Collision Problems by an Improved CAD-based Quantifier Elimination Algorithm

Hoon Hong*

November 21, 1991

Research Institute for Symbolic Computation
Johannes Kepler University
A-4040 Linz, Austria

Abstract

In 1987 Kalkbrenner and Stifter tested Collins' QE algorithm on various collision problems. In this paper we report the test results of applying Hong's improved QE algorithm on the same set of problems. The tests show that the improved QE algorithm gives significant speedups, ranging from 30 times to at least 7000 times.

1 Introduction

In robotics we are often interested in whether two objects will collide with each other. When the objects can be described algebraically, the collision problems can be easily formulated as quantifier elimination problems (QE) in the first order theory of real closed fields [2].

In 1975, Collins gave a QE algorithm based on Cylindrical Algebraic Decomposition [5], which was fully implemented by Arnon [1] in 1981. In 1990, Hong made various improvements of Collins' QE algorithm, mainly based on the idea of *Partial* Cylindrical Algebraic Decomposition [6].

In 1987, Kalkbrenner and Stifter tested Collins' QE algorithm on various collision problems [7]. In this paper, we report the test results of applying Hong's improved QE algorithm on the same set of problems. The test shows that the improved QE algorithm gives significant speedups., ranging from 30 times to at least 7000 times.

The plan of the paper is as following: In Section 2 we briefly describe the type of collision problems tested by Kalkbrenner and Stifter and also by us.

*Sponsored by the Gröbner Bases Project of the Austrian Ministry for Science and Research.

In Section 3 we review the main ideas on which Hong's improved algorithm is based. In Section 4 we present empirical comparisons between Collins' algorithm and the improved algorithm.

2 The Type of Collision Problems Tested

In a collision problem, we are interested in deciding whether several moving objects will collide at some time. More generally we are also interested in finding conditions on the shapes of the objects and their motion which avoid collision.

When the objects can be described as semi-algebraic sets and their displacements as semi-algebraic functions, the collision problems can be expressed as quantifer elimination problems in the first order theory of real closed fields. For the details, see [2].

In [7] Kalkbrenner and Stifter studied a type of collision problems with two objects, one stationary and the other moving along a given path. Specifically they studied the problems of the following structure:

Given: Two formulas $A(\vec{x})$ and $B(\vec{x})$ describing two semi-algebraic objects respectively. A path $\vec{p}(t)$ for the object B, expressed in parametric representation.

Task: Decide whether the two objects will collide, or find a necessary and sufficient condition on the collision-free path $\vec{p}$.

The decision problem can be easily formulated as:

$$(\exists t)(\exists \vec{x}) \;\; A(\vec{x}) \;\wedge\; B(\vec{x} - \vec{p}(t)).$$

The collision-free path condition problem can be formulated as:

$$\neg(\exists t)(\exists \vec{x}) \;\; A(\vec{x}) \;\wedge\; B(\vec{x} - \vec{p}(t))$$

or equivalently as

$$(\forall t)(\forall \vec{x}) \;\; \neg A(\vec{x}) \;\vee\; \neg B(\vec{x} - \vec{p}(t)).$$

3 Review of Improved QE algorithm

Now we give the highlights of Hong's improvement on Collins' QE algorithm. For the details, see [6, 3].

Collins' algorithm consists of three phases: projection, truth-invariant CAD construction, and solution formula construction. Hong made improvements on each of the three phases.

Improvements in the projection phase: By generalizing a lemma on which the proof of the original projection operation is based, we are able to reduce the size of the projection set significantly, and thus speed up the whole QE process.

Improvements in the truth-invariant CAD construction phase: By intermingling the stack constructions with the truth evaluations, we are usually able to complete quantifier elimination with only a partially built CAD, resulting in significant speedup.

Improvements in the solution formula construction phase: By constructing defining formulas for a collection of cells instead of individual cells, we are often able to produce simple solution formulas, without incurring the enormous cost of augmented projection.

4 Empirical Results

In this section we present empirical comparisons between Collins' QE algorithm and Hong's improved QE algorithm on several collision problems of the type described in Section 2.

