Department of Computer Science

UNIVERSITY OF BATH

Technical Report

The Piano Mover's Problem Reformulated

D. J. Wilson, R. J. Bradford, J. H. Davenport and M. England

Technical Report 2013-03 **June 2013**

ISSN 1740-9497

Contact Address:
Department of Computer Science
University of Bath
Bath, BA2 7AY
United Kingdom
URL: http://www.cs.bath.ac.uk

ISSN 1740-9497

THE PIANO MOVER'S PROBLEM REFORMULATED

D. J. WILSON, R. J. BRADFORD, J. H. DAVENPORT AND M. ENGLAND

Abstract. We revisit the classic problem of moving ladders of various lengths through a right-angled corridor. It has long been known that it is theoretically possible to tackle this problem through cylindrical algebraic decomposition (CAD): the valid positions of the ladder are described through polynomial equations and inequalities, which are then used to create a sign-invariant CAD. However, this formulation is too complex for use with current CAD technology, despite much progress in both CAD theory and hardware.

We present a new formulation of the problem, by first considering the invalid positions of the ladder, negating this formula and using this as input for Qepcad (CAD software). We are then able to construct CADs for various lengths of ladder and analyse these through Qepcad's two-dimensional plots functionality. We speculate on the reason our new formulation is more appropriate for the problem, suggest alternative formulations and discuss future work.

1. Original Formulation

In [Dav86], the author followed the work of [SS83], to provide a formulation of moving a 3 unit ladder through a 1 unit wide right-angled corridor (moving from position 1 to position 2 in Figure 1). Denoting the endpoints of the ladder as (x, y) and (w, z) we have:

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

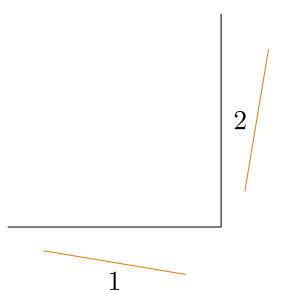

Figure 1. The problem considered in [Dav86]

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. Note, that in [Dav86], (1) was prefaced with existential quantifiers on w and z (but as the technology in [CH91] was not known this had little effect on the problem).

In [Dav86] it was shown, by hand, that inputting (1) to a CAD ([Col75]) would be be infeasible. Even with current hardware[1] and software incorporating the latest CAD technology (Qepcad-B 1.69 and Maple 16) it remains outside the realm of computation.

2. New Formulation

Let us consider the problem from a different perspective. First we describe all possible invalid regions, then take its negation to describe the valid regions. As in (1) We denote the endpoints of the ladder by (x, y) and (w, z).

2.1. Invalid Regions. If we consider the problem we can identify four possible ways for the ladder to not be in a valid configuration. These are shown in Figure 2.

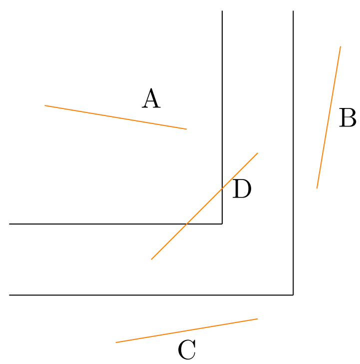

Figure 2. Invalid positions

We can identify each of these with an equivalent Tarski formulae:

- **A:** $x < -1 \wedge y > 1$ or $w < -1 \wedge z > 1$: this includes any collision with the 'inside' two walls along with the ladder being 'outside' the corridor.
- **B:** $x > 0$ or $w > 0$: this includes any collision with the rightmost wall along with the ladder being 'outside' the corridor.
- **C:** $y < 0$ or $z < 0$: this includes any collision with the bottommost wall along with the ladder being 'outside' the corridor.
- **D:** $(\exists t)\,[0 < t \wedge t < 1 \wedge x + t(w - x) < -1 \wedge y + t(z - y) > 1]$: this is the condition that there is a point along the line that is in the invalid top-left region.

