View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Bath Research Portal

Citation for published version:
Wilson, D, Davenport, JH, England, M & Bradford, RJ 2013, A "piano movers" problem reformulated. in Proceedings of SYNASC 2013: 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. IEEE, pp. 53-60, SYNASC 2013: 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, 23/09/13. https://doi.org/10.1109/SYNASC.2013.14

DOI:
10.1109/SYNASC.2013.14

Publication date:
2013

Document Version
Peer reviewed version

Link to publication

© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

University of Bath

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 13. May. 2019

A "Piano Movers" Problem Reformulated

David Wilson, James H. Davenport, Matthew England & Russell Bradford
Department of Computer Science, University of Bath, Bath, BA2 7AY, UK
E-mail: {D.J.Wilson, J.H.Davenport, M.England, R.J.Bradford}@bath.ac.uk

***Abstract*—It has long been known that cylindrical algebraic decompositions (CADs) can in theory be used for robot motion planning. However, in practice even the simplest examples can be too complicated to tackle. We consider in detail a "Piano Mover's Problem" which considers moving an infinitesimally thin piano (or ladder) through a right-angled corridor.**

Producing a CAD for the original formulation of this problem is still infeasible after 25 years of improvements in both CAD theory and computer hardware. We review some alternative formulations in the literature which use differing levels of geometric analysis before input to a CAD algorithm. Simpler formulations allow CAD to easily address the question of the existence of a path. We provide a new formulation for which both a CAD can be constructed and from which an actual path could be determined if one exists, and analyse the CADs produced using this approach for variations of the problem.

This emphasises the importance of the precise formulation of such problems for CAD. We analyse the formulations and their CADs considering a variety of heuristics and general criteria, leading to conclusions about tackling other problems of this form.

I. Introduction

A. A "Piano Movers" problem

In [24] the authors describe a *"Piano Movers" Problem* as follows: "given a body B and a region bounded by a collection of walls, either find a continuous motion connecting two given positions and orientations of B during which B avoids collisions with the walls, or else establish that no such motion exists." Such problems commonly arise in robotics.

A simple example from [14] is the problem of moving a ladder of length 3 through a right-angled corridor of width 1 (moving from position 1 to position 2 in Figure 1). A simple analysis shows there is no solution to this particular problem, and that it would only be possible to traverse the corridor with a ladder of length less than $\sqrt{8}$. We are interested in how this and similar piano movers problems may be decided automatically, with paths calculated when a solution exists.

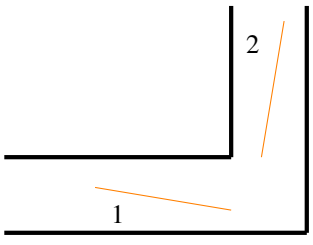

Fig. 1. The piano movers problem considered in [14]

In [25] the authors proposed a generic approach to piano movers problems in which the problem is described using polynomial algebra and then solved using the cylindrical algebraic decomposition (CAD) algorithm. However, for even very simple examples this approach can be computationally infeasible. In [14] the author applied the approach of [25] to the simple problem of the ladder just described, demonstrating the scale of the computations that would be required. Despite 25 years of improvements in both CAD theory and computer hardware, producing a CAD for the algebraic formulation given in [14] is still infeasible.

In Section II we provide a new formulation for which a CAD has been produced and from which path could be deduced. First we complete the introduction with a reminder of the theory of CAD, details of the original formulation and a summary of other formulations found in the literature. Some of these can solve the existential question of whether a path exists very quickly, but they cannot then give the actual path that would be required by a robot, which the formulation presented in Section II can. In Section III we consider generalisations of the problem and how the some of the formulations could be adapted while in Section IV we consider further adaptations to CAD technology for use with piano movers problems. Finally we give our conclusions in Section V.

B. Cylindrical algebraic decomposition

A cylindrical algebraic decomposition (CAD) is a partition of $\mathbb{R}^n$ into cells, constructed with respect to an input, usually either polynomials or formulae, in n ordered variables. Each cell is described by a semi-algebraic set (a finite sequence of polynomial equations and inequalities) and the cells are cylindrically arranged (meaning the projection of any two cells on the first k coordinates is either equal or disjoint).

A CAD is sign-invariant if the input polynomials have constant sign on each cell. Such a CAD allows for the solution of many problems defined by the polynomials. Collins provided the definition and first algorithm [1], motivated as a tool for quantifier elimination in real closed fields. Other applications range from robot motion planning to algebraic simplification technology [5], [15].

Collins' algorithm has two phases. The first, *projection*, applies a projection operator repeatedly to a set of polynomials, each time producing another set in one fewer variables. Together these contain the *projection polynomials*. The second phase, *lifting*, then builds the CAD incrementally from these polynomials. First $\mathbb{R}$ is decomposed into cells which are points and intervals corresponding to the real roots of the univariate polynomials. Then $\mathbb{R}^2$ is decomposed by repeating the process over each cell using the bivariate polynomials at a sample point of the cell. The output for each cell consists of *sections* of polynomials (where a polynomial vanishes) and *sectors* (the regions between these). Together these form the *stack* over the

cell, and taking the union of these stacks gives the CAD of $\mathbb{R}^2$. This process is repeated until a CAD of $\mathbb{R}^n$ is produced. The projection operator must be chosen in order to conclude that the CAD of $\mathbb{R}^n$ produced in this way is sign-invariant.

We note that CADs can depend heavily on the ordering of the variables. In [10] a problem was described which led to a cell count doubly exponential in the number of variables for one ordering, but constant in another. Heuristics to help pick the variable ordering are developed in [16], [7].

Since Collins published the original algorithm there has been much research into improvements with a summary of developments over the first twenty years given by [12]. Important advances include: the definition of finer projection operators to use in the first phase [20]; the introduction of Partial CAD to make use of the quantified structure of a formula when lifting [13]; the use of equational constraints to reduce the number of projection polynomials required [23]; the use of truth-table-invariant CADs (TTICADs) to apply equational constraint techniques more widely [6]; and an alternative approach to projection and lifting where the problem is solved in complex space and then refined to a CAD of real space [11].

