Path Planning for Dexterous Mobility

Dirk Ruiken and **Michael W. Lanighan** and **Roderic A. Grupen**
Laboratory for Perceptual Robotics
School of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003, USA
{ruiken, lanighan, grupen}@cs.umass.edu

Abstract

In order to overcome a large variety of run-time constraints, robots are being designed to be more resourceful by incorporating more sensory and motor options for any given task. The added flexibility provides a basis for *dexterous* problem solving, but challenges planners by increasing the complexity of search. Moreover, the cost of functionally equivalent options can vary dramatically. In the worst case, naive approaches to planning avoid expensive actions until inexpensive options are explored exhaustively leading to poor overall search performance. We present a dexterous robot that introduces multiple types of locomotor actions with significant differences in cost and situational value and apply standard search techniques to demonstrate the additional challenges that arise in the context of dexterous mobility. Results highlight incentives, opportunities, and impact for overcoming these challenges. Additionally, we present a prototype for a path planner that uses environmental features to define an efficient set of subgoals for dexterous motion planning.

Introduction

Mobile manipulators may have to deal with a large variety of tasks and terrains in unstructured environments. State-of-the-art robots are often designed with excess degrees of freedom and with redundant sensors leading to the potential for sensory and motor flexibility. However, applications of these new machines require more than the *potential* for flexibility. In seminal work by Nikolai Bernstein, dexterity was defined as "*the ability to solve a motor problem correctly, quickly, rationally, and resourcefully*" (Bernstein 1996). The sensorimotor potential for dexterity poses challenges for planners because of high dimensional action spaces as well as dramatic variations in the cost of alternative actions. In the worst case, the large difference between costs can result in exhaustive exploration of inexpensive actions before more costly actions are considered. This leads to generally poor search performance.

To explore this issue, we focus on path planning for dexterous mobility and present a dexterous robot that introduces multiple types of locomotor actions with significant differences in costs and situational values. Although dexterity has been traditionally concerned with manual tasks, "it's not unreasonable to apply [dexterity] to locomotion" (Ma and Dollar 2011).

A classical planner is implemented to support navigation in an environment with various types of obstacles. Experiments are limited to static environments that are completely known *a priori*. A traditional A* planner is used to highlight the challenges arising in this application. We also present a first version of a hierarchical planner which uses a sparse set of subgoals in order to alleviate these problems.

The rest of the paper is organized as follows: first we provide details on the uBot-6 mobile manipulator, its postural modes, and the path planning algorithms. Simulation results are presented and used to demonstrate the challenges facing planning in this context as well as display the performance of the hierarchical planner. The paper concludes with an outlook into future work.

Robot Platform

The study employs uBot-6, a toddler-sized mobile manipulator under developement at the Laboratory for Perceptual Robotics at the University of Massachusetts Amherst. It has 13 degrees of freedom (DOF): two wheels, trunk rotation, two 4-DOF arms, and a 2-DOF head. uBot-6 provides multiple postural modes that support redundant types of mobility control with vastly different costs and value. The robot can balance on two wheels, it can "scoot" in a prone posture, and it can knuckle walk like a chimpanzee (Figure 1 left to right).

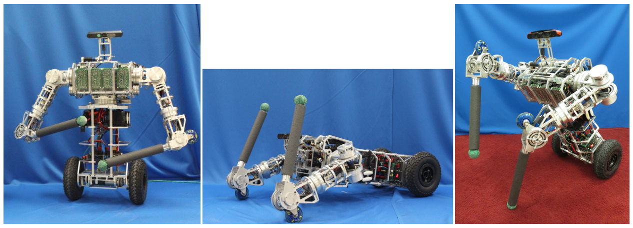


Figure 1: uBot-6 in several postural modes: balancing (left), prone scooting (center), knuckle-walking (right).

Balancing is very energy efficient and can be used on even ground. A small footprint and safety due to low input impedance of the balancer make mobility in this postural mode well suited for navigation around humans. Differential steering allows turning in place and navigation in tight spaces.

