Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
DAAAM Symposium

COLLISION DETECTION BY USING BEZIER CURVES AND GENETIC ALGORITHMS IN MOTION PLANNING

IRING, P[eter] & SCHREIBER, P[eter]

Abstract: *The paper describes the calculation of a trajectory collision with obstacles including mathematical and programming apparatus. It also proposes the design of piecewise functions for an obstacles description and easier implementation. The last part demonstrates the usage of Bezier curves and genetic algorithms for a trajectory optimization on a model problem with the description of used parameters.*
Key words: *genetic algorithms, Bezier curve, robot, collision detection*

1. INTRODUCTION

Intelligent path prediction is one of basic tasks in autonomous mobile robot control. There are several approaches for prediction varying in used method of path description and/or in method of optimization.

The sequence of line segments is one of the simplest method used for path description. Final path length is sum of all line segment lengths. To one of advantages of this method belongs its simplicity and lower CPU load comparing with other methods. Drawback of methods based on line segments are obvious. The created path is not "too close" to optimum and shape of trajectory is not "smooth enough". Robot or vehicle should stop at the end of each line segment in order to change the direction because sharp turns are not feasible for them (Sugihara & Smith, 2007). These approaches use genetic algorithms as optimization method and trajectory length as optimization criteria. Another method (Choi & Elkaim, 2010) for using GAs for optimization is usage of grid of adjacent cells with unit distance creating binary coded and fixed-length strings representing the paths. Total length of weighted path is criteria for minimization where Euclid distance of two cells calculated for partial result used in consecutive length summation.

Several researches use Bezier curves for path description. Also wrappers (corridor constraints) can be used for cubic Bezier curve as replacement of straight lines to make find optimal path. Many application can be found in filed of robot soccer competitions (Jolly et al., 2009) or robot collision avoidance (Skrjanc & Klancar, 2009) as well.

This article concentrates trajectory length calculation introduced in working space with obstacles and if it is possible to provide an effective way of handling with obstacles applicable in GAs.

2. COLLISION LENGTH CALCULATION

The calculation of Bezier curve length by formulas

$$\mathrm{l}=\int_0^1 \mathrm{f(t)dt}\approx\frac{1}{3}\mathrm{h}\left[\mathrm{f}(t_0)+2\sum_{j=1}^{\frac{n}{2}-1}\mathrm{f}(t_{2j})+4\sum_{j=1}^{\frac{n}{2}-1}\mathrm{f}(t_{2j-1})+\mathrm{f}(t_n)\right] \quad (1)$$

where $t_0 = 0$, $t_n = 1$ and

$$f(t)=\sqrt{g(t)} \quad (2)$$

$$g=(A_X^2+A_Y^2+A_Z^2)t^4+2(A_X\ B_X+A_Y\ B_Y+A_Z\ B_Z)t^3+[2(A_X\ C_X+A_Y\ C_Y+A_Z\ C_Z)+B_X^2+B_Y^2+B_Z^2]t^2+(B_X\ C_X+B_Y\ C_Y+B_Z\ C_Z)t+(C_X^2+C_Y^2+C_Z^2) \quad (3)$$

is well-transformable into the sequence of additions and multiplications and the integration using Simpson's rule is an algorithm-able method. A closed interval [0,1] was used during the integration for the curve length calculation.

For 3rd degree Bezier curve coefficients $A_X, A_Y, A_Z, B_X, B_Y, B_Z, C_X, C_Y, C_Z$ holds

$$A_i = 9(p_{i1}+p_{i2})+3p_{i3} \quad (4)$$
$$B_i = -12p_{i1}+6p_{i2} \quad (5)$$
$$C_i = 3p_{i1} \quad (6)$$

for $i \in \{X,Y,Z\}$ and where $P_1=\{p_{X1},p_{Y1},p_{Z1}\}$ $P_2=\{p_{X2},p_{Y2},p_{Z2}\}$ $P_3=\{p_{X3},p_{Y3},p_{Z3}\}$ are the control points of Bézier curve and determinate the shape of the trajectory. The point P_0 is always assumed to be placed in the origin of Cartesian coordinate system because the curve length does not depend on its position. It means that the whole curve is moved into the origin of a coordinate system. Therefore, it holds $P_0=\{0,0,0\}$.

