Incorporating Online Obstacle Avoidance in E-Graphs

Shobhit Chaurasia
The University of Texas at Austin
Email: shobhit@utexas.edu

***Abstract*— Planning with E-graphs speeds up motion planning by leveraging previous planning episodes, while providing provable bounds on the sub-optimality of the generated plans. However, E-graphs assume a static environment and cannot handle unseen or moving obstacles. To address this issue, this paper proposes an integration of potential fields based online obstacle avoidance using DMP framework into the E-graphs framework. The proposed integration allows E-graphs to adapt to unseen obstacles, while avoiding the need for re-planning on account of obstacle avoidance.**

I. Introduction

Motion planning is crucial for robots to perform navigation and manipulation tasks that are free from collisions in a static or dynamic environment. However, the high dimensional state space and the high degrees-of-freedom (DOF) in humanoid robots (e.g. HRP-4[1] with 34 DOFs) and mobile manipulators (e.g. PR2[2] with 20 DOFs) render motion planning computationally intractable.

Most of the motion planning techniques employ heuristics to limit exploration of the exploding state space [1]. Probabilistic methods such as Probabilistic Roadmaps [3] and Rapidly-exploring Random Trees (RRT) [5] facilitate high-speed collision-free planning. However, they lack theoretical guarantees on the the sub-optimality of the generated motion plans (they provide probabilistic completeness guarantees). Another technique to speed up planning is to learn from previous planning episodes. Planning with Experience Graphs (E-graphs) [8] uses an online approach to motion planning by leveraging motion plans generated in the past. It speeds up planning by channelizing the search to states that are already present in the E-graph. E-graphs are well suited for repetitive tasks in static environments (e.g. manipulating objects in a kitchen) in which past experiences can be utilized to minimize re-planning.

A crucial assumption that planning with E-graphs makes is the static nature of the environment under consideration. The states that E-graph encodes includes static obstacles like wall, furniture etc. Planning with E-graph assumes that the graph correctly encodes the connectivity of the environment. Hence, in the presence of moving or previously unseen static obstacles, the motion plans generated by E-graph might result in collisions. An agent having perceived the presence of such an obstacle would then be forced to fall back to exploring the original state-space without any useful assistance from the previous episodes of planning stored in the E-graph.

To address this shortcoming, this paper proposes a method to incorporate online obstacle avoidance in motion plans generated by E-graphs. The main challenge is to combine motion plans generated by E-graphs with obstacle avoidance in a manner that minimizes the need for re-planning on account of obstacle avoidance.

Obstacle avoidance in motion planning is a well-studied problem. Commonly used motion planning techniques assume static environments and generate collision-free paths using search methods such as A*, bi-directional search, and Simulated Annealing [1]. To account for moving/unseen obstacles, re-planning is often required. Hence, these methods do not align with the essence of planning with E-graphs which lies in minimizing the requirement for re-planning. A powerful approach to obstacle avoidance is using the Potential Fields Method [4] that envisions obstacles as point sources emanating a repulsive potential field. The power of this methods lies in it being domain-agnostic. In this paper, planning with E-graphs is augmented with potential fields based obstacle avoidance. In particular, Dynamic Movement Primitives (DMP) framework combined with Potential Fields [6] is used.

The paper is organized as follows. Section II briefly discusses core ideas of E-graphs and DMP. Obstacle avoidance using potential fields is discussed in Section III, followed by the proposed integration in Section IV. Section V presents simulation results which are discussed in Section. VI. In Section VII, directions for further investigation are proposed. Finally, the paper concludes with a note on technical contributions in Section VIII.

II. Background

A. Experience Graphs

E-graph (G^E) encodes a subset of the original search space (G) representing the connectivity of the environment. It stores the motion plans generated by previous instances of planning requests. New requests for planning are sped up by channelizing the search to nodes already expanded on G^E, followed by searching on G^E as much as possible instead of the much larger G.

Let $G(V, E)$ represent the graph modeling the original planning problem. $c(u, v)$ denotes the edge cost between two nodes $u, v \in G$. Further, let $G^E(V^E, E^E)$ represent the Experience graph that is built over time, with $c^E(x, y)$ denoting the edge cost between two nodes $x, y \in G^E$.

