

Title	対象のダイナミクスを考慮した最適経路問題に関する研究
Author(s)	湯, 紅偉
Citation	
Issue Date	2007-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/3582
Rights	
Description	Supervisor:平石 邦彦, 情報科学研究科, 修士

A Research on Optimal Path Problem Considering Dynamics of Objects

Hongwei Tang (510069)

School of Information Science,
Japan Advanced Institute of Science and Technology

Feb. 8, 2007

Key words: Optimal path, Obstacle avoidance, Piecewise linear systems

In this research, we study a problem of finding an optimal obstacle-avoidance trajectory of a moving object with given continuous dynamics.

As one of existing approach to this problem, the following method was proposed. First the entire region in which the object moves around is partitioned into subregion each of which is a convex polyhedron and has linear dynamics. Then the problem is formulated as an optimal control problem on a PWL (Piecewise Linear) system. Since in this method collisions with obstacles are avoided by adding restrictions on transitions between subregions, it does not necessarily give the optimal trajectory. Moreover, in order to improve the accuracy of the solution, it is necessary to make the partition finer, and as a result the computation time becomes longer. In this research, we focus on the process of region partitioning, and propose a new method to enable some of transitions forbidden in the previous method by adding new types of transitions between subregions. By this method, we can have more accurate solutions.

Comparing with the existing research, the following two new processes are added in the proposed approach.

The first one concerns “reachable regions”. With respect to collision with obstacles, there are the following three types of transitions between subregions. (i) Arbitrary path that goes through any point in the departure region and any point in the destination region does not intersect obstacle regions. (ii) Some of the paths intersect obstacle regions but some do not. (iii) All paths intersect obstacle regions. Obviously, the third type should be disabled, we only consider the first and the second types.

In the previous method, all transitions of the second type are forbidden and only the

transitions of the first type are allowed. Contrary, we allow here transitions of the second type with some restrictions. Concretely, we first compute the reachable region in the destination region that is reachable from some point in the departure-region. Then, instead of considering transitions from the departure region to all area of the destination region, we enable transitions from the destination region to the reachable region in the departure region. By doing this, we can include a part of transitions of the second type that do not intersect obstacle regions. By allowing such transitions forbidden in the previous research, we can have more accurate solutions.

The second one concerns “feasible regions”. At each step of moving, we can approximate the feasible regions for the current position as a rectangular region, based on the state equation of the object and the feasible region at the previous step. Since the feasible region is usually smaller than the departure region, we can have a wider reachable region in the destination region by using the feasible region as the current region, instead of using the departure region itself. In the proposed method, we compute intersection of the feasible region and the departure region at every step, and use it to compute the reachable region. In addition, it is able to compute parametric solution to the reachable region for any rectangular region in the departure region. That is, giving each position of the rectangle as a parameter, we can compute formulas representing the reachable region. We have implemented the above idea in computer programs and made computer experiments. As a result, we can obtain better solutions in shorter times.

Moreover, we made two improvements in the implementation. The first one is to give a Perl script that automatically generates a CLP (Constraint Logic Programming) code file. In the previous method, we need to rewrite a program every time when we change the initial data (the region for movement, positions of obstacles, and dynamics of the object). For this problem, we summarize how to compute transitions between subregions in the form of an algorithm. Given any input data, the Perl script generates all necessary CLP codes for computing the solution. This enables us to make many experiments in a shorter time.

The second one is to implement a visualization tool for drawing trajectories of the solution. Since the output of the CLP code is just a list of coordinates of points, it is hard to understand and evaluate it by intuition. We give a tool to convert the list of coordinates into an input of Matlab, which is used for drawing a graph of the trajectory.

Using the improvements described above, we can easily evaluate the proposed method for various kinds of problem instances.