C. Original formulation of the problem

In [14] Davenport considered building a CAD to solve the problem of moving a ladder of length 3 through a right-angled corridor of width 1 (as in Figure 1). Denoting the endpoints of the ladder as (x, y) and (w, z) and assuming the outer corner of the corridor is the origin, the formulation provided was

$$\begin{aligned}
&\big[(x-w)^2+(y-z)^2-9=0\big]\\
\wedge\ &\big[[yz\geq 0]\vee[x(y-z)^2+y(w-x)(y-z)\geq 0]\big]\\
\wedge\ &\big[[(y-1)(z-1)\geq 0]\\
&\quad\vee[(x+1)(y-z)^2+(y-1)(w-x)(y-z)\geq 0]\big]\\
\wedge\ &\big[[xw\geq 0]\vee\ [y(x-w)^2+x(z-y)(x-w)\geq 0]\big]\\
\wedge\ &\big[[(x+1)(w+1)\geq 0]\\
&\quad\vee[(y-1)(x-w)^2+(x+1)(z-y)(x-w)\geq 0]\big]. \quad (1)
\end{aligned}$$

The first equation in (1) describes the length of the ladder, and the remaining inequalities describe the valid positions, ensuring the ladder does not intersect any of the four walls.

In [14] the author completed the projection phase of Collin's CAD algorithm, finding over 250 distinct univariate projection factors with total degree as high as 26. The technology available for the paper did not allow for the simultaneous root isolation of these. With current hardware[1] and software incorporating the latest CAD theory (QEPCAD-B 1.69 [8] and MAPLE 16 [11]) it still remains outside the realm of computation to complete the construction of the CAD.

D. Other approaches

In robotics, piano mover's problems would typically be tackled using numerical methods to produce paths efficiently at the expense of the possibility of rounding errors. We are concerned with the development of symbolic approach and so do not examine numerical methods in this paper.

[1]Experiments in this paper were run on a Linux desktop with a 3.1Ghz Intel processor and 8.0Gb total memory

In [24] the authors of [25] proposed a separate approach for the piano movers problems restricted to the plane, which did not make use of CAD. This algorithm will typically run more efficiently than the CAD based approach but does not generalise to higher dimensional problems.

Tackling the problem with CAD has been revisited several times in the literature. In [19], the author discussed the problem suggesting the question of traversing the corridor was equivalent to the question as to whether there is a position of the ladder for which both extremities are in the two branches of the corridor. The verbal description of this reformulation seems misleading since a ladder could be positioned as such while still being unable to turn the corner as in Figure 2.

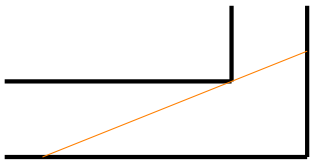

Fig. 2. A configuration of a ladder in which the endpoints are in opposite branches of the corridor.

Later in [19] the author reports on another reformulation using only one endpoint and the tangent of the half-angle between the x-axis and the ladder, reporting that a CAD can be produced for this using Collins' algorithm sufficient to conclude that the problem has no solution. Since no details of the algebraic formulation were provided we are unable to verify this or analyse this formulation further.

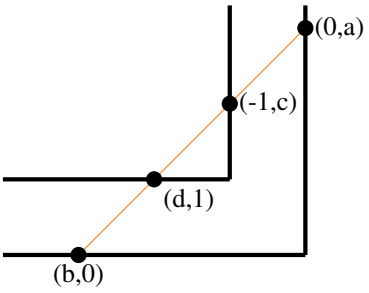

Fig. 3. A configuration of a ladder in which all four walls are intersected.

In [28], Wang uses "simple reasoning" to deduce that the ladder cannot traverse the corridor if and only if it intersects all four walls simultaneously. From this deduction the problem can be reformulated as follows: Let a, b, c, d be coordinates defining the intersection points as in Figure 3 and r be the length of the ladder). Then there is no solution if

$$\begin{aligned}
&(\exists a)(\exists b)(\exists c)(\exists d)[a^2+b^2=r^2\wedge r>0\\
&\quad\wedge a\geq 0\wedge b<0\wedge c\geq 1\wedge d<-1\\
&\quad\wedge c-(1+b)(c-a)=0\wedge d-(1-a)(d-b)=0]. \quad (2)
\end{aligned}$$

Due to its simplicity and the small number of free variables (only r is unquantified) QEPCAD can almost instantly deduce that the maximal length of the ladder is $\sqrt{8}$, using a CAD of 19 cells. When considering the same problem in [27] Wang noted that if the ladder intersected the outer walls and one of the inner walls then it must also intersect the other. Hence equation (2) could be simplified further by removing the final conditions

on lines 2 and 3. This further *topological reasoning* actually makes no difference here (QEPCAD's timings and cell counts are unchanged) but could be powerful for other problems.

In [22] McCallum approaches path-finding by considering transformations of objects by a translation (x, y) and a rotation θ. This produces a formulation of the ladder problem involving 21 equations and inequalities in a comparatively complicated boolean formula. Appealing to equational constraints and partial CAD techniques McCallum constructs a four-dimensional CAD of 16,138 cells in 429 seconds.

In [31] Yang and Zeng considered the problem in the case of a rectangular piano instead of a ladder and used geometric analysis to achieve a simple condition for the problem to have a solution. They parametrize the problem according to the position of a corner and the angle the rectangle makes with the horizontal axis. Through some highly non-trivial analysis they obtain a condition on a polynomial which, if true, implies the existence of a valid route. Applying their techniques to the case of the ladder of length L we see that the existence of a valid route is equivalent to the truth of

$$(\forall x)\, 4x^8 - 4(L-3)x^6 - 2(3L-6)x^4 - 2(L-3)x^2 + 1 > 0. \quad (3)$$

It takes QEPCAD just 1.936 seconds (mostly initialization time) and 5 cells to return: $L^2 - 8 < 0 \vee L < 0$.