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

We can therefore characterise the invalid regions using the following formula:

$$(2)\quad \begin{aligned}&[x<-1\wedge y>1]\vee[w<-1\wedge z>1]\vee[x>0]\vee[w>0]\vee[y<0]\vee\\ &[z<0]\vee(\exists t)\,[0<t\wedge t<1\wedge x+t(w-x)<-1\wedge y+t(z-y)>1]\end{aligned}$$

which (as t is not involved in any formulae outside the quantified one) can be rewritten in prenex form as:

$$(3)\quad \begin{aligned}(\exists t)\Big[&[x<-1\wedge y>1]\vee[w<-1\wedge z>1]\vee[x>0]\vee[w>0]\vee[y<0]\\ &\vee[z<0]\vee[0<t\wedge t<1\wedge x+t(w-x)<-1\wedge y+t(z-y)>1]\Big].\end{aligned}$$

As given, this is not of great use: it contains a spurious variable t and does not dictate that the ladder should have length 3. To deal with the first, we can use QEPCAD to eliminate t from (3). This takes just over 2 seconds (with initialisation), constructs 681 cells and returns the equivalent quantifier-free formula:

$$(4)\quad \begin{aligned}&[y<0]\vee[w>0]\vee[x>0]\vee[z<0]\\ &\vee[x+1<0\wedge y-1>0]\vee[w+1<0\wedge z-1>0]\\ &\vee[w+1<0\wedge yw-w+y+x\geq 0\wedge xz+z-yw+w-y-x>0]\\ &\vee[yw-w+y+x<0\wedge z-1>0\wedge xz+z-yw+w-y-x<0]\\ &\vee[y-1>0\wedge yw-w+y+x<0].\end{aligned}$$

Note that we can use QEPCAD to eliminate the spurious variable t from just the final (quantified) formula in (2) (just the formula describing case D). This takes 1063 cells and 0.2 seconds to calculate and returns an equivalent answer to (4). This is quicker but produces more cells than (3) as it cannot take full advantage of QEPCAD's in-built tools (such as formula simplification).

2.2. **Negation of Invalid Regions.** We now have a description of the invalid regions, (4), so we can describe the valid regions by taking its negation. The following formula represents the negation of (4):

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

2.3. **New Formulation.** We are therefore ready to give our new formulation of the valid regions:

$$(6)\quad \begin{aligned}&[(x-w)^2+(y-z)^2=9]\wedge[w\leq 0]\wedge[x\leq 0]\wedge[y\geq 0]\wedge[z\geq 0]\\ &\wedge[x\geq -1\vee y\leq 1]\wedge[w\geq -1\vee z\leq 1]\\ &\wedge\big[wy-w+x+y<0\vee w+1\geq 0\vee xz+z-yw+w-y-x\leq 0\big]\\ &\wedge\Big[yw-w+y+x\geq 0\vee\big[[z-1\leq 0\vee xz+z-yw+w-y-x\geq 0]\wedge y-1\leq 0\big]\Big]\end{aligned}$$

(which is simply $[(x-w)^2+(y-z)^2=9] \wedge (5)$).

2.4. **Solutions.** We can therefore try to tackle this new formulation using CAD. The formula (6) 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 (16933.701 seconds) of computation time a CAD was constructed with 285419 cells and an equivalent formula to (6) was given:

(7)
$$x \leq 0 \wedge y \geq 0 \wedge w \leq 0 \wedge z \geq 0 \wedge z^2 - 2yz + w^2 - 2xw + y^2 + x^2 - 9 = 0$$
$$\wedge \Big[[x+1 \geq 0 \wedge w+1 \geq 0] \vee \big[y-1 \leq 0 \wedge w+1 \geq 0$$
$$\wedge\, y^2w^2 - 2yw^2 + x^2w^2 + 2xw^2 + 2w^2 - 2xy^2w + 4xyw - 2x^3w - 4x^2w$$
$$-\, 4xw + x^2y^2 - 2x^2y + x^4 + 2x^3 - 7x^2 - 18x - 9 \geq 0\big] \vee \big[x+1 \geq 0$$
$$\wedge\, yw - w + y + x \geq 0 \wedge w^2 - 2xw + y^2 - 2y + x^2 - 8 > 0 \wedge z - 1 \leq 0\big]$$
$$\vee \big[x+1 \geq 0 \wedge yw - w + y + x \geq 0 \wedge y^2w^2 - 2yw^2 + x^2w^2 + 2xw^2 + 2w^2 - 2xy^2w$$
$$+\, 4xyw - 2x^3w - 4x^2w - 4xw + x^2y^2 - 2x^2y + x^4 + 2x^3 - 7x^2 - 18x - 9 \leq 0$$
$$\wedge\, z - 1 \leq 0\big] \vee [y - 1 \leq 0 \wedge z - 1 \leq 0]\Big]$$

