Dynamic Map: Representation of Interactions between Robots

Christian Zanardi

GRPR, École Polytechnique de Montréal
P.O. Box 6079, succ. "Centre-ville",
H3C 3A7, Montréal, CANADA
email:zanardi@ai.polymtl.ca

Introduction

As robotics applications become more complex, the need for tools to analyze and explain interactions between robots has become more acute. We introduce the concept of Dynamic Map (DM), which can serve as a generic tool to analyze interactions between robots or with their environment. We show that this concept can be applied to different kinds of applications, like a predator-prey situation, or collision avoidance.

The Dynamic Map (DM)

For a dynamic system, represented by the state equation $\dot{x} = f(x, u, t)$, where x is the state of the system, u the control vector, and t time, T-reachable regions are commonly defined as the set of all states that can be reached within time T from a same initial position $x(0)$, with an acceptable control vector u. Although these regions indicate *which* points can be reached, they do not inform about (the *quality* of) the trajectories leading to these points. This information is provided by a goodness functional g—depending upon the task at hand—defined at each point of the T-reachable regions, and that must be maximized. We call this extension of the reachable regions Dynamic Map (DM) of the system.

Constructing and Using the DM With a simple model of a car-like robot, where the control is the steering angle, the Dynamic Map can be constructed along two methods. First, all points that can be reached are exhaustively generated using bang-bang controls, along with the value on the associated functional. Second, it is possible to contruct the map, using only "limits curves." The points on those curves are defined to be the external boundaries of reachable regions. It is then simple to generate the shape of the Dynamic Map based on those curves.

Ideally the goodness functional g should take into account important criteria such as presence of obstacles, energy consumption, and relative density of trajectories. Indeed, the values corresponding to points that belong to an obstacle should be negative in order to prevent trajectories to pass by them. Energy consumption is an important criteria, if a robot need to join periodically with an other to regain power. Since planned trajectories may have to change according to new environmental conditions while the initial goals remain, it would be important to know the relative density of trajectories leading to the neighboorhood of a point—*i.e.* the number of different trajectories leading to a similar position—as a mesure of confidence in the planned trajectory. Figure 1 presents an example of a DM, with such a functional—represented as shades of gray—for a car-like robot.

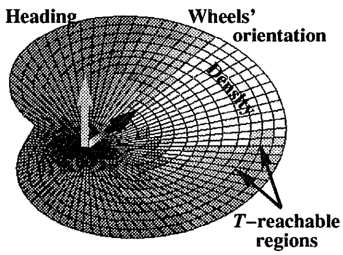

Figure 1: Example of DM

For example, if a mobile robot R is trying to evade its pursuer F, it would be safe to know *which* places R can reach before its adversary, and *how long* before F does. For such a problem, it is simply a matter of composing Dynamic Maps, *e.g.* through an addition. Thus, the robot will just have to combine an estimation of its adversary's map with its own, according to their relative pose. Then, using a simple maximization algorithm, the robot will be able to plan an evasion trajectory.

Conclusions

We presented here the Dynamic Map which is a tool to analyze interactions between mobile robots. One of the great advantages of the dynamic map—in a learning scheme context—is that it is built at the scale of the robot itself. Furthermore, it remains that the DM is a general concept that can be extended to a large number of dynamic systems, such as a robot manipulator.