[1]http://global.kawada.jp/mechatronics/hrp4.html
[2]http://www.willowgarage.com/pages/pr2/specs

Note the $c(u,v) = c^E(u,v)$ for all nodes $u, v \in G^E(\subseteq G)$. Planning with E-graphs uses a heuristic search with an admissible heuristic function $h^G(u,v)$ for computing an estimate of the cost of moving from u to v.

The core idea of E-graphs is to avoid exploring the original search space when similar motion plans are already present in the E-graph. More precisely, E-graphs use a special heuristic function which directs the search to a nearby node on the E-graph and re-uses parts of the motion plans stored in the E-graph to get the agent closer to the goal. For a state $s_0 \in V^G$, the heuristic function h^E is defined as:

$$h^E(s_0) = \min_{\pi} \sum_{i=0}^{N-1} \min\{\epsilon^E h^G(s_i, s_{i+1}), c^E(s_i, s_{i+1})\}$$

where $\pi =< s_0, s_1, \cdots, s_{N-1} >$ is a path with s_{N-1} as goal. ϵ^E is a scalar ≥ 1 weighting the original heuristic h^G with the aim of penalizing search on the original graph, and directing the search to expand states along parts of G^E.

E-graphs use weighted A* for searching. Weighted A* inflates the heuristic used by A* by a scalar $\epsilon > 1$ to speed up the search, while guaranteeing that the solution is no worse than ϵ times the optimal solution. Planning with E-graphs enjoy the theoretical guarantee that the cost of the generated solution is at most $\epsilon.\epsilon^E$ times worse than the cost of the optimal solution [8].

As noted in [8], leveraging E-graph leads to significant speed-up in planning for spatially repetitive tasks in a static environment (such as manipulation tasks in a household setting). While such settings present largely static environments, unforeseen static obstacles (e.g. toys lying on the ground) and moving obstacles (e.g. humans walking around) are not uncommon. Although it is possible for planning with E-graphs to account for such obstacles by keeping its graph up-to-date, an undesirable consequence of the same would be rendering of numerous motion plans stored in the E-graph invalid on account of their collision with the new obstacles. However, the said obstacles might only be present temporarily (e.g. books lying on the floor), or might be moving continuously. Disabling the previously learned motion plans on account of temporary collisions would, then, have the effect of unnecessary re-planning in future. This calls for incorporation of an online obstacle avoidance module which is independent of the E-graphs.

B. Dynamic Movement Primitives

Dynamic Movement Primitives (DMP) [6] is a framework for movement generation in which the agent's movements in the operational space are characterized by a set of differential equations, which uses a spring-mass damper system to guarantee goal converge, enhanced by additional coupling terms.

$$\begin{aligned}
\tau\dot{\mathbf{v}} &= \mathbf{K}(\mathbf{g}-\mathbf{x}) - \mathbf{D}(\mathbf{v}-\mathbf{v_d}) - \mathbf{K}(\mathbf{g}-\mathbf{x_0})s + \boldsymbol{\psi}(\mathbf{x},\mathbf{v}) \\
\tau\dot{\mathbf{x}} &= \mathbf{v} \\
\tau\dot{s} &= -\alpha s
\end{aligned}$$

where $\mathbf{x}$ and $\mathbf{v}$ are position and velocity of the system; $\mathbf{x_0}$ and $\mathbf{g}$ are start and goal positions respectively; v_d is the desired velocity at goal; τ is a temporal scaling factor; $\mathbf{K}$ is the spring constant and $\mathbf{D}$ is the damping coefficient (one for each DOF); s is a phase variable connecting the temporal space of all degrees-of-freedom. The real power of DMP-based motion generation comes from the flexibility it provides in adding coupling terms (perturbing forces) to explain the desired motion of the robot without sacrificing the stability of the desired movements, by abstracting away such specifications from the core spring-mass damper system that guarantees goal convergence.

One can add a non-linear coupling term $\mathbf{f}(s)$ to encode desired patterns in movement (e.g. through demonstration) [7]. Since pattern of movements are not at the core of online obstacle avoidance, this approach does not use this term. The obstacle avoidance aspect of DMP is due to the addition of a repellent acceleration term $\boldsymbol{\psi}(\mathbf{x},\mathbf{v})$ that is proportional to the gradient of a repelling potential field emanating from an obstacle. This term pushes the generated path away from an obstacle, while the spring-mass damper system pulls the path towards the goal, thereby producing the desired motion that circumvents an obstacle to converge to the goal. In the following section, different types of potential fields are discussed.