Prone scooting: by using the base wheels as well as passive wheels on the elbows, the robot can scoot in an Ackermann steering configuration. Arm motions are used for steering. While in this postural mode, the robot body can assume different body heights with implications in the minimum turn radius and power consumption. Two example body postures and their respective minimum turn radii can be seen in Figure 2. Scooting enables the robot to move under overhangs and with greater stability, but at the expense of energy relative to the balancing posture.

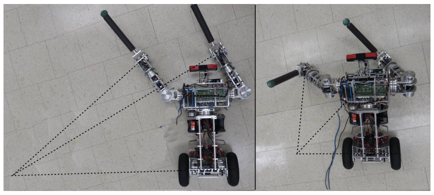


Figure 2: Comparison of turn radii for uBot-6 while prone scooting dependent on low body height: 1.2 m (left) and high body height: 0.68 m (right).

For *knuckle-walking* mobility the robot uses its base wheels and hands to walk. It provides uBot with the ability to traverse irregular terrain. However, this ability comes at the cost of higher energy consumption and lower speed compared to balancing or scooting.

uBot-6 can transition between its postural modes by following the paths in the postural transition graph (Figure 3). The postural modes, controllers, and costs of postural transitions are described in detail in (Kuindersma et al. 2009) and (Ruiken, Lanighan, and Grupen 2013). Compared to locomotion, transitions from one mode to another are relatively costly— in fact, they are an order of magnitude more expensive in terms of time and energy. As just two postural modes are sufficient to demonstrate the arising challenges for planners, for the experiments in this paper we use only balancing and prone scooting postural modes.

Cost Modeling

All mobility actions and postural mode transitions of the robot have been modeled with respect to time and energy based on empirical data. In this study, minimum-time plans are considered. In general, especially for mobile robots, the power consumption and probability of success of actions might be considered as well.

Table 1 shows the required time for driving in various postural modes and transitions between postural modes. Experiments use nominal velocities for driving actions though more detailed cost models and velocity constraints can be found in (Ruiken, Lanighan, and Grupen 2013). Prone scooting allows higher maximum drive velocities than balancing.

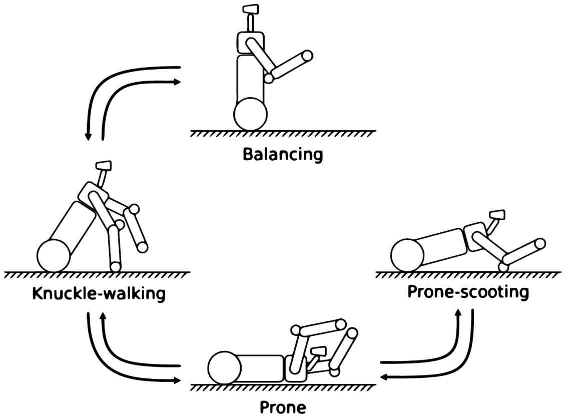


Figure 3: Postural modes and inter-modal transitions

Table 1: Cost with respect to time for driving and transitioning between postural modes.

Action	Cost (Time)
Driving: Balancing	$2.0\,\frac{s}{m}$
Driving: Prone scooting	$1.6\,\frac{s}{m}$
Transition: Balancing to low prone	$19.0\,s$
Transition: Low prone to balancing	$18.3\,s$

Path Planning

A navigation task of a mobile robot requires searching for a path in the environment using a set of actions with associated costs. Possibly the most well-known search algorithm is A* (Hart, Nilsson, and Raphael 1968), which finds an optimal path by combining actual and heuristic cost. While A* is very efficient, the size of the search space as well as time limitations often makes it infeasible to find a solution. To address this problem many anytime variations such as ARA* (Likhachev, Gordon, and Thrun 2003) have been developed which are able to quickly find a suboptimal solution and then use the remaining time to refine it. Another limitation of A* in real world settings is its inability to efficiently adjust to changes in dynamic environments. Efficient incremental variants of A* such as D* (Stentz and Mellon 1993) are able to repair existing solutions for a fraction of the cost of a complete replan. Both anytime and repair capabilities have been combined in algorithms like AD* (Likhachev et al. 2005).

The ability to change postural modes increases flexibility and resourcefulness, but also introduces a significant penalty for changing modes even though it can lead to obvious and efficient path plans. Heuristic planners will still find optimal solutions, but unless guided by very sophisticated heuristic functions, will avoid expensive actions such as mode transitions until inexpensive options are explored exhaustively.

