Towards a Formalization of the Active Corner Method for Collision Avoidance in PVS

Nishant Kheterpal
nskh@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Jean-Baptiste Jeannin
jeannin@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Abstract

For safety-critical tasks like collision avoidance, formal verification can provide the assurances required to deploy autonomous systems when lives are at stake. Many methods for verifying collision avoidance model a vehicle as a moving point mass, though the real vehicles have non-zero area. Motivated by this gap, our past work proposed a novel algorithm (the *active corner method*) for verifying collision avoidance for polygonal objects moving in the plane, presented a proof of its correctness, and detailed an automated implementation of the algorithm. This paper presents work-in-progress on a certifiable implementation that generates a machine-checkable PVS proof of correctness for the output quantifier-free safe region formulation. This work briefly discusses the proof approach from our original paper, presents a novel approach that leverages simpler geometric intuition, details what we have proven so far in PVS, and lays out our future research goals for this project.

Keywords: formal verification, collision avoidance, PVS

ACM Reference Format:
Nishant Kheterpal and Jean-Baptiste Jeannin. 2022. Towards a Formalization of the Active Corner Method for Collision Avoidance in PVS. In *Proceedings of (FTSCS '22).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

In safety-critical systems such as medical devices, power systems, automated vehicles, and airplanes, formal verification is important to assure correct, safe operation and prevent harm or loss of life. This work-in-progress paper concerns collision avoidance verification, an important task in motion planning for autonomous robotics in both aerospace and ground settings. In particular, we consider verification for collision avoidance when the object in question has non-zero area. Many approaches for collision avoidance reason about point objects, which is unrealistic for robotics applications where hardware systems (drones, wheeled robots, grasping arms) have volume and mass [2, 10, 13].

In our prior work [12], we introduced the *active corner method* for verifying collision avoidance for polygons moving along known planar trajectories and presented a paper proof of its correctness and completeness. The paper also presented an automated implementation of the *active corner method* in Python, which could produce a quantifier-free geometric representation of the safe region given an object and a trajectory. However, there is currently a gap between our paper proof and the implementation; the Python code could contain bugs or incorrectly implement the method.

Here, we present progress towards formalizing our implementation of the *active corner method* for collision avoidance in the Prototype Verification System (PVS), which couples a formal specification language with an interactive theorem prover [14, 15]. PVS has a lengthy history of use in the verification community and includes the NASALib library, which contains formalizations of mathematical concepts in geometry, calculus, and real analysis that are useful building blocks for the proofs we present. The existing formalizations mean that we do not have to redefine or re-prove the properties of trajectory functions and polygons that we rely on in our proof. We seek to automatically generate machine-checkable PVS proofs. Because of the complexity of both our algorithm and original proof, so far we have focused on building up a generalized proof structure, rather than a full PVS reimplementation of the algorithm from our prior work [12]. Our goal is to automatically generate checkable proof objects that can be used to check the correctness of the returned implementation output to bridge the trust gap between our paper proof and Python code. Note that we are not certifying our Python code; instead we seek to generate certificates of correctness. Along with the PVS work presented here, we envision modifying our Python implementation to return 1) a computationally efficient way to test for collision avoidance and 2) a proof certificate as described above.

In doing so, we also have derived a novel proof approach that holds for simple examples, presented below in Section 3. We are currently working on generalizing this proof approach in two ways: 1) generalizing to all convex polygons and 2) considering functions with bounded derivatives over a certain interval, as in the original proof from our prior work

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
FTSCS '22, December 7, 2022, Auckland, NZ
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/XXXXXXX.XXXXXXX

[12]. The original proof in [12] leveraged significant geometric intuition, such as slopes corresponding to the sides of polygons and bounds on the x- and y-coordinates of interior points of polygons. These geometric, visual concepts are easily understood but not well suited for formalization in a theorem prover; this work presents a more algebraic approach that is more easily expressed in an interactive theorem prover.

