CHAPTER III

Navigation

Navigation is the task of generating a series of commands that allow a robot to transit from its current pose to some goal pose. Within this task are a number of subtasks such as path planning, environment sensing, obstacle avoidance and others. As a robot transits through space from its current to goal pose it necessarily visits a series of intermediate poses that are a function of its kinematic constraints and the constraints of the environment it transits through. Collectively this set of poses as a time sequence is known as a path. The set of all possible poses is known as the task space and all kinematic feasible configurations is known as the configuration space. Typically the environment a robot traverses is not free space but also contains obstacles which constrain the paths which can be taken to the goal. While in some controlled environments the location of obstacles can be known *a priori* and supplied to the robot in the form of a map, often the robot will have the task of creating a map which describes the location of these obstacles. In some situations “global” sensors can view the entire environment which the robot will operate and can capture all of the constraints while in others (especially in the case of mobile robots) only “local” sensors mounted to the robot are available to integrate environmental limitations to the robot’s path as they are encountered. Even more challenging is the case where the environmental obstacles are not static but are moving adding a time varying element to the obstacle avoidance problem.

Algorithms that solve the navigation problem are sometimes broadly classified into four major categories (though there are many other categories which these algorithms could be grouped into). The first two, known as global or local, speak to the amount of spacial data or time horizon considered when planning a solution. If only a short time horizon or occlusions in the immediate spacial region are accounted for then the algorithm is considered local, otherwise it is thought of as global. The other two deal with how the spaces or paths are represented, continuously or discretely. If the algorithm

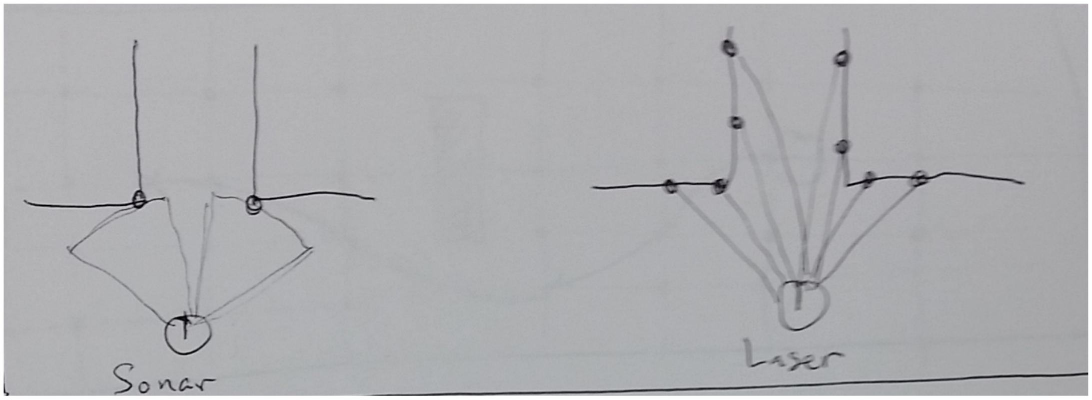

Figure 3: Sonar cones vs laser beams.

breaks up these spaces into finite resolution pieces then it is discrete. Continuous algorithms choose some sort of parameterization to the spacial or environmental constraints using a model. Often algorithms are some mixture of all of these four categories.

The algorithm used for navigation in this thesis is the GODZILA navigation algorithm, which is a form of local continuous variety based on the potential fields navigation strategy. While in many environments GODZILA solves the navigation problem, it can also be used in conjunction with global algorithms that can plan way points to the goal without having to consider the detailed local environment.

Something to consider is how different local sensors can affect the choice of navigation algorithm. The Nao humanoid platform used in this thesis is equipped with two sonar sensors for obstacle detection. Details on these can be read in Chapter II. These sensors have a broad angular range and cannot provide information about where within this cone obstacles are located. This allows the robot to avoid colliding with objects directly in front of it but restricts the available paths in the case where the width of the space between the obstacles approaches the width of the robot. Conversely, scanning laser rangefinders (such as the one mounted to the Nao and described in Chapter II) have a very high degree of angular resolution and provide hundreds of range measurements. This allows for a more accurate description of environmental occlusions allowing more paths to be considered for traversal.

The notion of what objects in the environment occlude a path and which do not can be viewed as a function of the gait (or mode of locomotion) used by the robot. If the robot is restricted to a plane, such as the case in many ground vehicles, then an object of similar size to the vehicle will prevent traversal through that position. If the robot

Figure 4: Global vs local.

can also fly then the robot can simply fly over it. While still not transiting through that exact position, the 2D projection of the path onto the plane will appear to have moved through the obstacle, making a 2D planner applicable to the problem. In Chapter **??** a crawling gait for the Nao robot is considered which allows it to go under occlusions that it would otherwise need to plan around.