The approaches of [28] and [31] are highly efficient but limited. They require, not insignificant, geometric deductions before presentation to CAD, and inform you only whether the ladder can or cannot pass through the corridor, revealing no information about possible paths. It would make sense to therefore use these sort of approaches as an initial test for a problem before constructing an inevitably far more-complicated CAD sufficient for planning routes.

Also, these reformulations give descriptions in the *real space*, meaning they describe the geometry of the plane in which the ladder exists. This is opposed to [25], [14] and the new formulation in Section II which describe the geometry in a four-dimensional *configuration space*, specifically coordinates of the endpoints that fix the ladder within the plane. This distinction is important since the former allows us to analyse whether a ladder can move through the corridor, but cannot provide the explicit path for it to do so. It can be said that [22] also works within a configuration space, however a non-trivial one where positions are encoded by transformations.

By not considering the whole configuration space in their formulations, Wang and Yang–Zeng also cannot consider whether the ladder is able to rotate within the corridor to exit in the opposite orientation (an important point for the generalisations of the problem discussed in Section III-B).

II. NEW FORMULATION OF THE PROBLEM

We consider the problem in configuration space, but from a different perspective than [14]. First we give a formula describing all possible invalid regions, then take its negation as a description of the valid regions. As in (1) we denote the endpoints of the ladder by (x, y) and (w, z).

A. Describing the invalid regions

We describe four canonical invalid configurations for the ladder. Each is identified with an equivalent Tarski formula and examples of each are given in Figure 4.

- A: $x < -1 \wedge y > 1$ or $w < -1 \wedge z > 1$: this describes any collision with the 'inside' walls along with the ladder being on the other side of these.
- B: $x > 0$ or $w > 0$: this describes any collision with the rightmost wall along with the ladder being on the other side.
- C: $y < 0$ or $z < 0$: this describes any collision with the bottommost wall along with the ladder being on the other side.
- D: $(\exists t)[0 < t \wedge t < 1 \wedge x + t(w - x) < -1 \wedge y + t(z - y) > 1]$: this ensures no inner point of the ladder lies in the invalid top-left region.

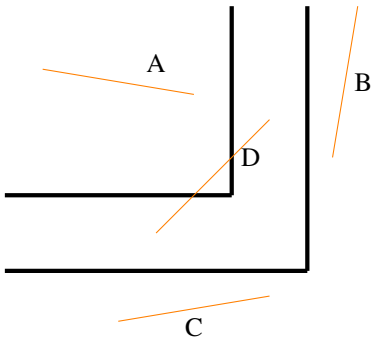

Fig. 4. Four canonical invalid positions of the ladder. Note from the algebraic descriptions that for positions A–C only one end need be outside the corridor.

We can hence characterise the invalid regions with:

$$\begin{aligned}&[x < -1 \wedge y > 1] \vee [w < -1 \wedge z > 1] \vee [x > 0] \\ &\vee [w > 0] \vee [y < 0] \vee [z < 0] \vee (\exists t)\big[0 < t \wedge t < 1 \\ &\qquad \wedge x + t(w - x) < -1 \wedge y + t(z - y) > 1\big]. \end{aligned} \quad (4)$$

This formula contains the "new" variable t used to represent any point on the ladder. We can use QEPCAD to eliminate t from (4) in just over 2 seconds, constructing 681 cells and returning the equivalent quantifier-free formula:

$$\begin{aligned}&[y < 0] \vee [w > 0] \vee [x > 0] \vee [z < 0] \\ &\vee [x + 1 < 0 \wedge y - 1 > 0] \vee [w + 1 < 0 \wedge z - 1 > 0] \\ &\vee [w + 1 < 0 \wedge yw - w + y + x \geq 0 \\ &\qquad \wedge xz + z - yw + w - y - x > 0] \\ &\vee [yw - w + y + x < 0 \wedge z - 1 > 0 \\ &\qquad \wedge xz + z - yw + w - y - x < 0] \\ &\vee [y - 1 > 0 \wedge yw - w + y + x < 0]. \end{aligned} \quad (5)$$

B. New formulation for CAD

We now have a description of the invalid regions, (5), so we can describe the valid regions by taking its negation:

$$\begin{aligned}&[w \leq 0] \wedge [x \leq 0] \wedge [y \geq 0] \wedge [z \geq 0] \wedge [x \geq -1 \vee y \leq 1] \\ &\wedge [w \geq -1 \vee z \leq 1] \wedge \big[wy - w + x + y < 0 \vee w + 1 \geq 0 \\ &\qquad \vee xz + z - yw + w - y - x \leq 0\big] \\ &\wedge \big[yw - w + y + x \geq 0 \vee \big[[z - 1 \leq 0 \\ &\qquad \vee xz + z - yw + w - y - x \geq 0] \wedge y - 1 \leq 0\big]\big]. \end{aligned} \quad (6)$$