Kalkbrenner and Stifter tested Collins' QE algorithm on 19 collision problems. We tested Hong's improved QE algorithm on 14 problems out of them. The five problems we did not tested are Example 1, Example 2, Example 3, Example 12, and Example 11. The first four examples are skipped because they are too trivial to be of interest. Example 11 is skipped because it is not a collision problem, but a path finding problem. We also divided the 14 problems into two categories: the first 7 problems do not have free variables and thus are decision problems, and the last 7 problems have a free variable k and thus are quantifier elimnination problems in proper.

In the following, A describes the stationary object, B describes the moving object, and p describes the path for the object B in parametric representation.

4.1 Decision Problems

Example 4: Square — Circle on a line

$$\begin{array}{lcl} A & : & x \leq 1 \ \wedge \ y \leq 1 \\ B & : & x^2 + y^2 = 1 \\ p & : & (t, 3 - t) \end{array}$$

Example 5: Square — Circle on a parabola

$$\begin{array}{lcl} A & : & x \leq 1 \ \wedge \ y \leq 1 \\ B & : & x^2 + y^2 = 1 \\ p & : & (t, t^2 - 2t + 3) \end{array}$$

Example 6: Square — Circle on a parabola

$$\begin{array}{lcl} A & : & x \leq 1 \ \wedge \ y \leq 1 \\ B & : & x^2 + y^2 = 1/2 \\ p & : & (t, t^2 - 2t + 3) \end{array}$$

Example 7: **Square — Circle on a cubic curve**

$$\begin{aligned} A &: x \le 1 \ \wedge \ y \le 1 \\ B &: x^2 + y^2 = 1 \\ p &: (t, t^3 - 5t + 6) \end{aligned}$$

Example 8: **Cube — Sphere on a line**

$$\begin{aligned} A &: x \le 1 \ \wedge \ y \le 1 \ \wedge \ z \le 1 \\ B &: x^2 + y^2 + z^2 = 1 \\ p &: (t, 0, -t + 3) \end{aligned}$$

Example 9: **Cube — Sphere on a quadratic curve**

$$\begin{aligned} A &: x \le 1 \ \wedge \ y \le 1 \ \wedge \ z \le 1 \\ B &: x^2 + y^2 + z^2 = 1 \\ p &: (t^2 + 1, 2t, t^2 - 2t + 2) \end{aligned}$$

Example 10: **Torus — Sphere on a quadratic curve**

$$\begin{aligned} A &: (\exists b)(\exists c)(b^2 + c^2 = 9/4 \ \wedge \ x^2 + (y - b)^2 + (z - c)^2 = 1/4) \\ B &: x^2 + y^2 + z^2 = 1 \\ p &: (t, t, t^2) \end{aligned}$$

4.2 Quantifier Elimination Problems

Example 13: **Line — Circle on a line**

$$\begin{aligned} A &: x = 5 \\ B &: x^2 + y^2 = 1 \\ p &: (t + 2, kt + 7) \end{aligned}$$

Example 14: **Line — Circle on a half line**

$$\begin{aligned} A &: x = 5 \\ B &: x^2 + y^2 = 1 \\ p &: (t + 2, kt + 7) \ \wedge \ t \ge 0 \end{aligned}$$

Example 15: **Half line — Circle on a half line**

$$\begin{aligned} A &: x = 5 \ \wedge \ y > 20 \\ B &: x^2 + y^2 = 1 \\ p &: (t + 2, kt + 7) \ \wedge \ t \ge 0 \end{aligned}$$

Example 16: **Half line — Circle on a line**

$$\begin{aligned} A &: x = 5 \ \wedge \ y \le 20 \\ B &: x^2 + y^2 = 1 \\ p &: (t + 2, kt + 7) \end{aligned}$$

	Kalkbrenner/Stifter	Ours
QE Algorithm	Collins'	Hong's
Computer Algebra System	SAC-2	SAC-2
Programming Language	FORTRAN	C
DPR and DQR	Assembly	C
Computer	IBM 4341	DEC Station 5000

