivalence, so the dependence on $\mathbf{q}_{\text{base}}$ is similarly unproblematic. More generally, however, asymptotic optimality of the method relies on that of the underlying methods, which may be chosen based on the requirements imposed based on the cost metrics.

The Cartesian product graph of these roadmaps need not be explicitly constructed. Rather an implicit representation can be lazily traversed by simply traversing the constituent graphs. Note also that because F may be a subset of $F_{\text{base}} \times F_{\text{arm}}$, collision checking must still be performed on the edges of the hybrid roadmap. In our implementation, this is done lazily during the graph search described in the following subsection.

4.2 Task Roadmap

To extend the hybrid roadmap described above to motion planning for a task, we use another Cartesian product to construct a spatio-temporal roadmap [5] as described in Alg. 2. This approach extends an asymptotically-optimal planner to maximize similarity to a learned task model comprising a finite sequence of *time steps* by permitting the cost metric to depend on task progress in the form of the current time step.

The construction of the spatio-temporal roadmap $G_{\text{st}} = G_{\text{s}} \times G_{\text{t}}$ is necessitated by our choice for C_{arm}, which depends on the current time step t. The specific choice of

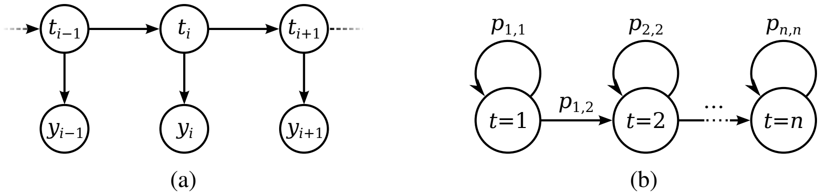


Fig. 3. (a) Hidden Markov model defining a distribution of sequences of feature space vectors with the hidden state comprising discrete time steps. **(b)** Restricted structure of the hidden Markov model.

```
Algorithm 2 MotionPlanForTask($\mathbf{q}_0$, $n_{\text{base}}$, $n_{\text{arm}}$, $T$)
Input: Initial configuration $\mathbf{q}_0$, numbers of samples $n_{\text{base}}$ and $n_{\text{arm}}$, number of time steps $T$
  $G_{\text{s}} \leftarrow$ BuildHybridRoadmap($\mathbf{q}_0$, $n_{\text{base}}$, $n_{\text{arm}}$)
  $G_{\text{t}} \leftarrow$ TaskStructure($T$) // Graph of time steps: linear graph with T vertices
  $v_0 \leftarrow$ NearestVertex($G_{\text{s}}$, $\mathbf{q}_0$)
  return ShortestPath($G_{\text{s}} \times G_{\text{t}}$, $(v_0, 1)$, $V(G_{\text{s}}) \times \{T\}$)
```

C_{arm} depends on the learned task model. In this paper, we consider a model defined by a time-homogeneous hidden Markov model (HMM) with a restricted structure and discrete state space of time steps that capture task progress [6]. We denote the time step by t and number them sequentially $1, 2, \ldots, T$ (see Fig. 3).

Under this model, an observation is a feature space vector $\mathbf{y} \in Y$. Recall that $\mathbf{y} = f_{\mathbf{a}}(\mathbf{q})$ lifts a configuration $\mathbf{q}$ into this feature space conditional on an environ description $\mathbf{a}$ that encodes the positions of salient objects in the environment. This transformation enables the model to capture relationships between the environment and configurations, e.g. that the pitcher must be placed over the bowl before pouring in Fig. 1. The specific choices of $f_{\mathbf{a}}$ used for the experiments are discussed in their respective subsections of Sec. 5, but the method requires only that it be differentiable. We model the observation distribution by a multivariate Gaussian distribution with a per-time step mean μ_t and covariance matrix Σ_t.

$$f_{\mathbf{a}_i}(\mathbf{q}_i) = \mathbf{y}_i \sim \mathcal{N}(\mu_{t_i}, \Sigma_{t_i})$$

Because the model was estimated from successful demonstrations, we expect a plan which is similar to the task model to be likely to accomplish the task. Following this intuition, we define edge costs C_{arm} that when minimized, maximizes the probability density function of the learned model. To do so, we first assign a cost to individual configurations for each time step as follows:

$$\begin{aligned} c(\mathbf{q},t) &= -\log p(f(\mathbf{q}) \,|\, \mu_t, \Sigma_t) \\ &= \frac{1}{2}\left(\|f(\mathbf{q}) - \mu_t\|_{\Sigma_t^{-1}} + \log\det 2\pi\Sigma_t \right). \end{aligned}$$