Although (6) describes the valid regions in terms of the endpoints it is missing any description of the relationship between these (fixing the length of the ladder). Hence our new formulation of the problem for CAD is

$$[(x-w)^2+(y-z)^2=9] \wedge (6). \tag{7}$$

C. Applying CAD

The formula (7) was given to QEPCAD (with initialisation parameters `+N500000000 +L200000`) under the variable ordering $x \prec y \prec w \prec z$. After a little under 5 hours (16,933.701 seconds) of computation time a CAD of $\mathbb{R}^4$ was constructed with 285,419 cells. The following equivalent formula to (7) was given:

$$\begin{aligned}
&x \le 0 \wedge y \ge 0 \wedge w \le 0 \wedge z \ge 0 \wedge (y-z)^2+(x-w)^2 = 9 \\
&\wedge \Big[[x+1 \ge 0 \wedge w+1 \ge 0] \vee \big[y-1 \le 0 \wedge w+1 \ge 0 \\
&\quad \wedge y^2w^2 - 2yw^2 + x^2w^2 + 2xw^2 + 2w^2 - 2xy^2w \\
&\qquad + 4xyw - 2x^3w - 4x^2w - 4xw + x^2y^2 - 2x^2y \\
&\qquad + x^4 + 2x^3 - 7x^2 - 18x - 9 \ge 0 \big] \\
&\quad \vee \big[x+1 \ge 0 \wedge yw - w + y + x \ge 0 \wedge w^2 - 2xw + y^2 \\
&\qquad - 2y + x^2 - 8 > 0 \wedge z - 1 \le 0 \big] \\
&\quad \vee \big[x+1 \ge 0 \wedge yw - w + y + x \ge 0 \wedge y^2w^2 - 2yw^2 \\
&\qquad + x^2w^2 + 2xw^2 + 2w^2 - 2xy^2w + 4xyw - 2x^3w \\
&\qquad\quad - 4x^2w - 4xw + x^2y^2 - 2x^2y + x^4 + 2x^3 - 7x^2 \\
&\qquad\quad - 18x - 9 \le 0 \wedge z - 1 \le 0 \big] \\
&\quad \vee [y-1 \le 0 \wedge z-1 \le 0] \Big].
\end{aligned} \tag{8}$$

The first line gives the conditions of the problem which are in conjunction with any valid configuration. The remaining lines give a large disjunction of clauses describing such configurations. The first clause is characterizing the positions where the ladder is entirely in the vertical corridor and the last clause where the ladder is entirely in the horizontal corridor. There are then three more clauses characterising positions in between. Any analysis of the decomposition of these equations requires knowledge of the adjacency of the four-dimensional CAD: this is highly non-trivial and discussed in Section II-D.

QEPCAD uses, amongst other theory, partial CAD techniques [13] to simplify its calculations and output. These can be suppressed by issuing the `full-cad` command. We note that doing so greatly increases the difficulty of the problem. Calculating a `full-cad` of (7) resulted in the construction of 1,691,473 cells taking just over a day of computation time (88,238.442 seconds). The quantifier-free formula returned is almost identical to the partial CAD version (8) (with a couple of cases split slightly differently).

We can attempt to speed up the construction by introducing quantifiers on one endpoint leading to a CAD of valid positions for one endpoint of the ladder, by prefixing (7) with $(\exists w)(\exists z)$. Using QEPCAD this took just over 50 minutes (3052.753 seconds) and produced only 5453 cells. The sharp reduction is a result of partial CAD techniques as described in [13]. The resulting quantifier-free formula is simply,

$$x \le 0 \wedge y \ge 0 \wedge [x+1 \ge 0 \vee y-1 \le 0], \tag{9}$$

which is the definition of the original corridor. The quantified version of (7) is simply asking for those points where it is possible to place an end of the ladder and have it in a valid position and so this formula is as expected. We note that the CAD used to construct the formula contains far more information than is needed — a CAD with only 17 cells is sufficient to describe the corridor.

The existential CAD is not sufficient to solve the path finding problem, and for our example the output (9) gives little useful information. However, providing quantified variables has drastically reduced the complexity of the problem and so can be a useful test for the feasibility of the problem (a CAD for the original formulation (1) remains infeasible under quantification). It can also be used in some cases (when the valid region for the endpoint is not the entire corridor) to show a ladder traversal is impossible: for example, if an invalid region were to 'block' the corridor.

QEPCAD can produce a visualisation of two-dimensional CADs through the `p-2d-cad` command. Figure 5 shows the output for the problem in the preceding paragraphs (so it refers to the existential CAD; the diagram for the non-quantified formulation is similar, but omits all cell boundaries within the corridor). The diagram is for x in the range $[-7,2]$ and y in the range $[-2,7]$ with a step of 0.025 (therefore if stacks are within 0.025 (with respect to x) or intra-stack cells are within 0.025 (with respect to y) they will not be distinguishable).

Figure 5 makes clear just how complicated the problem is when being tackled by CAD. There are certainly boundaries to cells that seem to be related to 'boundary cases' of the problem: when the ladder is 'stuck' trying to get around the corner. However, there are many boundaries with little significance for the real problem and so further development of the CAD technology to remove these would be beneficial.

D. Adjacency

The four-dimensional CADs of configuration space we have produced from (7) could be used to both determine the existence of a solution and then construct a path. However, to do the latter we need to first analyse the adjacency and connectedness of cells in the four-dimensional CAD. This is not currently possible with any existing technology and is certainly non-trivial. The process is described in two dimensions by [2] (which has been implemented in QEPCAD) while [3] generalises the approach to three dimensions. Further generalisations are not trivial however [4], with adjacency algorithms likely to work (without a change of coordinates) only for well-behaved input. We also note that in [25] the authors consider adjacencies between n and $(n-1)$-dimensional cells, but since we have an equational constraint, we are actually interested in adjacencies between $(n-1)$ and $(n-2)$-dimensional cells.

E. Choosing a formulation

An important question is why (7) is a better formulation for CAD than (1), and whether we could have predicted this.