3.1 Algorithm Classifications

3.1.1 Global

One approach to the navigation problem is to consider the entire task and configuration space of a problem when generating a solution. For example, the task space of a robot arm might be the pose of the end effector and all of the occlusions within that space. The configuration space would be the set of all valid joint angles the arm can attain. Using this an algorithm can compute a time sequence of poses or movement commands that the robot should execute in order to bring it from its current pose to the final pose. While this approach is ideal in the sense of generating a solution to the original problem, it has several practical disadvantages. One requirement of the algorithm is having a complete description of the task space. In many cases global knowledge of occlusions while planning is not available meaning that when new obstructions are observed the algorithm needs to replan the path. The other commonly encountered problem with these algorithms is the magnitude of the space to be planned through is so large that computing a solution cannot be done in real-time.

3.1.2 Local

A different approach to the navigation problem is generating commands based only on the current information available or a short history about the environment. With the robot arm example, there might be a sensor such as a camera that can detect objects that are obstructing the end effector from reaching the goal and instructs the arm to move around it. Such algorithms usually have the advantage of being quick to compute as compared to their global counterparts because they only have to consider a small subset of the task and configuration space. Their main disadvantage is that since they only use such a limited amount of information they can become trapped in local optima that do not allow the robot to achieve the desired pose.

3.1.3 Discrete

Orthogonal to the ideas of local and global path planning which deals with the scope of the time and space being considered when generating navigation commands is how these spaces are represented. One way to represent the space is to break it up into discrete representations and then plan through that. The task space for example could be divided into a uniform spacial regions which an algorithm can consider visiting when planning a path. Alternatively, as with the robot arm example, a finite set of movement commands can be considered and iterated through while generating a solution. Graph-based search algorithms such as Dijkstra's algorithm or A* are an example of discrete solvers. An advantage to this is paths with complex shapes can be generated to accommodate difficult constraints. A problem with this approach is how finely to resolve the task or configuration spaces. Coarser discretization can allow a solution to be computed rapidly but miss more optimal solutions.

3.1.4 Continuous

Continuous spacial representation avoids the problem of choosing a discretization resolution. The movement commands or paths are planned in a continuum allowing paths to take on intermediate values not available at a given discrete resolution. Continuous representation instead has a different problem of paths needing to take on particular solution forms. A path through the task space might be represented as a cubic spline from the current pose the goal pose. This path has access to the entire task space but can only have two bends in it. If there are obstructions in the environment which would require more turns than this a different representation would have to be chosen.

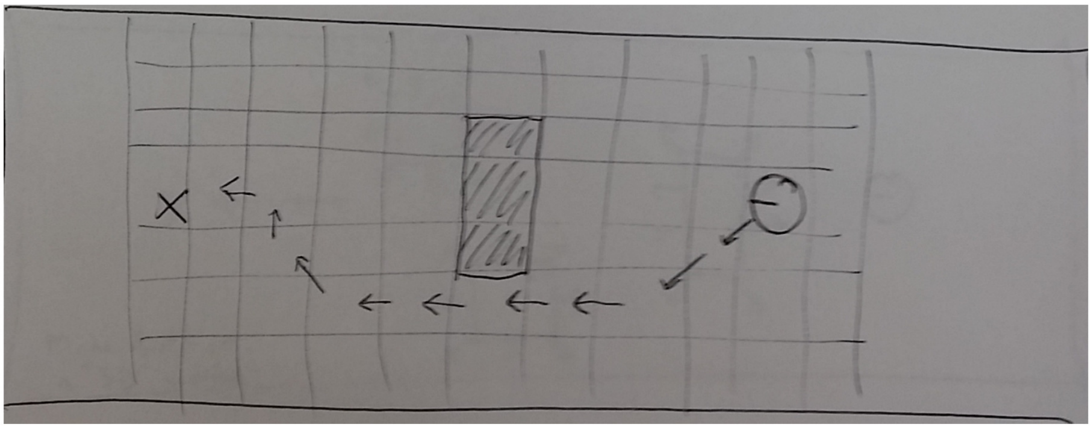

Figure 5: Grid world path.