Then, we let C_{arm} be simply the path integral over this cost $\int c(\mathbf{q}(s), t)\, \mathrm{d}s$. This is the continuous analog of the summation that arises from the Markov assumption made by the task model. Specifically, rather than the following distribution over discrete paths:

$$p(\mathbf{q}_0, \ldots, \mathbf{q}_n \,|\, t) \propto \prod_{i=0}^{n} p(\mathbf{q}_i \,|\, t) = e^{-\sum_{i=0}^{n} c(\mathbf{q}_i, t)},$$

we instead define a distribution using a functional over configuration space curves:

$$p(\mathbf{q} \,|\, t) \propto e^{-\int c(\mathbf{q}(s), t)\,\mathrm{d}s}.$$

Due to the structure of the spatio-temporal roadmap G_{st}, the cost of the same local plan associated with an edge in the hyrbid roadmap is often needed under multiple metrics, one for each time step. This structure allows the computation to be accelerated by a pre-computation as follows:

$$\begin{aligned}
&\|f(\mathbf{q}) - \mu_t\|_{\Sigma_t^{-1}} \\
&= \mathrm{Tr}\left((f(\mathbf{q}) - \mu_t)^{\mathrm{T}} \mathrm{L}_t \mathrm{L}_t^{\mathrm{T}} (f(\mathbf{q}) - \mu_t)\right) \\
&= \mathrm{Tr}\left(\mathrm{L}_t^{\mathrm{T}} (f(\mathbf{q}) - \mu_t)(f(\mathbf{q}) - \mu_t)^{\mathrm{T}} \mathrm{L}_t\right) \\
&= \mathrm{Tr}\left(\mathrm{L}_t^{\mathrm{T}} \left(f(\mathbf{q}) f(\mathbf{q})^{\mathrm{T}} - 2\mu_t f(\mathbf{q})^{\mathrm{T}} + \mu_t \mu_t^{\mathrm{T}}\right) \mathrm{L}_t\right)
\end{aligned}$$

where $\mathrm{L}_t \mathrm{L}_t^{\mathrm{T}} = \Sigma_t^{-1}$ is the Cholesky decomposition of Σ_t^{-1}. So,

$$\begin{gathered}
\textstyle\int_0^S c(\mathbf{q}(s))\,\mathrm{d}s = \\
S \log\det 2\pi\Sigma_t + \textstyle\int_0^S \|f(\mathbf{q}) - \mu_t\|_{\Sigma^{-1}}\,\mathrm{d}s = \\
S \log\det 2\pi\Sigma_t + \mathrm{Tr}\left(\mathrm{L}_t^{\mathrm{T}} \left(\mathrm{A} - 2\mathbf{b}\mu_t^{\mathrm{T}} + S\mu_t\mu_t^{\mathrm{T}}\right) \mathrm{L}_t\right) \\
\text{where}\, \mathrm{A} = \textstyle\int_0^S f(\mathbf{q}) f(\mathbf{q})^{\mathrm{T}}\,\mathrm{d}s \;\text{and}\; \mathbf{b} = \textstyle\int_0^S f(\mathbf{q})\,\mathrm{d}s\,.
\end{gathered} \tag{1}$$

Note that A and **b** are independent of t and can thus be precomputed numerically for a local path. Once this is done, edge costs may be computed using only simple matrix operations, making them very efficient to compute.

Under these definitions of C_{arm} and C_{base}, the cost of a plan can be rewritten as

$$C(\pi) = \int_0^S g(\pi(s), t)\,\mathrm{d}s \tag{2}$$

where $g(\mathbf{q}, t) = \kappa$ if the base is moving at time s and $g(\mathbf{q}, t) = c(\mathbf{q}, t)$ if the arm is moving. κ is the weight with which base movement is penalized.