On first glance, we see that the new formulation involves polynomials of lesser degree. One measure of CAD complexity is `sotd` (introduced in [16] as the sum of total degree of each monomial in each polynomial). Using this measure applied

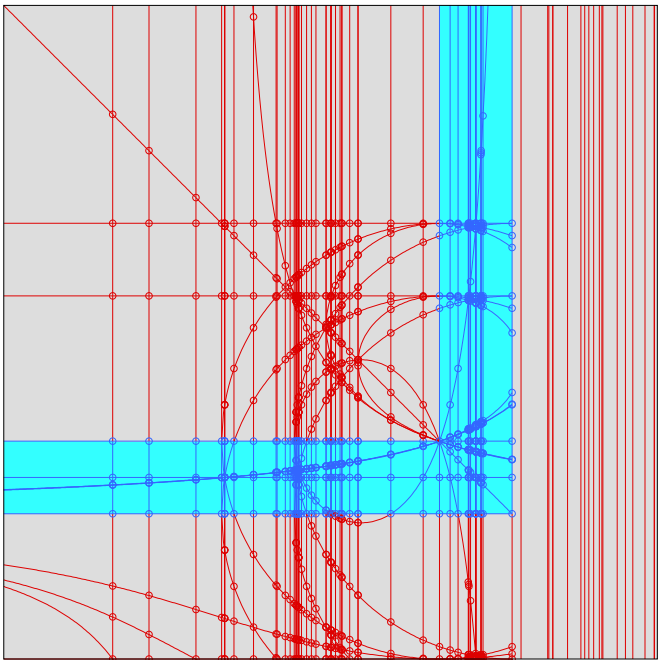

Fig. 5. A two-dimensional CAD of the (x, y) configuration space constructed from (7).

to the input polynomials as a heuristic certainly favours the new formulation: (1) has `sotd` 100 compared to (7) with an `sotd` of 33. The benefit is less obvious when taking an `sotd` of the full projection factor sets. The new formulation is still lower, but there is a smaller relative difference: 2006 is reduced to 1693. There are over 100 univariate polynomials in the projection sets of both formulations. Calculating `ndrr` (introduced in [7] as the number of distinct real roots of the univariate projection polynomials) also favours the new formulation, but again, not by an amount that indicates the changes in feasibility: 367 reduces to 301.

For comparison, we note that the approach by Wang leads to an `sotd` of 19 for the the top level projection polynomials, 98 for the full projection factor set and an `ndrr` of 17. McCallum's formulation has `sotd`'s of 68 and 32 (lower due to repeated factors) and an `ndrr` of 5. Yang-Zeng's approach gives `sotd`'s of 35 and 39, and an `ndrr` of 2. Hence full `sotd` and `ndrr` correctly predict that the CADs related to these approaches will be smaller than our reformulation.

These heuristics do not take into account the number of quantifiers which can be hugely influential in the complexity of a problem. The fact that Wang's formulation contained only a single unquantified variable is hugely instrumental in such an efficient construction. The effect of these quantifiers suggests the creation of more sophisticated heuristics. For example: sum of weighted total degrees. This would weight variables according to two properties: the overall variable ordering and which variables are quantified.

Let the CAD be created with respect to variables $x_1 \prec x_2 \prec \cdots \prec x_n$ where x_1 decomposes $\mathbb{R}^1$, $\{x_1, x_2\}$ decomposes $\mathbb{R}^2$ and so forth. Then assign a weight of i to variable x_i so that the polynomial $x_5^3 - x_1$ would have `sowtd` 16 rather than just an `sotd` of 4. In addition to this, if a variable is quantified then reflect this by halving its effect on `sowtd`. For the above polynomial, if x_5 was quantified then the `sowtd` would become 8.5. Applying these to the various formulations we get the following:

- Davenport (unquantified): `sowtd` $= 148$.
- Davenport (quantified): `sowtd` $= 92$.
- New formulation (unquantified): `sowtd` $= 72$.
- McCallum's formulation: `sowtd` $= 70$.
- New formulation (quantified): `sowtd` $= 46$.
- Wang's formulation: `sowtd` $= 27$.
- Yang–Zeng's formulation: `sowtd` $= 23$.

The `sowtd` measure gives an ordering matching the difference in cell counts, and has plausible-looking differences.

III. Generalising the Problem

A. Ladders of different length

The reformulation described in Section II was for a ladder of length 3. We know already that the maximum length of a ladder able to traverse the corner is $\sqrt{8}$ and similar geometric reasoning shows that the maximum length of a ladder able to reverse its orientation is $\sqrt{2}$. We compare the CAD for (7) (in which the ladder can not traverse the corridor) to the equivalent formulations with a ladder of shorter length. We consider four canonical cases which exhaust the possible scenarios:

Length 3: Ladder cannot traverse the corridor.
Length 2: Ladder can traverse the corridor but is unable to reverse its orientation.
Length $\frac{5}{4}$: Ladder can traverse the corridor and is able to reverse its orientation, but only within the 'corner'.
Length $\frac{3}{4}$: Ladder can traverse the corridor and reverse its orientation at any point within the corridor.

All the results are summarized in Table I. We compare both non-quantified and quantified versions (where the input formula was preceded by $(\exists w)(\exists z)$ as indicated by $\exists$). Note the length of the ladder is an explicit equational constraint and so QEPCAD automatically applies the theory of [23].

TABLE I. CADs of (7) modified by varying ladder length.

	EC-CAD		$\exists$ EC-CAD	
Length	Cells	Time (s)	Cells	Time (s)
3	285419	16286.431	5453	2941.024
2	314541	9863.950	5353	1922.837
5/4	404449	33042.101	5589	7312.347
3/4	446787	13146.195	4347	69.690
3 `full-cad`	1691473	88238.442	—	—

B. Angled corridors

We consider how the problem may be generalised to a non-right angled corridor. There are two canonical cases: that where the angle is obtuse as in Figure 6 and that where the angle is acute as in Figure 7.