The generalization of formula (1) leads to

$$\mathrm{l}=\int_{t_\alpha}^{t_\beta}\mathrm{f(t)dt}\approx\frac{1}{3}\mathrm{h}\left[\mathrm{f}(t_\alpha)+2\sum_{j=1}^{\frac{n}{2}-1}\mathrm{f}(t_{2j})+4\sum_{j=1}^{\frac{n}{2}-1}\mathrm{f}(t_{2j-1})+\mathrm{f}(t_\beta)\right] \quad (7)$$

where $t_\alpha \in [0,1]$ and $t_\beta \in [t_\alpha, 1]$. Although both the intervals are closed cases when $t_\alpha = 0$ or $t_\beta = 1$ represent only theoretical possibilities because points $P(p_X(0),p_Y(0),p_Z(0))$ and $P(p_X(1),p_Y(1),p_Z(1))$ are starting and ending points of trajectory and obstacle in one of these points means that there is no path to reach the objective. The main goal is then to find an effective way for searching parameters t_α and t_β.

The generalized formula (7) then can be used for the calculation of the collision of Bezier curve with obstacles.

3. OBSTACLES DESCRIPTION

It is not sufficient only to find out if the curve collides with an obstacle but also the exact amount of collisions. Therefore finding of the endpoints t_α and t_β of interval t is the most important part and the basis of the algorithm. Probably there are several possible ways of the calculation. This article focuses only on one of them.

A simple geometric description of obstacles in a working space is a precondition of the fast run of a collision algorithm. Collision of trajectory with an obstacle occurs when at least one point belonging to curve describing the trajectory lies inside the obstacle or in other words

$$f\{P(p_X(t),p_Y(t),p_Z(t))\}=1 \quad (8)$$

where

$$p_j(t)=\sum_{i=0}^{n=3}\binom{n}{i}p_{ji}(1-t)^{n-i}t^i \quad (9)$$

for $j \in \{X,Y,Z\}$.

Let us define a piecewise function f which identifies if the point of Bezier curve lies inside or outside the obstacle (outside or inside the working space / is a part of the obstacle or not).

$$f(t) = \begin{cases} 1, P\big(p_X(t), p_Y(t), p_Z(t)\big) \in O \\ 0, P\big(p_X(t), p_Y(t), p_Z(t)\big) \notin O \end{cases} \quad (10)$$

where O is the set of all points comprising the obstacle. The box obstacle is represented in a very simple way by its center $P(X_T, Y_T, Z_T)$ and dimensions a, b and c.

The definition of the function f gives us an opportunity to define a uniform interface for an easier implementation of obstacles and allows us to use it in a simple way regardless an obstacle shape.

4. ITERATIVE WALK ALONG THE BÉZIER CURVE AND GA USAGE

The algorithm walks along the Bezier curve by increasing the parameter t by predefined the step s. In each iteration is formula (9) evaluated three times for X, Y, Z and result in the position of the point P in 3D space for actual parameter t. Point P serves as an input for the piecewise function f which denotes if point P is or is not "inside" the obstacle. Two consequent values $f\{P(t-s)\} = 0$, $f\{P(t)\} = 1$ identify that the parameter t_α lies somewhere in the middle of $t-s$ and t. The endpoints of the interval $[t-s, t]$are used as input parameters with a smaller step (e.g. $\frac{1}{10}$ of actual s) in the next iteration of the algorithm to get a more accurate result. The analogue sequence of $f\{P(t-s)\} = 1$ and $f\{P(t)\} = 0$ represents the presence of t_β somewhere in the middle of t-s and t. The search for $t_\beta$can start at t_α and the values from $t = 0$ to $t = t_\beta$ do not need to be evaluated again for this obstacle. The algorithm continues until the desired accuracy is reached. The enough accurate parameters t_α and t_β are used for the collision length calculation. The collision detection algorithm has to be used with each obstacle in a working space.