The first few QFFs are simply setting up the problem. The interesting part is the final large clause (starting on the second line of (6)). This is a large disjunction of smaller clauses. The first clause is characterizing the positions of with the ladder entirely in the vertical corridor, the last clause is when the ladder is entirely in the horizontal corridor. There are then three more clauses characterising positions in between. These need further analysis to see if we can see whether boundaries between connected components of possible positions can be identified.

We can attempt to speed up the construction by introducing quantifiers on one endpoint: prefixing (6) with $(\exists w)(\exists z)$ (as was done in [Dav86] with (1)). This would give us a CAD of valid positions for one endpoint of the ladder. This took just over 50 minutes (3052.753 seconds) and produced only 5453 cells (this sharp reduction is likely to be a result of partial CAD techniques such as those described in [CH91]). The resulting quantifier-free formula is simple:

$$x \leq 0 \wedge y \geq 0 \wedge [x+1 \geq 0 \vee y-1 \leq 0] \tag{8}$$

This is obviously just the definition of the original corridor - the quantified version of (6) is simply asking for those points where it is possible to place an end of the ladder and have it in a valid position. This is therefore what we should expect and a useful check for any errors in our logic or experimentation.

Qepcad can produce a visualisation of two-dimensional CADs through the `p-2d-cad` command. Figure 3 shows the output for the existential problem given in the preceding paragraphs. 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).

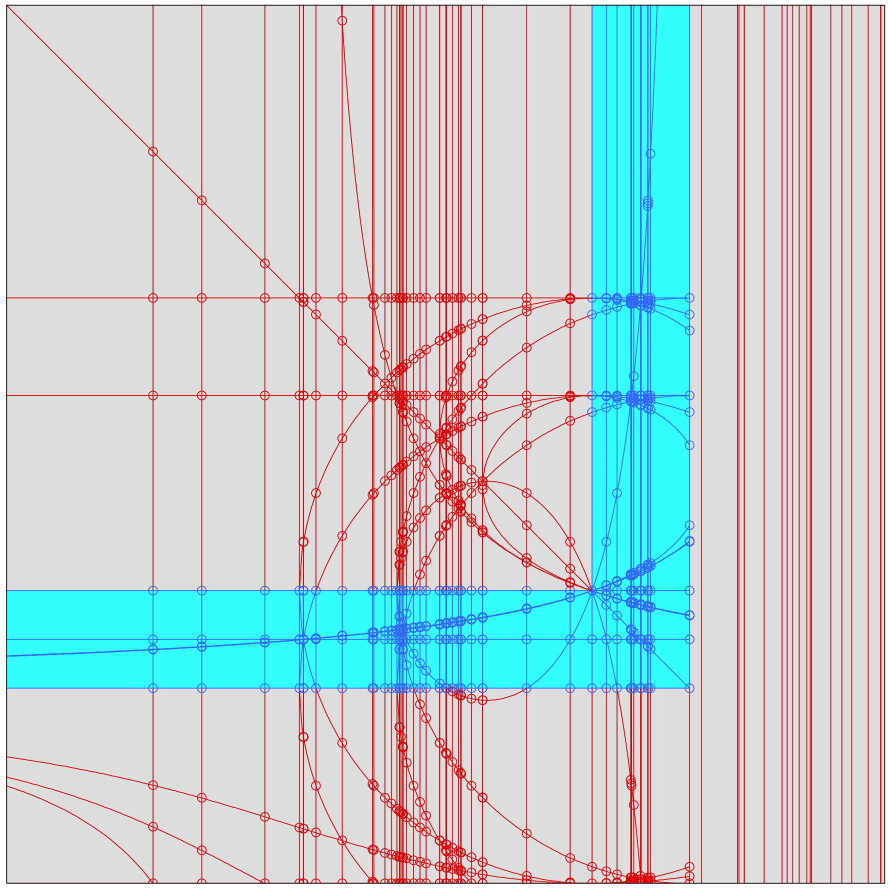

FIGURE 3. Existential CAD for (6)

Figure 3 shows clearly just how complicated the problem is when being tackled by CAD. There are certainly boundaries to cells that would seem to be related to 'boundary cases' of the problem: when the ladder is 'stuck' trying to get around the corner.