For a general obtuse angled corridor the right hand corridor walls have equations $y = \tan(\theta)x$ and $y = \tan(\theta)x + 1$. This results in the following formulation of the invalid positions:

$$
\begin{aligned}
&[x<0\wedge y>1]\vee[y<0]\vee[x>0\wedge y>\tan(\theta)x+1]\\
&\vee[y<\tan(\theta)x]\vee[w<0\wedge z>1]\vee[z<0]\\
&\vee[w>0\wedge z>\tan(\theta)w+1]\vee[z<\tan(\theta)w]\\
&\vee(\exists t)[0<t\wedge t<1]\\
&\quad\wedge\Big[[x+t(w-x)<0\wedge y+t(z-y)>1]\\
&\qquad\vee[y+t(z-y)<0]\vee[x+t(w-x)>0\\
&\qquad\quad\wedge y+t(z-y)>\tan(\theta)(x+t(w-x))+1]\\
&\qquad\vee[y+t(z-y)<\tan(\theta)(x+t(w-x))]\Big].
\end{aligned}
\tag{10}
$$

For a general acute angled corridor the right hand corridor walls have equations $y=-\tan(\psi)x$ and $y=-\tan(\psi)(x+1)$. This results in the following formulation of the invalid positions:

$$
\begin{aligned}
&[y<0]\vee[y>-\tan(\psi)x]\\
&\vee\left[x<-\left(\frac{\tan(\psi)+1}{\tan(\psi)}\right)\wedge y>1\wedge y<-\tan(\psi)(x+1)\right]\\
&\vee[z<0]\vee[z>-\tan(\psi)w]\\
&\vee\left[w<-\left(\frac{\tan(\psi)+1}{\tan(\psi)}\right)\wedge z>1\wedge z<-\tan(\psi)(w+1)\right]\\
&\vee(\exists t)[0<t\wedge t<1]\wedge\Big[x+t(w-x)<-\left(\frac{\tan(\psi)+1}{\tan(\psi)}\right)\\
&\quad\wedge y+t(z-y)>1\\
&\quad\wedge y+t(z-y)<-\tan(\psi)(x+t(w-x)+1)\Big].
\end{aligned}
\tag{11}
$$

If $\tan$ of the angle in question is an algebraic number (for example if the angle is a rational multiple of π) then we can compute an exact solution to these problems using CAD. However for other cases we would either need to approximate the value of $\tan(\theta)$ or treat it as an additional variable in configuration space.

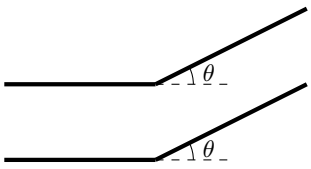

Fig. 6. Generic obtuse angled corridor

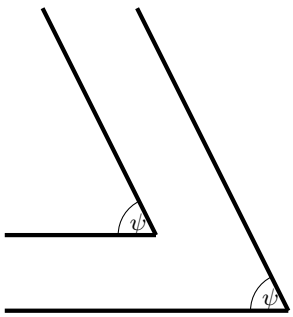

Fig. 7. Generic acute angled corridor

As with the formulation for the right angled corridor we then eliminate the extra parameter t, take the negation of the quantifier free formula, conjunct the equational constraint describing the length of the ladder, and construct a CAD according to this new formula. Hence the new formulation in Section II may be generalised easily, although constructing the CAD may be more computationally difficult. Generalising Wang and Yang–Zeng's methods is not always straightforward due to them being so reliant on geometrical reasoning, as demonstrated in the examples below. It should be possible to adapt [22] for angled corridors, although care may need to be taken that certain trigonometric identities hold.

1) Obtuse $\pi/4$-angled corridor: Let the walls of the right angled corridor make an angle of $\pi/4$ with the horizontal.

We can generalise Wang's idea and consider when the ladder intersects all four walls at once. As with the right-angled corridor, this provides us with the maximal length of the ladder. This approach would work for all obtusely angled corridors (under the same constraint of $\tan(\theta)$ being algebraic).

QEPCAD can answer this question with 27 cells in 5.717 seconds to return

$$r=0\vee 2r^6-93r^4-172r^2-125\geq 0.$$

The appropriate solution is

$$\sqrt{\frac{1}{6}\sqrt[3]{667143+4452\sqrt{159}}+\frac{(2539/2)}{\sqrt[3]{667143+4452\sqrt{159}}}+\frac{31}{2}}$$

which is approximately 6.6786.

Tackling this problem with our method, we first we eliminate t from the invalid regions. This takes 170,597 cells and 230.881 seconds. After forming the complete formulation, QEPCAD fails to construct the relevant CAD after constructing 50,000,000 cells (the self-imposed limit of the `+N50000000` parameter when calling QEPCAD). The extra complexity is because the diagonal corridor is not aligned with the directions of projection.

2) Acute $\pi/4$-angled corridor: Let the walls of the right angled corridor make an angle of $3\pi/4$ with the horizontal to form an acutely angled corridor with angle $\pi/4$.

We can naïvely apply Wang's method to the acutely angled corridor. QEPCAD uses 39 cells in 4.520 seconds to return

$$r=0\vee 2r^6+9r^4-17r^2-125\geq 0.$$

The appropriate solution is

$$\sqrt{\frac{1}{6}\sqrt[3]{4644+249\sqrt{249}}+\frac{61}{2\sqrt[3]{4644+249\sqrt{249}}}-\frac{3}{2}}$$

which is approximately 1.8443.

Unfortunately this does not give a complete answer to the problem. It is possible to fit a rod of length $\sqrt{5}$ (greater than the above value) by placing it within the corner. This disparity is because naïvely applying Wang's idea does not take into account the possibility of reversing the orientation of the ladder necessary for ladders of larger lengths. To adapt Wang's idea to include this reverse orientation would require some non-trivial geometric reasoning.

As our formulation is based within configuration space, it acknowledges the extra condition of orientation, but with added complexity expense. If we try our formulation with $r = 2$ (as $1.8443 < 2 < \sqrt{5}$) we first eliminate t from the invalid regions. This takes 91,583 cells and 86.647 seconds. If we then try to solve the problem by constructing the relevant CAD we fail after constructing 50,000,000 cells (the self-imposed limit of the `+N50000000` parameter when calling QEPCAD).

IV. Adapting CAD technology for future Work

QEPCAD makes use of the theory of equational constraints to reduce the number of projection polynomials (and hence the number of cells in the CAD) along with partial CAD techniques. However, we note that there are further savings that could be made given the presence of the equation and we discuss these ideas and their potential in this section.