Other planning approaches for dexterous robots have been attempted. Notably, motion primitives have been used with probabilistic road-maps (Hauser et al. 2008; Hauser and Latombe 2010). A*-like approaches using metrics such as power consumption, time, etc. are more commonly used for wheeled robots (Likhachev and Ferguson 2009). We utilize a heuristic based planner to showcase the challenges facing planners that exploit dexterity in a mobility domain.

Dexterous mobility on the uBot-6 introduces only a small number of transitions, but it is already difficult to create an adequate heuristic. The difficulty of finding suitable heuristics in similar domains has also been noted by (Hauser et al. 2008). We demonstrate the strong impact of dexterity and large variance in action costs on path planning performance by using a classical A* search with a *state lattice* on a simulated uBot-6. Additionally, we present the prototype of a hierarchical A* that utilizes environmental features to define efficient subgoals.

Classical A*

We use a standard A* search as a representative of classic heuristic search. We refrain from using any of its aforementioned variants as we are dealing with fully known static environments in our experiments. We are using a *state lattice* (Pivtoraiko and Kelly 2005) as a discretized representation of our state space. Each node is a discretization of a robot configuration with x, y location and heading θ in a postural mode m. Connections between *states* represent feasible paths and are constructed from the actions available to the robot. Each action can either be a short locomotion primitive or a postural mode transition. The connections in the state lattice are generated from feasible actions and therefore a found path will also be feasible making this representation well suited for path planning.

The action space of each lattice state consists of all locomotion and transition actions that are feasible given the current robot pose and configuration. For each postural mode, a set of actions with different drive distances and turn radii is available. The minimal turn radius R_{min} varies for the different postural modes. Upright balancing is the most versatile mode and allows turning in place through differential steering ($R_{min} = 0\,m$). Prone scooting has restrictions on the minimal turn radius with $R_{min} = 1.2\,m$ for low body height. Each action results in a trajectory from the current robot pose to another lattice state. An action is added to the action space of a state if no point along the trajectory is in collision with the environment. Depending on the postural mode, collision checks are performed in different collision maps corresponding to the respective body height and robot footprint. Additionally, all transitions to other postural modes are included if the transition actions can be executed collision free.

We extend a proven heuristic to explore the problems arising from the dramatic cost differences of actions. The heuristic is based on pre-computed values stored in a *heuristic look-up table* (Knepper and Kelly 2006), which allows quick retrieval of heuristic values during search. As the entire state space—despite being discretized—is very large, it is not practical to pre-compute and store heuristic values for all states. Instead, similar to (Likhachev and Ferguson 2009), we pre-compute heuristic values for a simplified state space based on just position and postural mode (x, y, m). This relaxation corresponds to the state space of a holonomic robot. Contrary to our non-holonomic robot, this relaxed model allows movement in any direction independently of heading. Using Dijkstra's search (Dijkstra 1959) starting from the goal we label states with the minimal cost they can be reached with. Movement in 2D uses Euclidean straight-line distance weighted with movement costs in the corresponding postural mode. Transition costs are the same as used in the actual search.

Feature driven Hierarchical A*

The second search algorithm we use is a hierarchical planner which employs A* on both abstract and low levels. The global planner finds an abstract path in a sparse set of subgoals which are sampled around features detected in the environment. The features used here are volumetric edges with discontinuities in depth as well as the frontier of explored space. While vertical edges (corners) and the frontier of unexplored territory promise discovery of new features through changed viewpoints in 2D, horizontal edges indicate places in which the transition into a different postural mode can both reveal new features and potentially indicate changed terrain constraints.

The availability and cost of transitions between neighboring subgoals is determined by a local planner. We use the described A* with state lattice. As the features are all detected visually, the paths between subgoals can be found very quickly even with a simple heuristic based on straight line distance. To avoid long searches when transitions are not available, the search is bound to a maximum search cost dependent on the heuristic. Transitions between postural modes only happen at subgoals and thus do not impede local search.