It is important to realise we haven't solved the problem yet. To truly 'solve' whether the ladder can traverse the corridor the adjacency and connectedness of cells in the four-dimensional CAD needs to be analysed. This is, obviously, highly non-trivial and needs investigated further.

3. Further results

More tests were run with QEPCAD. All the results are summarized in Tables 1 and 2 and run with the same initialization parameters: `+N500000000 +L200000`. Figures for the existential CADs are given in Appendix A.

3.1. **Suppressing partial CAD.** QEPCAD uses, amongst other theory, partial CAD techniques ([CH91]) and equational constraints ([McC99]) to simplify its calculations and output. These can be (at least partially) suppressed by issuing the `full-cad` command. Doing so greatly increases the difficulty of the problem.

Calculating a `full-cad` of (6) resulted in the construction of 1691473 cells taking just over a day of computation time (88238.442 seconds). The quantifier-free formula returned is essentially identical to the partial CAD version (7) with a couple of cases split slightly differently:

(9)

$$x \leq 0 \wedge y \geq 0 \wedge w \leq 0 \wedge z \geq 0 \wedge z^2 - 2yz + w^2 - 2xw + y^2 + x^2 - 9 = 0$$
$$\wedge \Big[[x+1 \geq 0 \wedge w+1 \geq 0] \vee \big[y - 1 \leq 0 \wedge w + 1 \geq 0 \wedge y^2w^2 - 2yw^2 + x^2w^2 + 2xw^2 + 2w^2 - 2xy^2w + 4xyw - 2x^3w - 4x^2w - 4xw + x^2y^2 - 2x^2y + x^4 + 2x^3 - 7x^2 - 18x - 9 \geq 0\big] \vee [w+1 < 0 \wedge yw - w + y + x = 0 \wedge z = 0] \vee \big[yw - w + y + x > 0 \wedge y^2w^2 - 2yw^2 + x^2w^2 + 2xw^2 + 2w^2 - 2xy^2w + 4xyw - 2x^3w - 4x^2w - 4xw + x^2y^2 - 2x^2y + x^4 + 2x^3 - 7x^2 - 18x - 9 \leq 0 \wedge z - 1 \leq 0\big]$$
$$\vee [y - 1 \leq 0 \wedge z - 1 \leq 0] \vee [w^2 - 2xw + y^2 - 2y + x^2 - 8 > 0 \wedge z - 1 \leq 0] \Big]$$

3.2. **Shorter Ladder.** We know, from basic geometry, that the maximum length of a ladder able to traverse the corner is $\sqrt{8}$ and the maximum length of a ladder able to rotate its orientation is $\sqrt{2}$. It would be of interest to see how (6) (which can not traverse the corridor) compares to ladders of a shorter length that can traverse the corridor and possibly reverse its orientation. We consider four 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.

We have already consider the first case so look at the following three cases.

3.2.1. *Length 2.* This length can navigate the corridor but is unable to reverse its orientation.