III. Potential Fields for Obstacle Avoidance

A. Static Potential Field

The potential field emanating from an obstacle is modeled as a static field whose influence is constrained within a region around the obstacle parametrized by a scalar p_0. A reasonable value for p_0 could be proportional to the radius of the bounding sphere of the robot under consideration. The repelling acceleration is given by $\boldsymbol{\psi}(\mathbf{x},\mathbf{v}) = -\vec{\nabla}U_s(\mathbf{x})$. The static potential field as proposed in [4] is given by:

$$U_s(\mathbf{x}) = \begin{cases} \frac{\eta}{2}(\frac{1}{p(\mathbf{x})} - \frac{1}{p_0})^2 & \text{if } p(\mathbf{x}) \leq p_0 \\ 0 & \text{if } p(\mathbf{x}) > p_0 \end{cases}$$

where η is a scalar constant controlling the strength of the field.

B. Dynamic Potential Field

The potential field emanating from an obstacle is modeled as a dynamic field whose influence is *not* constrained within a region around the obstacle. Its strength is inversely proportional to the distance from the obstacle, and depends on the relative velocity of the robot and obstacle, and their direction of approach to each other. The repelling acceleration is given by $\boldsymbol{\psi}(\mathbf{x},\mathbf{v}) = -\vec{\nabla}U_d(\mathbf{x},\mathbf{v})$. The dynamic potential field as proposed in [6] is given by:

$$U_d(\mathbf{x}) = \begin{cases} 0 & \text{if } 0 \leq \theta \leq \frac{\pi}{2} \\ \lambda(-\cos(\theta))^{\beta}\frac{||\mathbf{v}||}{p(\mathbf{x})} & \text{if } \frac{\pi}{2} < \theta \leq \pi \end{cases}$$

where η is a scalar constant controlling the strength of the field, and θ is the angle between the velocity ($\mathbf{v}$) and the position ($\mathbf{x}$) of robot relative to the obstacle. Dynamic potential field provides a more general obstacle avoidance scheme since not only does it account for moving obstacles, but also takes into consideration robot's speed and direction of approach even for static obstacles.

IV. Combining E-graphs with DMP

The broad structure of the proposed integration is inspired from the two-level path planning paradigm discussed in [2]. In their approach, a global planner relies on a global view of the environment's connectivity to generate a candidate path that minimizes the heuristic distance between the start and goal position. This is followed by a local planning stage which adapts the generated candidate path to obstacles, and locally optimizes the path length and smoothness of motion. In our case, planning with E-graphs is analogous to the global planning stage, while the online obstacle avoidance module works locally in the proximity of an obstacle, modifying the generated path to circumvent the obstacle.

Potential fields based DMP provides a domain-agnostic framework for online obstacle avoidance. The DMP framework can be easily plugged on top of E-graph generated plans for adapting movement plans to unseen or moving obstacles. The core idea that the approach exploits is the goal convergence guarantee that DMP provides. The first step is to identify points on the motion plan generated by E-graph that are in the proximity of the obstacle (e.g. within the region of influence of the potential field emanating from the obstacle). The start position for DMP is set to the first such point. The goal position for DMP is set to the corresponding point on the motion plan on the other side of obstacle. Since DMP guarantees convergence to the goal position, this approach gives us the guarantee that deviation on account of obstacle avoidance would not require re-planning on the part of E-graph, since the DMP will bring the agent back on the path originally generated.

In addition to the goal convergence guarantees, what makes DMP based online obstacle avoidance fitting to this domain is that it is fully decoupled from the underlying E-graph planner. It only influences the motion plans locally, without affecting the paths stored in the E-graph. Thus, unforeseen "temporary" obstacles do not end up polluting the useful past experiences stored in E-graph for future use.

It is worth noting that movement generation with DMPs itself provides a full-fledged framework for motion planning. In certain aspects, such as encoding a desired pattern in movement through basis functions, DMP framework is indeed more expressive than traditional search-based or sampling-based motion planners. A potential solution to the obstacle avoidance problem with E-graphs could then be to use DMP to generate entire motion plans and store these plans in the E-graphs. While such an approach is not technically incorrect and does account for unseen and moving obstacles, it lacks the theoretical guarantees that heuristic based planners (e.g. weighted A* in E-graphs) enjoy with regards to the quality of the generated path as compared to the optimal path.