4.3 Task-Guided Gibbs Sampling

While the method described in the previous section is asymptotically optimal, it converges very slowly in high-dimensional configuration spaces. To improve convergence, we employ biased sampling. Motivated by importance sampling, our goal is to effectively project the task model from feature space into configuration space. To do so explicitly would require strong assumptions about the form of f. Rather, we observe that we only need to *sample* from the projected distribution. Furthermore, these samples do

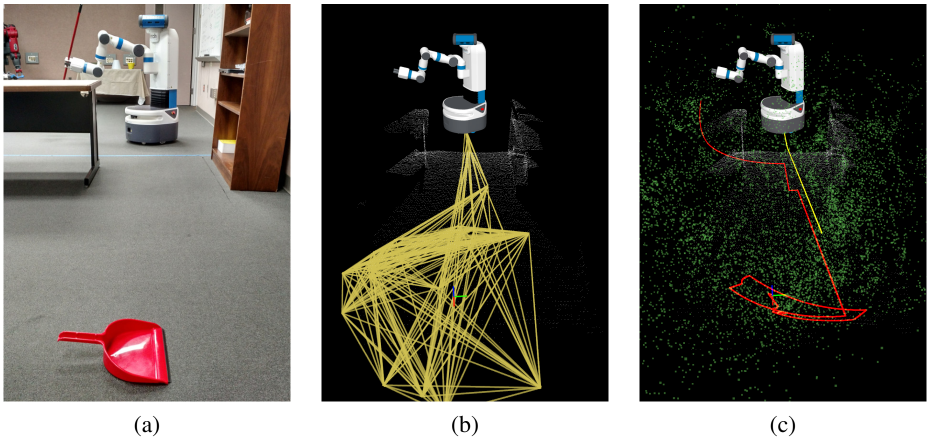


Fig. 4. (a) Sweeping task using the Fetch robot. **(b)** Biased RRG* of base motions (yellow) around the obstacle point cloud (white). **(c)** Hybrid roadmap (green) with the base (yellow) and end-effector (red) motion of the final plan.

not need to be independent of each other, only approximate the desired distribution in the limit. Following this insight, we propose to use a Markov Chain Monte Carlo sampling strategy. Specifically, we employ a variant of Gibbs sampling tightly integrated with the hybrid nature of the roadmap.

Gibbs sampling is simply an approach to sampling from the joint distribution of two dependent variables by fixing one and sampling the other conditional on that value, then repeating that process with the variables' roles reversed. Following this approach, we sample individual base and arm configurations separately while the distributions as a whole are interdependent. To sample an arm configuration, we first select a base configuration $\hat{\mathbf{q}}_{\text{base}}$ at random from the base roadmap G_{base}. We then sample a feature vector $\hat{\mathbf{y}}$ at random from the task model. Finally, we numerically solve for the arm configuration that results in a feature vector most similar to $\hat{\mathbf{y}}$ given $\hat{\mathbf{q}}_{\text{base}}$ by solving the following nonlinear least-squares minimization problem:

$$\arg\min_{\mathbf{q}_{\text{arm}}} \|f_{\hat{\mathbf{a}}}(\hat{\mathbf{q}}_{\text{base}}, \mathbf{q}_{\text{arm}}) - \hat{\mathbf{y}}\|_{\Sigma^{-1}}$$

Sampling a base configuration proceeds similarly and different numbers of base and arm samples can be accomodated by simply skipping some extension steps (see Alg. 3). This entire procedure replaces Alg. 1. Because the configurations in G_{base} are also in F_{base}, we are effectively biasing the distribution of arm configurations towards those which achieve the task with the base in feasible locations. Similarly, we are sampling base configurations which are useful when the arm is not in a self-colliding state.

The sampling approach described above can be readily adapted to sampling from a sequence of multivariate normal distributions as found in the task model. We first select a number of samples n to compute in a batch to ensure the samples are spread evenly across the distributions. Rather than sampling a fixed number from each of the discrete distributions, we gradually alter the distribution from which $\hat{\mathbf{y}}$ is sampled at each iteration. Specifically, we linearly interpolate between adjacent distributions in the model, which can be computed by a simple procedure. For each $i < n$, we consider continuous $t = i(T-1)/n$. Because normal distributions are additive, we can interpolate between them in multiple equivalent ways. Perhaps the most intuitive way is by the explicit parameters:

$$\begin{aligned} \mu_{(i)} &= (t - \lfloor t \rfloor) \cdot \mu_{\lfloor t \rfloor} + (\lceil t \rceil - t) \cdot \mu_{\lceil t \rceil} \\ \Sigma_{(i)} &= (t - \lfloor t \rfloor) \cdot \Sigma_{\lfloor t \rfloor} + (\lceil t \rceil - t) \cdot \Sigma_{\lceil t \rceil} \end{aligned}$$

However, sampling using Σ directly requires that we compute the Cholesky decomposition which takes $\Theta(b^3)$ time. This is reasonable to perform for each distribution in the model, but unnecessary for each sample.