The 4-dimensional CAD produced has 314541 cells and is constructed in just under three hours (10231.070 seconds). The equivalent quantifier-free formula produced is:

(10)

$$x \le 0 \wedge y \ge 0 \wedge w \le 0 \wedge z \ge 0 \wedge z^2 - 2yz + w^2 - 2xw + y^2 + x^2 - 4 = 0 \wedge \Big[[x+1 \ge 0 \wedge w+1 \ge 0] \vee \big[y-1 \le 0 \wedge w+1 \ge 0 \wedge y^2w^2 - 2yw^2 + x^2w^2 + 2xw^2 + 2w^2 - 2xy^2w + 4xyw - 2x^3w - 4x^2w - 4xw + x^2y^2 - 2x^2y + x^4 + 2x^3 - 2x^2 - 8x - 4 \ge 0\big] \vee \big[x+1 \ge 0 \wedge yw - w + y + x \ge 0 \wedge w^2 - 2xw + y^2 - 2y + x^2 - 3 > 0 \wedge z - 1 \le 0\big] \vee \big[x+1 \ge 0 \wedge yw - w + y + x \ge 0 \wedge y^2w^2 - 2yw^2 + x^2w^2 + 2xw^2 + 2w^2 - 2xy^2w + 4xyw - 2x^3w - 4x^2w - 4xw + x^2y^2 - 2x^2y + x^4 + 2x^3 - 2x^2 - 8x - 4 \le 0 \wedge z - 1 \le 0\big] \vee [y-1 \le 0 \wedge z-1 \le 0]\Big]$$

Adding quantifiers to tackle the existential question for an endpoint created 5353 cells in slightly over 30 minutes (1997.280 seconds) to produce (8).

3.2.2. *Length* $\frac{5}{4}$. This length can navigate the corridor and is able to reverse its orientation (but only via turning around in the 'corner').

The four-dimensional CAD created contains 404449 cells and was produced in around 9.5 hours (34288.130 seconds). The quantifier-free formulae outputted is:

(11)

$$x \le 0 \wedge y \ge 0 \wedge w \le 0 \wedge z \ge 0 \wedge 16z^2 - 32yz + 16w^2 - 32xw + 16y^2 + 16x^2 - 25 = 0 \wedge \Big[[x+1 \ge 0 \wedge w+1 \ge 0] \vee [y-1 \le 0 \wedge w+1 \ge 0 \wedge 16y^2w^2 - 32yw^2 + 16x^2w^2 + 32xw^2 + 32w^2 - 32xy^2w + 64xyw - 32x^3w - 64x^2w - 64xw + 16x^2y^2 - 32x^2y + 16x^4 + 32x^3 + 7x^2 - 50x - 25 \ge 0] \vee [x+1 \ge 0 \wedge yw - w + y + x \ge 0 \wedge 16w^2 - 32xw + 16y^2 - 32y + 16x^2 - 9 > 0 \wedge z - 1 \le 0] \vee [x+1 \ge 0 \wedge yw - w + y + x \ge 0 \wedge 16y^2w^2 - 32yw^2 + 16x^2w^2 + 32xw^2 + 32w^2 - 32xy^2w + 64xyw - 32x^3w - 64x^2w - 64xw + 16x^2y^2 - 32x^2y + 16x^4 + 32x^3 + 7x^2 - 50x - 25 \le 0 \wedge z - 1 \le 0] \vee [y-1 \le 0 \wedge z-1 \le 0]\Big]$$

Adding quantifiers to ask the existential question for a single endpoint created 5589 cells in 2 hours (7559.598 seconds) producing the formula (8).

3.2.3. *Length* $\frac{3}{4}$. This length is able to navigate the corridor and change its orientation at any moment.

	CAD		EC-CAD	
Length	Cells	Time (s)	Cells	Time (s)
3	285419	16933.701	285419	16286.431
2	314541	10231.070	314541	9863.950
5/4	404449	34288.130	404449	33042.101
3/4	446787	13652.885	446787	13146.195
3 `full-cad`	1691473	88238.442	—	—

TABLE 1. Results for solving (6) with varying lengths. EC indicates that the equational constraint was explicitly stated.

	∃ CAD		∃ EC-CAD	
Length	Cells	Time (s)	Cells	Time (s)
3	5453	3052.753	5453	2941.024
2	5353	1997.280	5353	1922.837
5/4	5589	7559.598	5589	7312.347
3/4	4347	72.282	4347	69.690

TABLE 2. Results for solving the existential version (input formula was preceeded by $(\exists w)(\exists z)$) of (6) with varying lengths. EC indicates that the equational constraint was explicitly stated .