A. Extending Equational Constraints

First, as pointed out in [18], the theory of [23] allows us to only lift with respect to the equational constraint for the the final lift. However, QEPCAD appears to lift with respect to all projection polynomials (including the non-equational constraints). Considerable savings can be made by implementing this idea. If more than one equational constraint is present in a problem (for example if there were multiple ladders) then the full power of TTICAD (as described in [6]) can be used to simplify the resulting CAD further.

B. Building a layered CAD for the problem

As mentioned earlier we are concerned with the adjacencies and connectedness of our CAD of configuration space. For this problem we are mainly concerned with those cells in the CAD of full-dimension as these describe regions where the configuration of the ladder is free to move. We note that the key adjacencies for these cells are those through two-dimensional cells: an adjacency of two three-dimensional cells through a one- or zero-dimensional cell would correspond to an infeasible situation in the real physical space for all but boundary cases (i.e. the ladder having to "tightrope walk" a one-dimensional subspace of $\mathbb{R}^2$).

The idea of building CADs containing only cells of full-dimension has been investigated previously in [21], [26], [9]. We have generalised the idea to produce CADs with cells of specified dimension and higher, which we call *layered CADs*. Algorithms to produce these are presented in [30] along with a discussion of their topological properties, possible applications and an implementation in MAPLE built over the authors' `ProjectionCAD` package, [17], [18]. Work on these objects and their properties is ongoing.

C. Lifting to a manifold

In the configuration space, all valid cells must lie on the three-dimensional manifold described by the equation $(x - w)^2 + (y - z)^2 = 9$ so we are only concerned with cells where the equation is satisfied (and the ladder has the desired length). We can therefore construct an order-invariant CAD of three-dimensional space using the projection polynomials for input with an equational constraint [23], and when lifting over this with respect to the equation, discard all sectors. This leaves just the sections: precisely the cells on the manifold. We have implemented this approach using our MAPLE package [18].

Within the manifold, the most important cells are those of full-dimension (with respect to the manifold) as cells of a lower dimension relate to physically infeasible situations (i.e. one-dimensional subspaces of $\mathbb{R}^2$). We can restrict our CAD to produce only these cells through a smarter lifting stage.

Any full-dimensional cell on the three-dimensional manifold must project onto a three-dimensional cell in the induced CAD of $\mathbb{R}^3$ (as it is a section of the equational constraint). We start by constructing the projection set with respect to the equational constraint, producing 11 polynomials in y, w, z. We then build just the full-dimensional CAD cells in $\mathbb{R}^3$: 64,764 cells in 16,991.400 seconds.

We can now lift over these cells with respect to the manifold (an equational constraint). We construct a stack over each cell and extract any sections (those cells lying on the manifold). This process is relatively quick, produces 101,924 cells in 1020.860 seconds. The total time to construct the three-dimensional decomposition of the manifold is therefore 18,012.3 seconds, producing 101,924 cells.

It is not yet feasible to construct all cells on the manifold (or indeed the three-dimensional CAD to lift over) using MAPLE, partly as our implementation does not yet take advantage of partial CAD techniques. We expect that with further improvements a CAD of the manifold sufficient for constructing valid paths (one with two and three-dimensional cells) could be built.

V. Conclusions

We considered a classic example of a piano mover's problem, how a ladder can traverse a corridor, and the solution via CAD. Despite years of improvements to CAD theory and computer hardware, building a CAD for the original formulation in [14] remains infeasible. However, by reformulating the problem CADs can be produced in a matter of seconds, demonstrating how problem formulation is essential to the feasibility of a CAD problem. Further evidence of this was presented in [7].

We presented a new formulation of the problem for which a CAD can be produced. There are other solutions in the literature [28], [22], [31] but these differ in important ways. In [28], [31] the authors relied heavily upon mathematical deduction performed by hand before input to CAD. While this is the most powerful reformulation tool available it is not trivial to automate or generalise. In [22] the author described configurations in a non-trivial manner involving translations and rotations which may complicate subsequent analysis of the space. Our reformulation in Section II uses only a simple negation of the problem, a technique that could be performed algorithmically by CAD technology, using heuristics to decide the appropriate formulation to use.

Another distinction is that the approaches in [28], [31] are firmly rooted within the two-dimensional space of the corridor while the new formulation presented here deals with the configuration space of the ladder: a three-dimensional manifold within four-dimensional space. This means that whilst the

approaches in [28], [31] are able to answer the question "Can the ladder get through the corridor?" they cannot answer the question "How can the ladder get through the corridor?". The approach of [22] would be able to answer the latter question, but only after some non-intuitive analysis of the trigonometric space described by the formulation.

These distinctions may seem trivial for the problem at hand, but they would become far more important in generalisation. Indeed, even for a ladder in an acute-angled corridor it may be that the only feasibility path involves rotating the ladder in the corner (reversing its orientation). This would not be provided by a simple affirmation that a path existed and in the case of Wang's formulation the possibility would not be considered since this formulation requires the orientation to be fixed. It is hard to think of a mathematical argument that takes this into account without needing the full configuration space.

Finally, we have also introduced the idea of restricting lifting in CAD, to cells of full dimension lying on the given manifold. This is much more efficient than producing a full CAD, returning just over a third of the cells for our example. These techniques could be applied to any problem with an equational constraint (lifting to appropriate manifold, or indeed hypersurface) and have now been investigated in generality and formalised in [29].

Although a generic symbolic solution to robot motion planning was provided in theory by [25], in general it remains infeasible to the present day, with numerical methods providing the only practical approach. The ideas presented in this paper show that progress is still possible, but that it will likely follow from more appropriate formulations of problems just as much as advances in theory and technology.

ACKNOWLEDGEMENTS

This work was supported by EPSRC grant: EP/J003247/1.

REFERENCES

[1] D. Arnon, G.E. Collins, and S. McCallum. Cylindrical algebraic decomposition I: The basic algorithm. *SIAM J, Comput.*, 13:865–877, 1984.