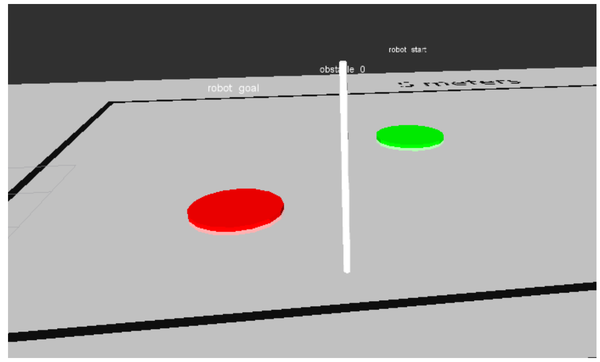


Fig. 1. Experimental Setup: Light gray region represents the world map, and black regions represent static obstacles. Green and red circles represent robot's base's start and goal positions respectively. The white stick is an unseen obstacle.

V. Results

The proposed method was tested in a simulated domain on RViz. Experiments involved a robot navigating in an environment having obstacles that are not part of the states present in the E-graph. Figure. 1 shows the experimental setup. The light gray region represents the static world map, having static obstacles (like walls) encoded by black regions. This map represents the static view of the world as encoded in the E-graph. White stick (interactive markers) represent unseen obstacles that can be moved around.

Fig. 2(a) presents a scenario in which an unseen obstacle lies near paths stored in E-graph. If the underlying graph is updated to account for the new obstacle, a major chunk of useful experiences stored in the E-graph will become invalid. Otherwise, the generated path will result in collision. Fig. 2(b) shows how DMP integration into E-graphs not only prevents collision with the new obstacle, but also preserves the previous experiences stored in E-graph. Additionally, such a non-intrusive integration allows the E-graph to add the new experience (without the obstacle avoidance adaptation) to its set with the hope that this new obstacle is only temporarily there, and the originally planned motion could be re-used in future.

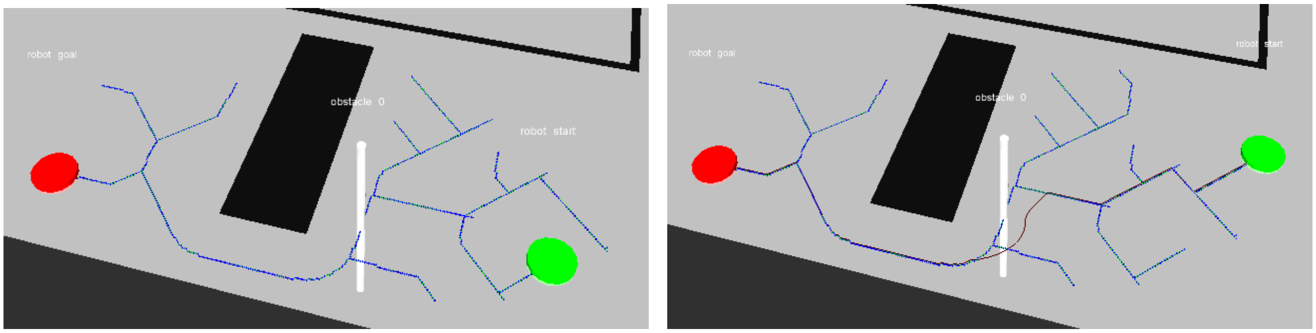
Fig. 3 compares the effect of obstacle avoidance using static fields vs. dynamic fields for a static obstacle (using comparable parameter values). Fig. 4 shows the simulation result for a moving obstacle at it moves from the bottom left corner to the top right corner, momentarily coming close to the robot's planned path.

Note how in all the cases, the DMP module only gets activated locally near the obstacle and brings the robot back to the originally planned path.

VI. Discussion

A. Hypothesis

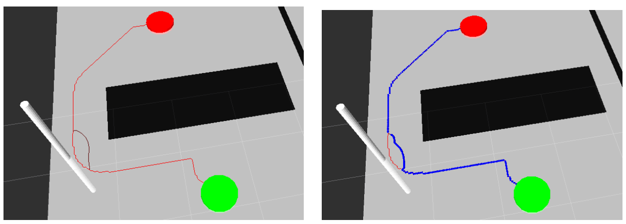
The original hypothesis going into this project was that the underlying functioning and utility of plans generated by E-graph would remain completely unaffected by the incorporation of obstacle avoidance. More precisely, the



(a) E-graph state prior to a new planning request. An unseen obstacle can render the past experiences stored in E-graph invalid.