The four-dimensional CAD created contains 446787 cells and takes just under four hours (13652.885 seconds) to produce. The quantifier-free formula produced is as follows:

$$
\begin{aligned}
(12)\quad & x \le 0 \wedge y \ge 0 \wedge w \le 0 \wedge 4w - 4x + 3 \ge 0 \wedge z \ge 0 \\
& \wedge 16z^2 - 32yz + 16w^2 - 32xw + 16y^2 + 16x^2 - 9 = 0 \wedge \Big[[x+1 \ge 0 \wedge w+1 \ge 0] \\
& \vee \big[y - 1 \le 0 \wedge w + 1 \ge 0 \wedge 16y^2w^2 - 32yw^2 + 16x^2w^2 + 32xw^2 + 32w^2 - \\
& 32xy^2w + 64xyw - 32x^3w - 64x^2w - 64xw + 16x^2y^2 - 32x^2y + 16x^4 + 32x^3 + \\
& 23x^2 - 18x - 9 \ge 0\big] \vee \big[32x^2 + 32x + 7 < 0 \wedge 16w^2 - 32xw + 16y^2 - 32y + 16x^2 + 7 > 0 \\
& \wedge z - 1 \le 0\big] \vee \big[x + 1 \ge 0 \wedge 16y^2w^2 - 32yw^2 + 16x^2w^2 + 32xw^2 + 32w^2 - \\
& 32xy^2w + 64xyw - 32x^3w - 64x^2w - 64xw + 16x^2y^2 - 32x^2y + 16x^4 + \\
& 32x^3 + 23x^2 - 18x - 9 \le 0 \wedge z - 1 \le 0\big] \vee [y - 1 \le 0 \wedge z - 1 \le 0] \Big]
\end{aligned}
$$

Adding quantifiers to answer the existential question for a single endpoint created 4347 cells in just over a minute (72.282 seconds), again producing (8) as the output.

3.3. **Equational Constraints.** For all lengths, the length of the ladder is an equational constraint. This can be explicitly stated to QEPCAD which allows it to apply the theory of [McC99].

The results are given in Tables 1 and 2 under the columns labelled EC-CAD and ∃ EC-CAD. As can be seen the number of cells does not change when the equational constraint is declared, although there is a slight speed up in time. This is likely due to the fact that Qepcad automatically identifies and uses equational constraints when possible, and explicitly declaring them skips this step.

4. Layered CADs

We note that the CADs produced by Qepcad both with and without quantified variables were unwieldy. It would therefore be good to attempt recent work on creating only cells of certain dimension (building on [McC93] and [Str00]) to obtain only those full-dimensional cells.

Unfortunately due to a small bug in the `RegularChains` package (a subfunction uses lists, with a fixed maximum length, rather than tables) we are unable to compute such a CAD. This is not a problem with the theory, but rather an implementation bug that is beyond our control.

5. TTICAD

It is possible to recast (6) into disjunctive normal form for input into TTICAD ([BDE+13]). Na ively this produces 24 QFFs with over 8 equations and inequalities in each. However, due to the single equational constraint we can form one large QFF to emulate equational constraints. Unfortunately we are unable to test this due to the same bug preventing us from constructing a CAD in Section 4.

6. Heuristics

An obvious question is why (6) is a better formulation than (1), and whether we could have predicted its greater efficiency.

On first glance, the new formulation involves polynomials of lesser degree. Indeed, taking `sotd` (introduced in [DSS04]) of the inputs favours the new formulation: (1) has `sotd` 100 compared to (6) with an `sotd` of 33.

This effect is less obvious when taking an `sotd` of the full projection sets. The new formulation is still lower, but there is a smaller difference: 2006 is reduced to 1693.

There are over 100 univariate polynomials in the projection sets of both formulations so calculating `ndrr` (introduced in [BDEW13] which considers formulation of CAD problems in a more general setting) directly is costly. If we calculate the `ndrr` of each polynomial separately (possibly counting roots repeatedly) we get a relatively small difference that still indicates the new formulation as better: 367 reduces to 301.

7. Conclusions

The formulation of this simple "piano mover's problem" given in [Dav86] and derived from [SS83] is still intractable after 25 years of improvement in hardware and software. However, our alternative formulation is computable within reasonable time and memory constraints.