[2] D. Arnon, G.E. Collins, and S. McCallum. Cylindrical algebraic decomposition II: An adjacency algorithm for the plane. *SIAM J, Comput.*, 13:878–889, 1984.

[3] D.S. Arnon, G.E. Collins, and S. McCallum. An adjacency algorithm for cylindrical algebraic decompositions of three-dimensional space. *J. Symb. Comput.*, 5(1/2):163–187, 1988.

[4] S. Basu, A. Gabrielov, and N. Vorobjov. Semi-monotone sets. *J. Eur. Math. Soc.*, 15:635–657, 2013.

[5] R. Bradford and J.H. Davenport. Towards better simplification of elementary functions. In *Proc. ISSAC '02*, pages 16–22. ACM, 2002.

[6] R. Bradford, J.H. Davenport, M. England, S. McCallum, and D. Wilson. Cylindrical algebraic decompositions for boolean combinations. In *Proc. ISSAC '13*, pages 125–132. ACM, 2013.

[7] R. Bradford, M. England, J.H. Davenport, and D. Wilson. Optimising problem formulations for cylindrical algebraic decomposition. In J. Carette, D. Aspinall, C. Lange, P. Sojka and W. Windsteiger, editors, *Intelligent Computer Mathematics*, volume 7961 of *Lecture Notes in Computer Science*, pages 19–34, Springer Berlin, 2013.

[8] C.W. Brown. QEPCAD B: A program for computing with semi-algebraic sets using CADs. *ACM SIGSAM Bulletin*, 37(4):97–108, 2003.

[9] C.W. Brown. Constructing a single open cell in a cylindrical algebraic decomposition. In *Proc. ISSAC '13*, pages 133–140. ACM, 2013.

[10] C.W. Brown and J.H. Davenport. The complexity of quantifier elimination and cylindrical algebraic decomposition. In *Proc. ISSAC '07*, pages 54–60, ACM, 2007.

[11] C. Chen, M. Moreno Maza, B. Xia, and L. Yang. Computing cylindrical algebraic decomposition via triangular decomposition. In *Proc. ISSAC '09*, pages 95–102. ACM, 2009.

[12] G.E. Collins. Quantifier elimination by cylindrical algebraic decomposition – 20 years of progress. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts & Monographs in Symbolic Computation, pages 8–23. Springer-Verlag, 1998.

[13] G.E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *J. Symb. Comput.*, 12:299–328, 1991.

[14] J.H. Davenport. A "Piano-Movers" Problem. *SIGSAM Bull.*, 20(1-2):15–17, 1986.

[15] J.H. Davenport, R. Bradford, M. England, and D. Wilson. Program verification in the presence of complex numbers, functions with branch cuts etc. In *14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, SYNASC 2012, pages 83–88. IEEE, 2012.

[16] A. Dolzmann, A. Seidl, and T. Sturm. Efficient projection orders for CAD. In *Proc. ISSAC '04*, pages 111–118. ACM, 2004.

[17] M. England. An implementation of CAD in Maple utilising McCallum projection. Department of Computer Science Technical Report series 2013-02, University of Bath. Available at `http://opus.bath.ac.uk/33180/`, 2013.

[18] M. England. An implementation of CAD in Maple utilising problem formulation, equational constraints and truth-table invariance. Department of Computer Science Technical Report series 2013-04, University of Bath. Available at `http://opus.bath.ac.uk/35636/`, 2013.

[19] J. Marchand. The algorithm by Schwartz, Sharir and Collins on the piano mover's problem. In J.-D. Boissonnat and J.-P. Laumond, editors, *Geometry and Robotics*, volume 391 of *Lecture Notes in Computer Science*, pages 49–66. Springer, 1989.

[20] S. McCallum. An improved projection operation for cylindrical algebraic decomposition of three-dimensional space. *J. Symb. Comput.*, 5(1-2):141–161, 1988.

[21] S. McCallum. Solving polynomial strict inequalities using cylindrical algebraic decomposition. *The Computer Journal*, 36(5):432–438, 1993.

[22] S. McCallum. A computer algebra approach to path finding in the plane. In J. Harland, editor, *Proc. Computing: The Australasian Theory Symposium (CATS)*, pages 44–50, 1997.

[23] S. McCallum. On projection in CAD-based quantifier elimination with equational constraint. In *Proc. ISSAC '99*, pages 145–149. ACM, 1999.

[24] J.T. Schwartz and M. Sharir. On the "Piano-Movers" Problem: I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on Pure and Applied Mathematics*, 36(3):345–398, 1983.

[25] J.T. Schwartz and M. Sharir. On the "Piano-Movers" Problem: II. General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.

[26] A. Strzeboński. Solving systems of strict polynomial inequalities. *J. Symb. Comput.*, 29(3):471–480, 2000.

[27] D. Wang. Reasoning about geometric problems algebraic methods. DOC Technical Report, Imperial College, Univ. of London, 1991.

[28] D. Wang. Geometry machines: From AI to SMC. In J. Calmet, J.A. Campbell, and J. Pfalzgraf, editors, *Artificial Intelligence and Symbolic Mathematical Computation (AISMC)*, volume 1138 of *Lecture Notes in Computer Science*, pages 213–239. Springer, 1996.

[29] D. Wilson, R. Bradford, J.H. Davenport and M. England. Cylindrical algebraic sub-decompositions. *Submitted for publication*, exp. 2014.

[30] D. Wilson and M. England. Layered cylindrical algebraic decomposition. Department of Computer Science Technical Report series 2013-05, University of Bath. Available at `http://opus.bath.ac.uk/36712/`, 2013.

[31] L. Yang and Z. Zeng. Symbolic solution of a piano movers' problem with four parameters. In H. Hong and D. Wang, editors, *Automated Deduction in Geometry*, volume 3763 of *Lecture Notes in Computer Science*, pages 59–69. Springer Berlin Heidelberg, 2006.