(b) E-graph state after the new planning request. Obstacle avoidance handles the unseen obstacle, while preserving the experiences stored in E-graph.

Fig. 2. Blue paths represent the motion plans stored in E-graph. Brown path on right represents the path returned by E-graph+DMP for a new request.



(a) Static potential field

(b) Dynamic potential field

Fig. 3. Effect of obstacle avoidance for a static obstacle using static vs dynamic potential field. Red path is the one planned by E-graph alone. Brown path on left and blue path on right is returned by E-graph + DMP

incorporation of obstacle avoidance module would not trigger a requirement for re-planning.

While, theoretically, DMP provides goal convergence guarantees, the actual final position of the robot at the end of DMP might not precisely line up with node on E-graph corresponding to the path originally generated. This necessitates the need for a re-planning request to the E-graph to generate a plan between the actual final position of the robot and the intended goal position (that lies on the original path). However, this re-plan request is expected to be insignificant since the two nodes under consideration will be close to each other. In the experiments, this issue only caused an extra expansion of 1-5 nodes.

Hence, theoretically and ideally, the hypothesis claiming the unnecessity of re-planning on account of online obstacle avoidance holds. However, practically, we need to relax it a bit to allow for minimal re-planning to account for numerical inaccuracies in DMP.

B. Limitations

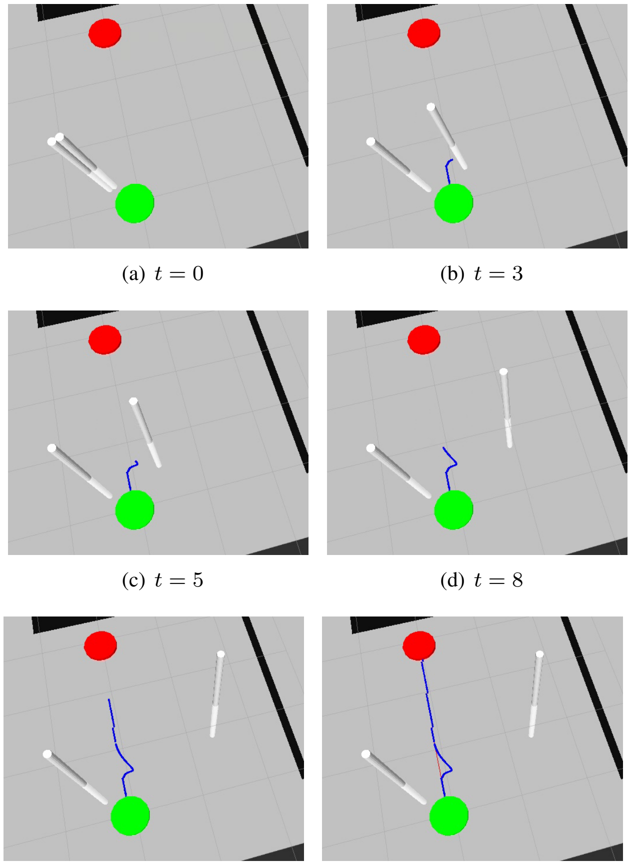
1) Moving Obstacles: Moving obstacles pose a non-trivial problem for the proposed integration. The dynamic potential fields based DMP when used independently can handle moving obstacles since the start and goal positions are predetermined and fixed. However, in its integration with E-graphs, the goal position needs to be anticipated in advance. The only constraint on the goal is that it should lie on the path originally generated by E-graph. However, for determining that position, the robot needs to anticipate the obstacle's motion. Such an anticipation requires active perception on the part of the robot to predict obstacle's path, or a simplifying assumption that the obstacle is moving with a constant speed whose value is known to the robot.

Fig. 5 shows how crucial timely perception and accurate estimation of obstacle's motion is to the result of the DMP module. In this example, the robot was deliberately provided delayed information about the position of the obstacle (i.e. the switch to the DMP module was delayed). The start position fed to DMP was alarmingly close to the obstacle. This resulted in a powerful repelling force experienced by the robot due to the dynamic potential. The resulting high speeds imparted to the robot on account of obstacle avoidance could be breaking.