Instead, we observe that, similar to the previous section, we are only interested in sampling from the distribution, not explicitly constructing it. So it suffices to sample $\mathbf{y}_- \sim \mathcal{N}(\mu_{\lfloor t \rfloor}, \Sigma_{\lfloor t \rfloor})$ and $\mathbf{y}_+ \sim \mathcal{N}(\mu_{\lceil t \rceil}, \Sigma_{\lceil t \rceil})$, and linearly interpolate between these samples, yielding $\tilde{\mathbf{y}}_i = (t - \lfloor t \rfloor) \cdot \mathbf{y}_- + (\lceil t \rceil - t) \cdot \mathbf{y}_+$. This follows from the additive property of normal distributions.

Algorithm 3 BUILDHYBRIDROADMAPGIBBS($\mathbf{q}_0$, n_{base}, n_{arm})

```
Input: Initial configuration $\mathbf{q}_0$, number of samples $n$
  $G_{\text{base}} \leftarrow \{\mathbf{q}_0\}$
  $G_{\text{arm}} \leftarrow \{\mathbf{q}_0\}$
  $n \leftarrow \max(n_{\text{base}}, n_{\text{arm}})$
  for $i \leftarrow 1$ to $n$ do
    $\hat{\mathbf{q}}_{\text{base}} \leftarrow$ randomly from $G_{\text{base}}$ // sample arm using fixed base
    $\hat{\mathbf{y}}_1 \leftarrow$ SAMPLE($\mathcal{N}(\mu_{(i)}, \Sigma_{(i)})$)
    $\mathbf{q}_{\text{arm}} \leftarrow \arg\min_{\mathbf{q}_{\text{arm}}} \|f_{\hat{\mathbf{a}}}(\hat{\mathbf{q}}_{\text{base}}, \mathbf{q}_{\text{arm}}) - \hat{\mathbf{y}}_1\|_{\Sigma_i^{-1}}$
    if $(i \cdot n_{\text{arm}}$ rem $n) < n_{\text{arm}}$ then
      EXTEND($G_{\text{arm}}$, $\mathbf{q}_{\text{arm}}$)
    end if
    $\hat{\mathbf{q}}_{\text{arm}} \leftarrow$ randomly from $G_{\text{arm}}$ // sample base using fixed arm
    $\hat{\mathbf{y}}_2 \leftarrow$ SAMPLE($\mathcal{N}(\mu_{(i)}, \Sigma_{(i)})$)
    $\mathbf{q}_{\text{base}} \leftarrow \arg\min_{\mathbf{q}_{\text{base}}} \|f_{\hat{\mathbf{a}}}(\mathbf{q}_{\text{base}}, \hat{\mathbf{q}}_{\text{arm}}) - \hat{\mathbf{y}}_2\|_{\Sigma_i^{-1}}$
    if $(i \cdot n_{\text{base}}$ rem $n) < n_{\text{base}}$ then
      EXTEND($G_{\text{base}}$, $\mathbf{q}_{\text{base}}$)
    end if
  end for
  return $G_{\text{base}} \times G_{\text{arm}}$
```

Examples of the resulting base roadmap and hybrid roadmaps are shown for a sweeping task in Fig. 4 and for a liquid pouring task in Fig. 1d.

4.4 Guiding Manifolds

In prior work [5], numeric optimization was used to seed the roadmap with local minima of each distribution to find high quality paths quickly in unconstrained space. There were generally finitely many such minima because the feature space was well-behaved and higher-dimensional than the configuration space. This is not always the case for mobile manipulators, where the configuration space may be larger than the feature space or the Jacobian of f may be singular. One effect of this is that the local minima no longer form a so-called *guiding path* but a *guiding manifold* for each distribution. Generally, this manifold is where the configuration exactly satisfies the learned means, that is $f_{\hat{\mathbf{a}}}(\mathbf{q}) = \mu$.