Our method may not be suitable for all applications. A geometric argument was needed to work out the invalid regions so it is not clear how useful this will be in general situations. However, it is perhaps an indication that although the methods of [SS83] can be applicable in a wide range of problems it can be beneficial to look for alternative formulations (even if only applicable in special cases).

There are similarities in some respects to a reformulation of the Joukowski transformation that Christopher Brown described in a personal communication: he also considered a negation of the problem as it was in a better form for QEPCAD.

There are other methods to tackle this particular problem. For example, the conformal map mapping $w \mapsto z$ with

$$w = \frac{2}{\pi}\left(\tanh^{-1}\left(\sqrt{z}\right) - \tan^{-1}\left(\sqrt{z}\right)\right) \tag{13}$$

maps a rotated version of this corridor (defined by lines $y = 0 \wedge x \geq 0$, $y = 1 \wedge x \geq -1$, $x = 0 \wedge y \leq 0$ and $x = -1 \wedge y \leq 1$) to the upper half plane. It may be worth considering this as a way to transform our equation. Obviously this would introduce some rather tricky non-polynomial expressions, but they may be able to estimated in a similar way to METITARSKI.

We can also parametrize the ladder in a different manner: we could use (x, y) as a designated endpoint, then use θ to denote the angle of the ladder with respect to the positive x-axis. This would introduce sin and cos terms but these could be eliminated polynomial estimates (as in METITARSKI) or by introducing new variables s and c for $\sin(\theta)$ and $\cos(\theta)$ with the defining equation $s^2 + c^2 = 1$.

This problem certainly deserves further attention, and the ideas given here need to be probed to see if they can provide a general framework for tackling problems in robot motion planning through CAD. It is, however, evidence that reformulating a problem can be more beneficial than advances in software or hardware.