In contrast to existing A* based hierarchical planners like HPA* (Botea, Müller, and Schaeffer 2004), the abstract version of the search space is not just a grid with reduced resolution, but instead is dependent on features in the environment which are linked both to exploration and constraints on postural modes.

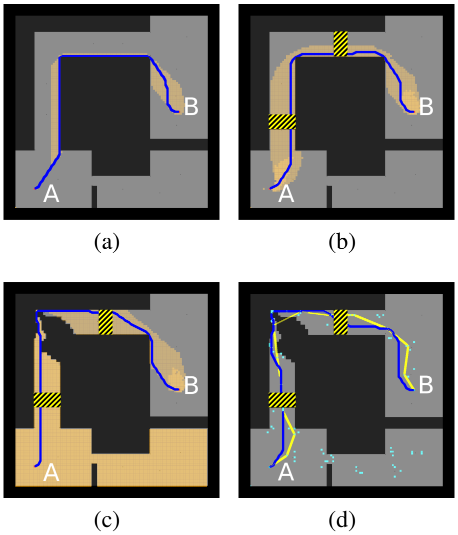
Experiments and Results

Dexterous mobility can be useful in a variety of situations. For example, a household helper with dexterous mobility will not be hindered by objects laying in its path and is capable of fetching a box that has been stored under the bed. In search and rescue settings a robot with dexterous mobility may overcome a wider variety of environmental obstacles.

In order to use the right combination of postural modes in such situations, we can use a planner. But dramatically different action costs pose challenges to planners as they rather explore inexpensive actions exhaustively before using vastly expensive alternatives. We use three similar example environments of varying difficulty to showcase the utility of dexterity in mobility and also highlight the challenges that arise for planners from it.

Imagine a search and rescue setting. In three scenarios the robot has to move from location A in one room through an angled hallway to location B in another room. In the first scenario (Figure 4(a)) the hallway is unobstructed and the robot can stay in balancing postural mode all the way. In the second scenario (Figure 4(b)), both entrance and exit of the hallway are obstructed by an obstacle which prevents the robot from passing through upright. After changing its postural mode, the hallway can be traversed by prone scooting. The third scenario (Figure 4(c)) is identical to the second scenario, but the hallway is more narrow. Due to relatively large minimum turn radii while prone scooting, the corner can only be taken while balancing. Thus the robot has to scoot to enter and exit the hallway, but needs to be balancing to take the obstructed corner.

For each scenario, obstacle maps are provided for balancing and prone scooting postural modes. Both maps can be considered as being dilated such that we can treat the robot as a point. As we are not using a 3D simulation environment, an additional map provides the location of vertical and horizontal edges needed for feature extraction.



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

Figure 4: Scenarios 1 - 3 are shown in (a) - (c) respectively. The paths found are shown in blue. Cells shaded orange were expanded in search. In (b) and (c) the hallway separating A and B is obstructed with low-hanging obstacles preventing balancing mobility (black-yellow shaded areas). Additionally, in scenario 3, rubble narrows the passage throught the hallway. Fig. (d) shows the abstract and refined path found by the hierarchical planner for scenario 3.

Figure 4 shows the solutions found by the two algorithms for each scenario and illustrates the state space expanded. A roll-out of the plan found by A* for scenario 3 can be seen in Figure 5. Table 2 shows the search time and the number of expanded states for the two algorithms in each scenario. As expected, the scenario 1 and 2 pose no problem to the A* search as the costs are reflected very well by the heuristic. But in scenario 3 we see a large negative effect when the heuristic is not accurate enough. In this case turning the corner is not possible while prone scooting due to non-holonomic constraints. As the used heuristic is a holonomic approximation, in this case it is not accurate enough. As a result all less expensive actions are exhaustively tried before the necessary transitions between postural modes are considered. By reducing the state space using environmental features, the hierarchical planner yields a substantial improvement to this search time in scenario 3 though the found paths are not optimal.

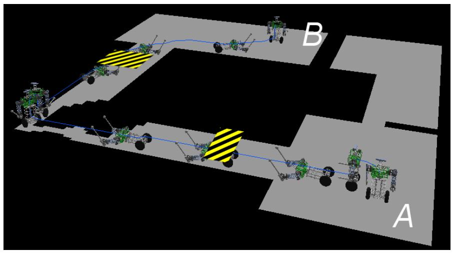


Figure 5: Roll-out of solution shown in Figure 4(c).