We find it both intuitive and empirically effective to sample on these guiding manifolds to find good paths quickly in addition to the random sampling that ensures asymptotic optimality. Seeding in this way also improves the initial distribution of Gibbs samples, replacing the usual burn-in step for MCMC samplers. To accomplish this, we locally-optimize randomly sampled configurations similarly to Sec. 4.3.

$$\arg\min_{\mathbf{q}_{\text{arm}}} \| f_{\hat{\mathbf{a}}}(\hat{\mathbf{q}}_{\text{base}}, \mathbf{q}_{\text{arm}}) - \mu_t \|_{\Sigma_t^{-1}}$$

While this approach does not provide guarantees about the distribution of the resulting samples, no such guarantee is required because these are only used to seed the roadmap. We note that prior work has considered similar problems in greater depth [18, 15].

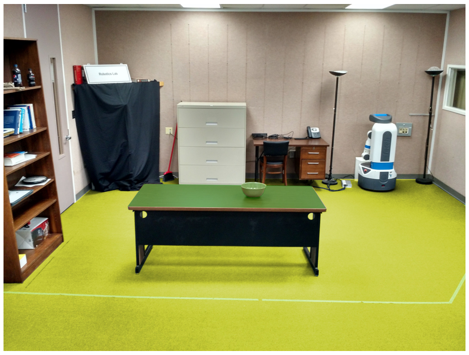


Fig. 5. Household environment used for both tasks, with the floor highlighted in yellow. The table surface for the liquid pouring task is shown in green.

5 Results

We evaluated the method on two household tasks: a liquid pouring task and a sweeping task. In both experiments, the salient objects and obstacles were sensed with the integrated Primesense RGBD. To demonstrate the real-world applicability of the method, we perform collision detection directly against the point cloud. The learned models for both tasks each had 12 time steps. All timings were performed on an Intel Xeon E5-1680 CPU with 8 cores running at 3.40GHz and 64GB of RAM.

Example executions of both tasks are shown in the attached video, including a scenario which required the robot to navigate a doorway, part of the environment which was not included in any of the demonstrations.

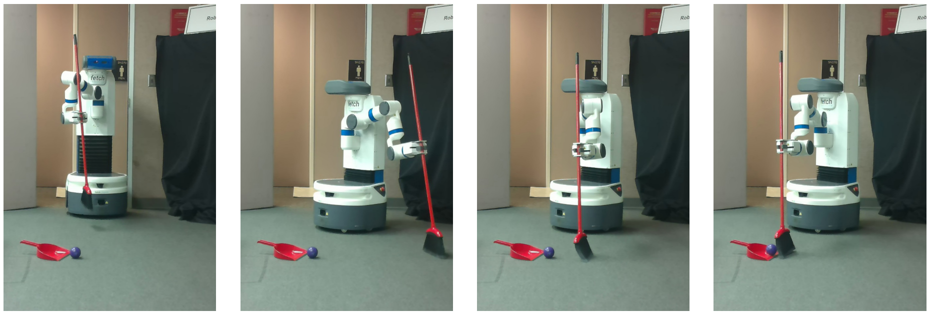


Fig. 6. Sweeping task execution.

5.1 Sweeping Task

In this task, we required the robot to navigate while holding a broom which it then used to sweep the floor toward a dustpan (see Fig. 6). In each of the demonstrations and the subsequent evaluations, we indepdently sampled the initial position of the robot and the dustpan uniformly at random from a 5.5m by 4.8m rectangle on the floor, rejecting those positions in collision with the environment (see Fig. 5).

For the task, the feature function was given by:

$$f_{\mathbf{a}}(\mathbf{q}) = \begin{bmatrix} \mathbf{q}_{5\ldots 11} \\ v(\mathrm{K}_{\mathrm{end}}(\mathbf{q})^{-1}\mathrm{K}_{\mathrm{dustpan}}(\mathbf{a})) \\ v(\mathrm{K}_{\mathrm{dustpan}}(\mathbf{a})^{-1}\mathrm{K}_{\mathrm{gripper}}(\mathbf{q})) \end{bmatrix}$$

where $v(\mathrm{K})$ denotes the translational part of affine transformation K, $\mathrm{K}_{link}(\mathbf{q})$ denotes the forward kinematics of *link* in configuration $\mathbf{q}$, and $\mathrm{K}_{\mathrm{dustpan}}(\mathbf{a})$ denotes the pose of the dustpan in the environment described by $\mathbf{a}$.