Appendix A. Figures

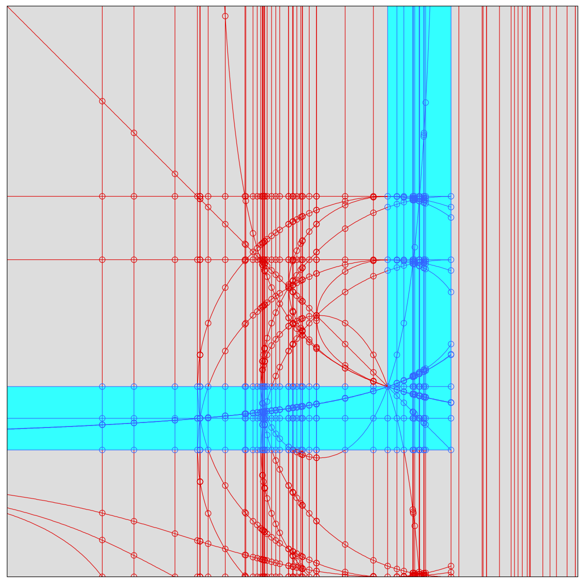

Figure 4. Length 3, existential CAD

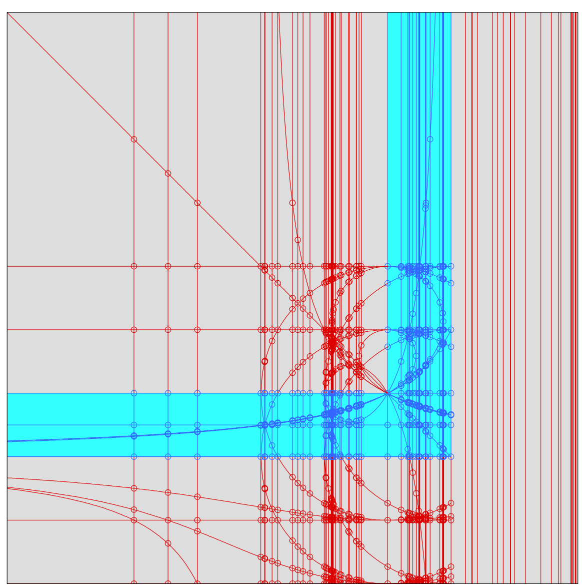

Figure 5. Length 2, existential CAD

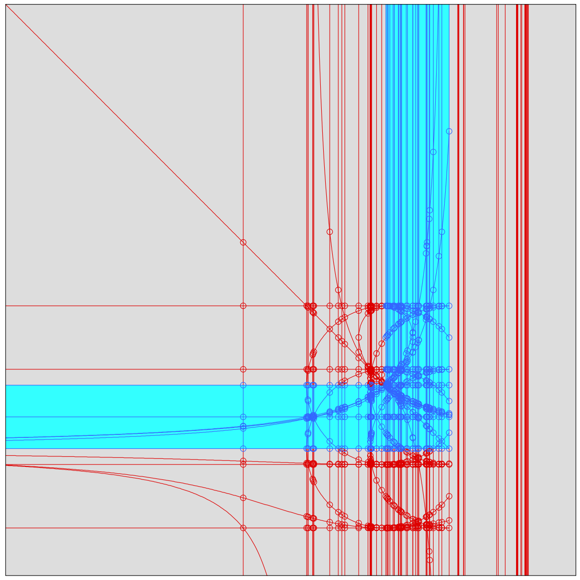

Figure 6. Length 5/4, existential CAD

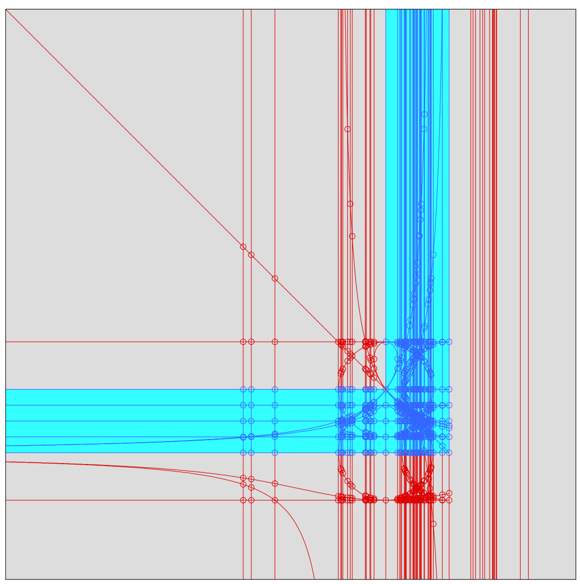

Figure 7. Length 3/4, existential CAD

References

[BDE+13] Russell J Bradford, James H Davenport, Matthew England, Scott McCallum, and David J Wilson, *Cylindrical algebraic decompositions for boolean combinations*, Proc. ISSAC '13, 2013.

[BDEW13] Russell Bradford, James Davenport, Matthew England, and David Wilson, *Optimising problem formulation for cylindrical algebraic decomposition*, Conferences on Intelligent Computer Mathematics: CICM 2013, 2013.

[CH91] George E Collins and Hoon Hong, *Partial Cylindrical Algebraic Decomposition for quantifier elimination*, Journal of Symbolic Computation **12** (1991), no. 3, 299–328.

[Col75] George E Collins, *Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition*, Automata Theory and Formal Languages 2nd GI ... (1975).

[Dav86] James H Davenport, *A "Piano Movers" Problem*, ACM SIGSAM Bulletin (1986).

[DSS04] Andreas Dolzmann, Andreas Seidl, and Thomas Sturm, *Efficient Projection Orders for CAD*, Proc. ISSAC '94, ACM Press, 2004, pp. 111–118.

[McC93] Scott McCallum, *Solving Polynomial Strict Inequalities Using Cylindrical Algebraic Decomposition*, The Computer Journal **36** (1993), no. 5, 432–438.

[McC99] ———, *On Projection in CAD-Based Quantifier Elimination with Equational Constraint*, Proc. ISSAC '99, ACM Press, 1999, pp. 145–149.

[SS83] J.T. Schwartz and M. Sharir, *On the piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds*, Advances in applied Mathematics **4** (1983), no. 3, 298–351.

[Str00] Adam Strzeboński, *Solving systems of strict polynomial inequalities*, Journal of Symbolic Computation **29** (2000), no. 3.