Table 1: Test Environments

Example 17: Line — Circle on a line

$$\begin{array}{lcl} A & : & y = 3 \\ B & : & x^2 + y^2 = 1 \\ p & : & (t+2, kt+7) \end{array}$$

Example 18: Line — Circle on a half line

$$\begin{array}{lcl} A & : & y = 3 \\ B & : & x^2 + y^2 = 1 \\ p & : & (t+2, kt+7) \ \wedge \ t < 0 \end{array}$$

Example 19: Half Line — Circle on a line

$$\begin{array}{lcl} A & : & y = 3 \ \wedge \ x < 0 \\ B & : & x^2 + y^2 = 1 \\ p & : & (t+2, kt+7) \end{array}$$

4.3 Test Environments

In Table 1 we show the test environments for Kalkbrenner/Stifter's and ours. Both implementations are done on top of the computer algebra system SAC-2 developed by Collins [4]. When Kalkbrenner/Stifter carried out their experiments, the SAC-2 system was written in the programming laguage ALDES developed by Loos [8], and translated into FORTRAN. In 1989, Hong translated SAC-2 system into C language, and implemented the improved QE algorithm on top of the C-based SAC-2. Our other experiments in Sun3 show that C-based SAC-2 system is about 3 times faster than FORTRAN-based SAC-2 system.

DPR and DQR stand for digit product and digit quotient/remainer respectively. They are primitives for long integer arithmetic, and are called numerous times during typical algeraic computations. Thus it is usuall rewritten in assembly language for speedup. For our experiments, however, we coud not manage to rewrite them in assembly language mainly because we could not get hold of a manual. Our other experiments in Sun3 indicate that QE system with assembly version of DPR/DQR is about 3 times faster than QE system with C version of DPR/DQR.

4.4 Test Results

In Table 2 we show the timings for Collins' QE algorithm and the Improved QE algorithm. The timings of Collins' QE are from Kalkbrenner/Stifter [7]. All the times are in seconds. Several tests have been aborted after 2 hours of computation, which is indicated by the symbol $>$ appearing before 7200.

From the table we can see that in all cases, except Example 10, the improved QE algorithm is much faster than Collins' QE. Especially Example 7 and Example 9 show very significant speedup.

In Table 3 we show the output formulas produced by Collins' QE algorithm and the improved algorithm. For several examples, the tests did not run to completion and we do not know the output formulas which would have been produced. These cases are indicated by the symbol "?". For Examples 15 and 19, the output formulas produced by Collins' QE algorithm is so large that we did not include here.

4.5 Discussion on Example 10

Example 10 naturally draws out attention since even the improved QE algorithm requires more than 2 hours. Let us take a closer look at this problem and try to find some way to solve it.

This problem is different from all others in that the stationary object is described by a quantified formula. This results in an input formula with 6 variables instead of 3 or 4 variables. The trace of the test shows that even the projection phase has not been completed after 2 hours of computation.

One way around this difficulty is to break the problem into two subproblems and tackle each one by one. First, we eliminate quantifiers from the quantified formula describing the torus. Next, using the resulting quantifier-free formula, we form a decision problem with 4 variables. Specifically, we first eliminate quantifiers from the formula:

$$(\exists b)(\exists c)\ \ (b^2 + c^2 = 9/4\ \wedge\ x^2 + (y-b)^2 + (z-c)^2 = 1/4),$$

obtaining an equivalent quantifier-free formula $\bar{A}(\vec{x})$. Then we decide the formula:

$$(\exists t)(\exists \vec{x})\ \ \bar{A}(\vec{x})\ \wedge\ B(\vec{x} - \vec{p}(t)).$$

We tried this method and found that the two objects will collide. Specifically, the first quantifier elimination produced the following formula after about 2000 seconds:

$$z^4 + 2y^2z^2 + 2x^2z^2 - 5z^2 + y^4 + 2x^2y^2 - 5y^2 + x^4 + 4x^2 + 4 \leq 0.$$

Then after about 1200 seconds of computation, the decision prodecedure reported collision. Therefore this method took 3200 seconds in total.