The resulting feature space was 13-dimensional. We performed 14 kinesthetic demonstrations of the task via teleoperation. The number of demonstrations was one greater than the dimensionality of the feature space to avoid singular covariance matrices.

We compared our TGGS approach to uniform and goal-biased sampling strategies. Goal-biasing was performed by sampling base positions in a Gaussian distribution around the dustpan and sampling arm configurations from a joint-space Gaussian distribution estimated from the demonstrations. For all sampling strategies, we sampled 250 arm configurations and 50 base configurations in the hybrid roadmap. The full configuration space RRG* used 7500 samples because this produced comparable planning times. In all cases, a spatio-temporal roadmap was constructed from the resulting spatial roadmap and used for planning using the same learned cost metric. Furthermore,

Roadmap	Sampling	Success	Planning Time
Hybrid	**TGGS**	**90%**	**158s**
	TGGS w/o Cost Optimization		174s
	Goal Bias	50%	288s
	Uniform	0%	261s
RRG*	Goal Bias	20%	281s

Table 1. Results for the sweeping task across 10 random scenarios.

because all of the methods considered were asymptotically-optimal, these results indicate how quickly each method converges to successful plans while in the limit, they will all converge to equally good solutions. The cost optimization is that described in (1), which only affects planning time, not the resulting path, and thus not the success rate.

Because the number of samples was fixed for each roadmap type, the differences in timings were largely caused by slower collision checks against the longer edges in the non-task-guided strategies despite the slower sample computation when using TGGS. The most common cause of failure for the goal bias and uniform methods was a lack of samples low enough to sweep but high enough to avoid collision between the block of the broom and the floor, which together form a long narrow passage. In contrast, TGGS sampled densely in this region to produce high-quality sweeping motions.

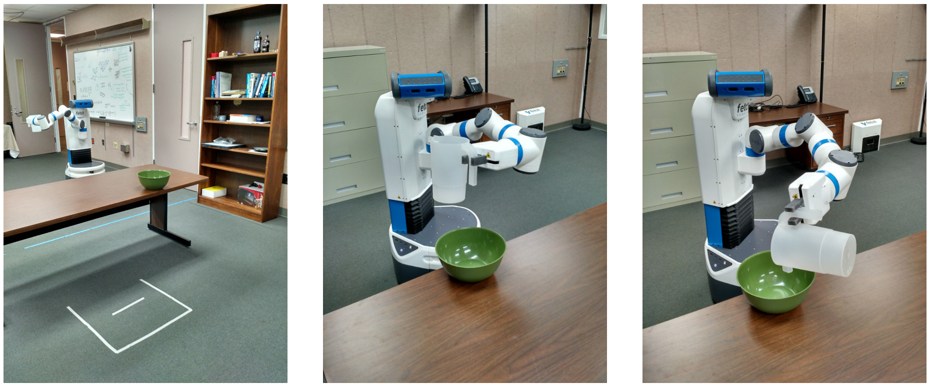


Fig. 7. Liquid pouring task execution.

5.2 Liquid Pouring Task

In this task, we required the robot to navigate while holding a pitcher of water and pour the water into a bowl on a table without spilling (see Fig. 7). In each of the demonstrations and the subsequent evaluations, we independently sampled the initial position of the robot uniformly at random from a 5.5m by 4.8m rectangle, rejecting those positions in collision with the environment (see Fig. 5). Similarly, we sampled the bowl's position uniformly at random from the 1.8m by 0.75m surface of the table.

For this task, the feature function was given by:

$$f_{\mathbf{a}}(\mathbf{q}) = \begin{bmatrix} \mathbf{q}_{6\ldots11} \\ v(\mathrm{K}_{\text{end}}(\mathbf{q})^{-1}\mathrm{K}_{\text{bowl}}(\mathbf{a})) \\ v(\mathrm{K}_{\text{bowl}}(\mathbf{a})^{-1}\mathrm{K}_{\text{gripper}}(\mathbf{q})) \end{bmatrix}$$

where $v(\mathrm{K})$ denotes the translational part of affine transformation K, $\mathrm{K}_{link}(\mathbf{q})$ denotes the forward kinematics of *link* in configuration $\mathbf{q}$, and $\mathrm{K}_{\text{bowl}}(\mathbf{a})$ denotes the pose of the bowl in the environment described by $\mathbf{a}$. We note that this differs from the feature function used for the sweeping task only in the omission of one joint, which was found to be unnecessary for capturing this task. The resulting feature space was 12-dimensional, and we performed 13 kinesthetic demonstrations using the same method as for the sweeping task.