Table 2: Search results for A* and hierarchical A* with sparse subgoals (HA*) over the three scenarios

Scenario	Alg.	Search time	Expanded states
1	A*	0.21 s	7327
2	A*	0.51 s	22850
3	A*	192.73 s	4798315
1	HA*	1.82 s	28
2	HA*	6.54 s	73
3	HA*	13.01 s	144

Conclusion and Future Work

We presented the dexterous mobility domain as an instance of planning for dexterous systems. We have demonstrated on a simple example problem how heuristic search—which underlies many state of the art techniques—is challenged by dramatic differences in action costs. While these issues can partially be overcome with more accurate heuristic models of the problem, as robots become more dexterous these heuristics will become increasingly difficult to model accurately enough. We introduced a hierarchical search algorithm that might alleviate some of the arising problems from dramatically different action costs in dexterous mobility by using a sparse search space built from environmental features.

Future work encompasses extending dexterous mobility to more postural modes. We plan to extend the hierarchical planner to partially observable and dynamic environments. As the waypoints are already based on features which also provide potential exploration of unknown terrain, one strategy is to include the expected information gain to guide the search algorithm more efficiently. To this end, we plan on investigating symbolic planners (Dornhege et al. 2009) with our sparse feature representation.

Acknowledgment

The authors gratefully acknowledge the support of NASA-GCT-NNX12AR16A and ONR-MURI-N000140710749 for this work.

References

Bernstein, N. A. 1996. On dexterity and its development. In Latash, M. L., and Turvey, M., eds., *Dexterity and its development*. Mahwah, NJ: Lawrence Erlbaum Associates. 1–237.

Botea, A.; Müller, M.; and Schaeffer, J. 2004. Near optimal hierarchical path-finding. *Journal of game development* 1(1):7–28.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* 1(1):269–271.

Dornhege, C.; Eyerich, P.; Keller, T.; Trüg, S.; Brenner, M.; and Nebel, B. 2009. Semantic attachments for domain-independent planning systems. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*, 114–121. AAAI Press.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on* 4(2):100–107.

Hauser, K., and Latombe, J.-C. 2010. Multi-modal motion planning in non-expansive spaces. *The International Journal of Robotics Research* 29(7):897–915.

Hauser, K.; Bretl, T.; Harada, K.; and Latombe, J.-C. 2008. Using motion primitives in probabilistic sample-based planning for humanoid robots. In *Algorithmic Foundation of Robotics VII*. Springer. 507–522.

Knepper, R. A., and Kelly, A. 2006. High performance state lattice planning using heuristic look-up tables. In *IROS*, 3375–3380.

Kuindersma, S.; Hannigan, E.; Ruiken, D.; and Grupen, R. 2009. Dexterous mobility with the ubot-5 mobile manipulator. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, 1–7.

Likhachev, M., and Ferguson, D. 2009. Planning long dynamically feasible maneuvers for autonomous vehicles. *The International Journal of Robotics Research* 28(8):933–945.

Likhachev, M.; Ferguson, D.; Gordon, G.; Stentz, A.; and Thrun, S. 2005. Anytime dynamic a*: An anytime, replanning algorithm. *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)* 262–271.

Likhachev, M.; Gordon, G.; and Thrun, S. 2003. Ara*: Anytime a* with provable bounds on sub-optimality. *Advances in Neural Information Processing Systems (NIPS)* 16.

Ma, R. R., and Dollar, A. M. 2011. On dexterity and dexterous manipulation. In *Advanced Robotics (ICAR), 2011 15th International Conference on*, 1–7. IEEE.

Pivtoraiko, M., and Kelly, A. 2005. Generating near minimal spanning control sets for constrained motion planning in discrete state spaces. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, 3231–3237. IEEE.

Ruiken, D.; Lanighan, M. W.; and Grupen, R. A. 2013. Postural modes and control for dexterous mobile manipulation: the umass ubot concept. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, 721–726. IEEE.

Stentz, A., and Mellon, I. C. 1993. Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics and Automation* 10:89–100.