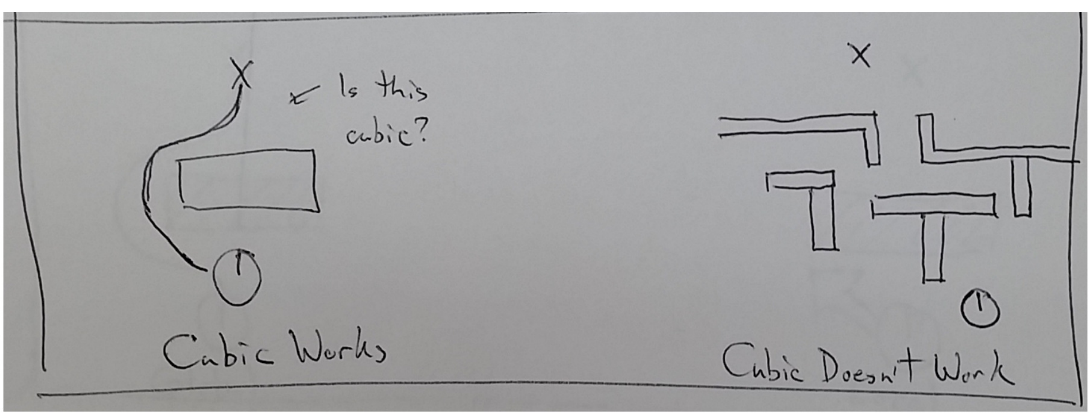

Figure 6: Environments where cubics work and doesn't.

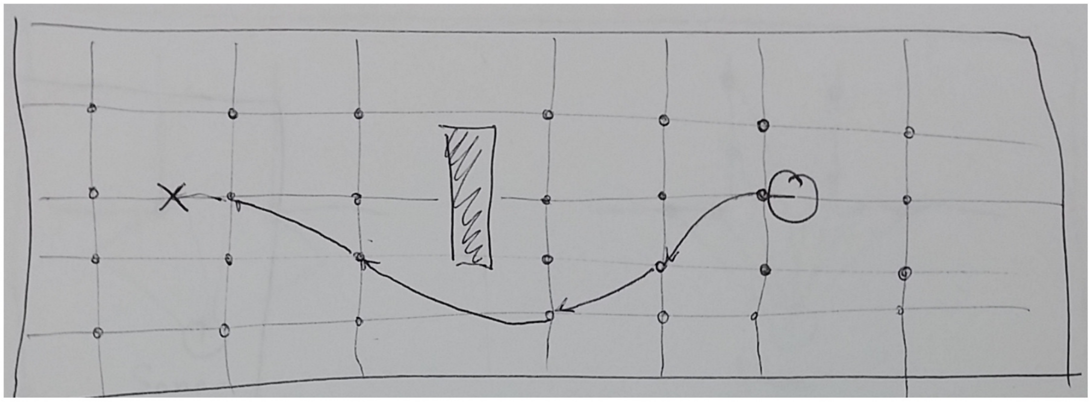

Figure 7: Lattice planner, hybrid discrete and continuous.

3.1.5 Composite Approaches

In order to combine the strengths and combat the weakness the above approaches are often combined to form a more robust navigation solution. Global plans can be generated as a series intermediate goals or way points for local planners to navigate towards. These way points can be planned on a coarse discrete grid that is quick to compute while low dimensional continuous trajectories are plotted between them. Lattice planners are an example of such a composite algorithm.

3.2 Potential Fields

Potential fields algorithms are continuous local planners. They work on the concept of modeling the robot as a sort of charged particle (like an electron) and obstacles in the environment produce a repulsive field pushing the robot away from them. The goal location in turn acts as a attractive force, pulling the robot towards it. As the robot move through the environment it detects objects and generates motion commands based on their range and direction.

3.3 GODZILA

GODZILA (Game-theoretic Optimal Deformable Zone with Inertia and Local Approach) [1] is a local continuous navigation algorithm based on the potential fields idea. It has three additions to potential fields which are a straight line planner, stochastic local minima escape strategy, and an inertia term to dampen high frequency movements and back tracking.

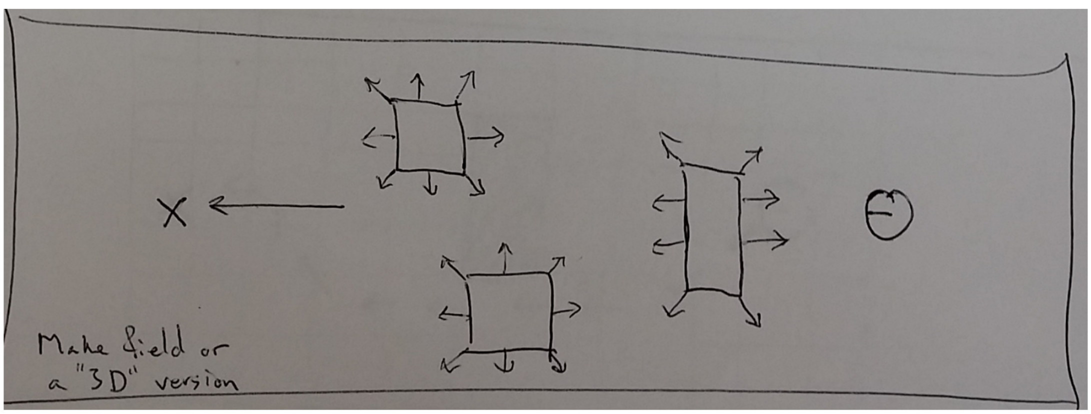

Figure 8: Visualization of potential fields.