The resulting plans were successful in 90% of the evaluations we performed. The single failure was due to the liquid missing the bowl during the pouring motion, indicating that the method failed to converge to a sufficiently good plan in the alotted time.

Additionally, we measured plan cost (as defined by (2)) with varying planning times using different sampling approaches (see Fig. 8) on a representative scenario from the liquid pouring task. While goal-biasing converged much more quickly than uniform sampling, TGGS was faster still. Furthermore, using a hybrid roadmap improves convergence even when using goal-biasing.

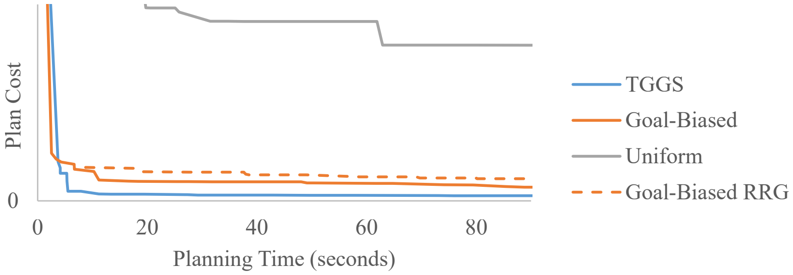


Fig. 8. Plan cost with varying planning times using different sampling approaches on the liquid pouring task. Solid lines indicate variants using hybrid roadmaps, and the dotted line indicates an RRG* was used.

6 Conclusion

We described and analyzed TGGS, an approach to sampling configurations that incorporates information from a learned model of a mobile manipulation task. This sampling strategy was tightly incorporated into a hybrid roadmap construction scheme that decomposes the planning space into that of the manipulator arm and mobile base and uses sampling-based planners most appropriate to each. We demonstrated the efficacy of this approach on two household tasks with the Fetch robot. In future work, we plan to extend TGGS to other task and robot models and explore optimizations with the goal of replanning at reactive rates in response to changes in the environment.

Acknowledgments

We thank Armaan Sethi for his assistance with the demonstrations and experimental evaluation. We used the Rviz visualization tool to generate various figures. This research was supported in part by the U.S. National Science Foundation (NSF) under awards CCF-1533844 and IIS-1149965.

References

1. Akgun, B., Stilman, M.: Sampling heuristics for optimal motion planning in high dimensions. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 2640–2645 (2011)
2. Amato, N.M., Dill, K.A., Song, G.: Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. Journal of Computational Biology 10(3-4), 239–255 (2003)
3. Balkcom, D.J., Mason, M.T.: Time optimal trajectories for bounded velocity differential drive vehicles. The International Journal of Robotics Research 21(3), 199–217 (2002)
4. Berenson, D., Simeon, T., Srinivasa, S.S.: Addressing cost-space chasms in manipulation planning. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 4561–4568 (May 2011)
5. Bowen, C., Alterovitz, R.: Closed-loop global motion planning for reactive execution of learned tasks. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS) (Sep 2014)
6. Bowen, C., Ye, G., Alterovitz, R.: Asymptotically-optimal motion planning for learned tasks using time-dependent cost maps. IEEE Trans. Automation Science and Engineering (2015)
7. Burns, B., Brock, O.: Sampling-based motion planning using predictive models. In: IEEE Int. Conf. Robotics and Automation (ICRA). pp. 3120–3125 (2005)
8. Calinon, S., Li, Z., Alizadeh, T., Tsagarakis, N.G., Caldwell, D.G.: Statistical dynamical systems for skills acquisition in humanoids. In: 12th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids). pp. 323–329. IEEE (2012)
9. Cambon, S., Alami, R., Gravot, F.: A hybrid approach to intricate motion, manipulation and task planning. Int. Journal of Robotics Research (IJRR) 28(1), 104–126 (2009)
10. Candido, S., Kim, Y.T., Hutchinson, S.: An improved hierarchical motion planner for humanoid robots. In: 8th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids). pp. 654–661. IEEE (2008)
11. Devaurs, D., Siméon, T., Cortés, J.: Enhancing the transition-based rrt to deal with complex cost spaces. In: IEEE Int. Conf. Robotics and Automation (ICRA). pp. 4120–4125 (2013)
12. Dietrich, A., Wimbock, T., Albu-Schaffer, A., Hirzinger, G.: Reactive whole-body control: Dynamic mobile manipulation using a large number of actuated degrees of freedom. IEEE Robotics & Automation Magazine 19(2), 20–33 (2012)
13. Eppner, C., Sturm, J., Bennewitz, M., Stachniss, C., Burgard, W.: Imitation learning with generalized task descriptions. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 3968–3974 (May 2009)