2) Multiple Obstacles: In the current implementation, one call to the DMP module is designed to handle only one obstacle. This call to DMP adapts the path to avoid the said obstacle and bring the agent back to the original path. The current implementation can handle multiple "mutually exclusive obstacles", which do not interfere with each others' DMP-module call. However, the current implementation cannot handle multiple obstacles which come in the way of other obstacles' avoidance paths being generated by the DMP module. This could be easily addressed by actively changing the pre-selected goal for an old obstacle and switching to the new obstacle in the current DMP module call.

VII. Future Work

The proposed integration was tested only for 2-dimensional navigation tasks. Evaluating this approach for full-body planning could be an interesting extension, and might further reveal the strengths and weaknesses of this integration. Another direction of investigation could be to incorporate richer obstacle avoidance schemes such as the one proposed in [2] which too is based on potential fields, but also takes into consideration the configuration of robot while clearing obstacles in the path generated by a global planner. Another instance of a more expressive obstacle avoidance scheme is link collision avoidance proposed in [6] which uses at its heart potential fields based DMP framework. This



(a) $t = 0$ (b) $t = 3$

(c) $t = 5$ (d) $t = 8$

(e) $t = 14$ (f) $t = 16$

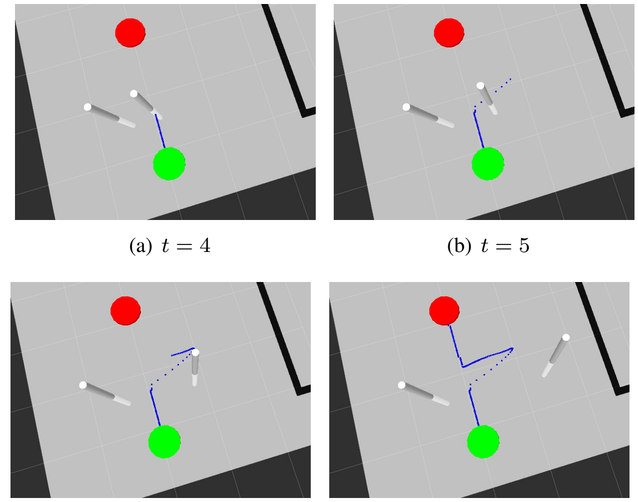
Fig. 4. Obstacle avoidance using dynamic potential field for a moving obstacle. Red line shows the original path planned by E-graph. The left white marker in all the figures shows the initial position of the obstacle, while the other marker shows its current position as it moves.

scheme extends the said framework to account for collision of robot's links with the obstacle, something which is crucial especially for arm navigation tasks. Finally, addressing the issues mentioned in Section VI-B and catering to other potential hurdles that might come with a full-body implementation could further strengthen the proposed approach.

VIII. Technical Contributions

The main technical contributions include the overall idea of incorporating online obstacle avoidance in E-graphs, and reviewing existing obstacle avoidance techniques to gauge their applicability in the domain of planning with E-graphs, keeping in mind the aim to minimize re-planning on account of obstacle avoidance. Additionally, a prototype for the proposed integration was built on top of the E-graphs implementation borrowed from the Search-based Planning Library (SBPL)[3]. The prototype implements the proposed integration for 2D navigation tasks with previously unseen static and moving obstacles. It includes visualization for the motion plans generated using E-graphs and DMP. The code is available as a GitHub repository[4].

[3]https://sbpl.pc.cs.cmu.edu/redmine/projects/ egraph-environments/wiki
[4]https://github.com/shobhit6993/egraphs-with-dmp



(a) $t = 4$ (b) $t = 5$

(c) $t = 7$ (d) $t = 8$

Fig. 5. Perils of inaccurate estimation of moving obstacle's position/path: Blue dots progressively show the position of the robot following the path generated by E-graphs + DMP. The time interval between any two consecutive blue dots is same; farther dots indicate higher speed.

References

[1] Y. K. Hwang and N. Ahuja. Gross motion planninga survey. *ACM Computing Surveys (CSUR)*, 24(3):219–291, 1992.

[2] Y. K. Hwang and N. Ahuja. A potential field approach to path planning. *Robotics and Automation, IEEE Transactions on*, 8(1):23–32, 1992.

[3] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, 1996.

[4] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

[5] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 995–1001. IEEE, 2000.

[6] D.-H. Park, P. Pastor, S. Schaal, et al. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 91–98. IEEE, 2008.

[7] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 763–768. IEEE, 2009.

[8] M. Phillips, B. J. Cohen, and S. Chitta. E-graphs: Bootstrapping planning with experience graphs. 2012.