Problem	Collins'	Improved	Speedup
Example 4	121.1	0.9	135
Example 5	80.7	0.9	90
Example 6	64.0	1.0	64
Example 7	>7200.0	1.0	>7200
Example 8	99.0	2.9	34
Example 9	>7200.0	3.3	>2182
Example 10	>7200.0	>7200.0	?
Example 13	22.6	0.3	75
Example 14	31.2	0.3	104
Example 15	351.2	4.8	73
Example 16	278.7	2.9	96
Example 17	18.9	0.3	63
Example 18	27.5	0.4	69
Example 19	211.7	2.2	96

Table 2: Computing Time

Problem	Algorithm	Output Formula
Example 4	Both	$0 = 0$
Example 5	Both	$0 = 0$
Example 6	Both	$1 = 0$
Example 7	Collins	?
	Improved	$0 = 0$
Example 8	Both	$0 = 0$
Example 9	Collins	?
	Improved	$0 = 0$
Example 10	Both	?
Example 13	Both	$1 = 0$
Example 14	Both	$1 = 0$
Example 15	Collins	a formula with 11 atomic formulas
	Improved	$4k^2 - 39k + 84 \geq 0 \wedge 4k - 13 \leq 0$
Example 16	Collins	$4k^2 - 39k + 84 > 0 \wedge k - 6 > 0$
	Improved	$4k^2 - 39k + 84 > 0 \wedge 4k - 13 \geq 0$
Example 17	Both	$k = 0$
Example 18	Collins	$k < 0 \wedge k = 0$
	Improved	$k \leq 0$
Example 19	Collins	a formula with 11 atomic formulas
	Improved	$k \leq 0 \vee (3k^2 - 16k + 15 \geq 0 \wedge 2k - 5 \geq 0)$

Table 3: Output Formula

5 Conclusion

In this paper, we reported the test results of Hong's improved CAD-based QE algorithm on various collision problems previously studied by Kalkbrenner and Stifter. These tests show that the improved QE algorithm is siginficantly faster than Collins' original algorithm. Except Example 10, all the examples which were nontrivial with Collins' algorithm now became trivial, taking only few seconds. It renews the hope of practical applicability of QE algorithms for robot motion planning.

As illustrated in the last section, Example 10 is still very time consuming and seems to be a challenging problem for future research and continued improvements.

I woud like to thank Michael Kalkbrenner and Sabine Stifter for explaining their experiments to me.

References

[1] D. S. Arnon. Algorithms for the geometry of semi-algebraic sets. Technical Report 436, Computer Sciences Dept, Univ. of Wisconsin-Madison, 1981. Ph.D. Thesis.

[2] B. Buchberger, G. E. Collins, and B. Kutzler. Algebraic methods for geometric reasoning. *Annual Review of Computer Science*, 1989.

[3] G. E. Collins and H. Hong. Partial CAD construction in quantifier elimination. Technical Report OSU-CISRC-10/89 TR45, Computer Science Dept, The Ohio State University, 1989. To appear in *Journal of Symbolic Computation.*

[4] G. E. Collins and R. G. K. Loos. SAC-2 system documentation. Technical report. In Europe available from: R. G. K. Loos, Universität Tübingen, Informatik, D-7400 Tübingen, W-Germany. In the U.S. available from: G. E. Collins, Ohio State University, Computer Science, Columbus, OH 43210, U.S.A.

[5] George E. Collins. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In *Lecture Notes In Computer Science*, pages 134–183. Springer-Verlag, Berlin, 1975. Vol. 33.

[6] H. Hong. *Improvements in CAD–based Quantifier Elimination.* PhD thesis, The Ohio State University, 1990.

[7] M. Kalkbrenner and S. Stifter. Some examples for using quantifier elimination for the collision problem. Technical Report 87-22, Research Institute for Symbolic Computation, 1987.

[8] R. G. K. Loos. The algorithm description language ALDES (Report). *ACM SIGSAM Bull.*, 10(1):15–39, 1976.