14. Gammell, J.D., Srinivasa, S.S., Barfoot, T.D.: Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In: Int. Conf. Robotics and Automation (ICRA). pp. 3067–3074 (2015)
15. Havoutis, I., Ramamoorthy, S.: Motion planning and reactive control on learnt skill manifolds. Int. Journal of Robotics Research (IJRR) 32(9-10), 1120–1150 (2013)
16. Iehl, R., Cortés, J., Siméon, T.: Costmap planning in high dimensional configuration spaces. In: IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (AIM). pp. 166–172 (2012)
17. J. Claassens, Y.D.: Exploiting affordance symmetries for task reproduction planning. In: Humanoids. IEEE (2012)
18. Jaillet, L., Porta, J.M.: Asymptotically-optimal path planning on manifolds. Robotics: Science and Systems VIII p. 145 (2013)
19. Kaelbling, L.P., Lozano-Pérez, T.: Hierarchical task and motion planning in the now. In: IEEE Int. Conf. Robotics and Automation (ICRA). pp. 1470–1477 (2011)
20. Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., Schaal, S.: STOMP: Stochastic trajectory optimization for motion planning. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 4569–4574 (May 2011)
21. Karaman, S., Frazzoli, E.: Optimal kinodynamic motion planning using incremental sampling-based methods. Proc. IEEE Conf. Decision and Control pp. 7681–7687 (Dec 2010)
22. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. Int. J. Robotics Research 30(7), 846–894 (Jun 2011)
23. Kunz, T., Stilman, M.: Manipulation planning with soft task constraints. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 1937–1942 (2012)
24. Kurniawati, H., Hsu, D.: Workspace importance sampling for probabilistic roadmap planning. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). vol. 2, pp. 1618–1623. IEEE (2004)
25. Lindemann, S.R., LaValle, S.M.: Current issues in sampling-based motion planning. In: Robotics Research. The Eleventh International Symposium. pp. 36–54 (2005)
26. Pilania, V., Gupta, K.: A hierarchical and adaptive mobile manipulator planner with base pose uncertainty. Autonomous Robots 39(1), 65–85 (Jun 2015)
27. Plaku, E., Kavraki, L.E., Vardi, M.Y.: Hybrid systems: from verification to falsification by combining motion planning and discrete search. Formal Methods in System Design 34(2), 157–182 (2009)
28. Şucan, I.A., Chitta, S.: Motion planning with constraints using configuration space approximations. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 1904–1910 (2012)
29. Sun, Z., Hsu, D., Jiang, T., Kurniawati, H., Reif, J.H.: Narrow passage sampling for probabilistic roadmap planning. IEEE Transactions on Robotics (TRO) 21(6), 1105–1115 (2005)
30. Tanner, H.G., Loizou, S.G., Kyriakopoulos, K.J.: Nonholonomic navigation and control of cooperating mobile manipulators. IEEE Transactions on Robotics and Automation 19(1), 53–64 (2003)
31. Tanwani, A.K., Calinon, S.: Learning robot manipulation tasks with task-parameterized semitied hidden semi-markov model. IEEE Robotics and Automation Letters 1(1), 235–242 (2016)
32. Wise, M., Ferguson, M., King, D., Diehr, E., Dymesich, D.: Fetch and freight: Standard platforms for service robot applications. In: Workshop on Autonomous Mobile Service Robots (2016)
33. Wolfe, J., Marthi, B., Russell, S.J.: Combined task and motion planning for mobile manipulation. In: Int. Conf. Automated Planning and Scheduling. pp. 254–258 (2010)
34. Yang, Y., Brock, O.: Elastic roadmaps—motion generation for autonomous mobile manipulation. Autonomous Robots 28(1), 113 (2009)