A drawback of this algorithm is that collisions with "thin" obstacles (e.g. a piece of tin – one dimension is considerably smaller then the others) may not be detected because of a large step. If some control points of Bezier curve are located close to each other (in other words one control point is isolated – "too far" from the others) the algorithm may not detect the collision with the obstacle in case of a large step as well. Choosing a too large step increases the performance of the algorithm on one the hand but decreases the precision on the other. The length of collision is an input parameter for a penalization function in a later stage of GA. Therefore a imprecise calculation may have a negative influence on the convergence of GA. To improve convergence proper form of testing purposed in (Tanuska et al., 2009) have to be chosen.

There is a demand on speed (high performance) of the algorithm because each individual of each generation has to be evaluated for a collision. The amount of the collision has to be calculated for each obstacle in working space. According to overall amount of collision each individual has to penalized. ,

The given example uses an additive and static form of a penalization. It was chosen for its simplicity as a very first draft during the testing of the algorithm described in this paper.

Given example visualized in fig. 1 uses population size 100 with mating pool size 80, double values for gene coding, single type of crossover, global mutation with probability 0.9, and fixed chromosome length.

The graphical output shows the best individual as a result of GA for searching an optimal path from the starting point [0,0,0] to the ending point [100,100,100] after 200 generations and wire model of obstacles.

Each coordinate of each point was coded as one double value gene except the starting and ending points. It is not necessary to include these points into the search because they are already known. In depicted example step s=0.01 was used in the first iteration of algorithm for walk along the curve. The algorithm found the path close to optimal solution using a gap between the obstacles.

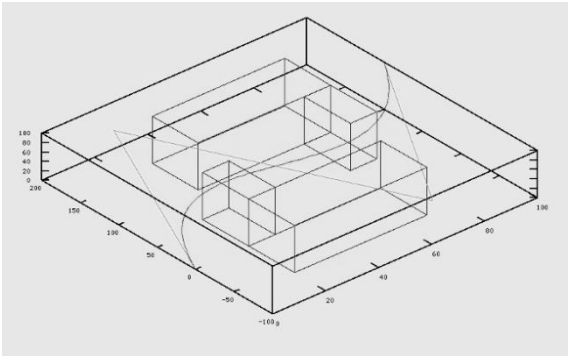

Fig. 1 Graphical representation of the best individual in the working space with obstacles

5. CONCLUSION

This paper shows that calculation of trajectory length can be reused also for successful collision detection. Introduced algorithm with suitable geometric description of basic shapes of obstacles provide also the way for calculation of length of collided trajectory. Setting of proper iteration step has main influence on algorithm performance and precision of calculation. Obstacles create non-continuities in working space. Usage of penalty functions seems to be appropriate method for dealing with them also in this specific case of optimization.

For an effective run of the described algorithm it is necessary to choose the proper form and type of a penalty function. The drawbacks of described algorithm have to be solved. The usage of chained Bézier curves has not been implemented and tested yet and also further research of genetic operations (mutation, crossover) in curves connections is interesting area for a deeper exploration. The most successful chromosomes coding has not been defined yet either.

6. ACKNOWLEDGEMENTS

This article has been supported by the Slovak Grant Agency of the Ministry of Education within the Project VEGA 1/0368/08 "Artificial Intelligence in Flexible Manufacturing Systems Control".

7. REFERENCES

Choi, J. & Elkaim, G. H. Bezier curve for trajectory guidance. *Availible from:* www.soe.ucsc.edu/~elkaim/Documents/ BezierWCES08.pdf *Accessed:* 2010-05-01

Jolly, K. G.; Sreerama Kumar, R. & Vijayakumar, R. A (2009) *Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits*. Robot. Auton. Syst. 57, 1, 23-33

Sugihara, K. & Smith, J. (1997) Genetic algorithms for adaptive motion planning of an autonomous mobile robot. Computational Intelligence in Robotics and Automation, 1997. CIRA'97., *Proceedings., 1997 IEEE Inter-national Symposium*, 138--143

Skrjanc, I. & Klancar, G. (2010) Optimal cooperative collision avoidance between multiple robots based on bernstein-bézier curves. *Robot. Auton. Syst*. 58, 1, 1-9

Tanuska, P.: Moravcik, O. & Vazan, P. (2009) The base testing activities proposal. In: *Annals of AAAM and Proceedings of DAAAM Symposium. 25 - 28th November 2009*, Vienna, Austria. DAAAM International Vienna, 2009. - ISBN 978-3-901509-70-4