PVS has been used to verify properties of hybrid systems in the past [1, 16] along with tools like Isabelle [8]. Our work in generating checkable proof certificates is reminiscent of the Proof Module in CVC [4, 11]. PVS itself can export externally-checkable proof certificates [9]. Other work in certificate checking includes finite-precision error bounds in Coq and HOL4 [5] and proof witnesses for SMT solving in Coq [3].

2 Overview

An intuitive formulation for collision avoidance along a path is a *quantified* one: for all points $(x_\mathcal{T}, y_\mathcal{T})$ along a path $\mathcal{T}$, ensure that an obstacle at coordinate (x_O, y_O) is not inside of a volume v centered at $(x_\mathcal{T}, y_\mathcal{T})$. That is,

$$\forall (x_\mathcal{T}, y_\mathcal{T}) \in \mathcal{T}, (x_O, y_O) \notin v(x_\mathcal{T}, y_\mathcal{T}) \tag{1}$$

However, this quantified representation has one issue: the $\forall$ quantifier cannot be easily used at runtime. Simulation-based approaches may rely on imperfect discretization, and the Cylindrical Algebraic Decomposition algorithm for eliminating quantifiers is doubly exponential in the number of total variables [6, 7]. As such, we need to find a way to eliminate the universal $\forall$ quantifier; certifying such an approach in PVS is the focus of this work. Our prior work introduced a fully symbolic method of generating a quantifier-free equivalent to a quantified statement of collision avoidance in the form of Equation (1). We call this an *explicit* representation of the "safe region", or the set of obstacle locations where, given a polygon's trajectory, a collision will not occur.

The key idea of our active corner method as presented in our past work is that, to construct verifiably correct "safe regions" (or reachable sets) for polygonal objects moving in the plane, we must consider the shape of the object in addition to the paths of its corners [12]. Failing to consider the shape of the object at a finite number of locations (and only following the paths of its corners) would miss the "notch", as illustrated in Figure 1. In particular, we have shown that while the derivative of the trajectory remains in a certain interval, it suffices to follow only two "active" corners.

Because the object moves without rotating in our formulation, we can shift a (known) trajectory for the object center to get equations of motion for the active corners. For a vertex v_i with relative-to-center coordinates $(\Delta x_i, \Delta y_i)$, we can express its path as $y = f(x + \Delta x_i) + \Delta y_i$. Going forward, we will consider only symmetric polygons for simplicity, but our method holds for asymmetric polygons as detailed in prior work [12]. To test if an obstacle at (x_O, y_O) is unsafe, we can

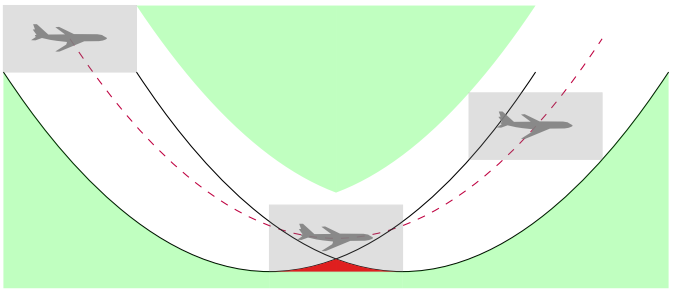

Figure 1. A rectangular airplane moving along a planar trajectory. At the transition point at the parabola's vertex, the "notch" is visible and shaded in red; part of the object lies outside of the corner-trajectories at this point.

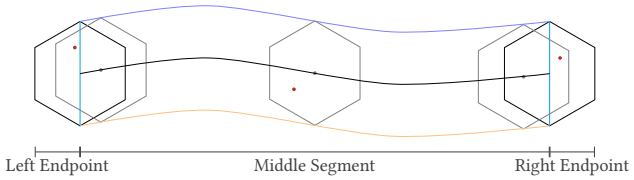

Figure 2. Sections of proof: the three cases of our proof account for the transition points, where the active corners change, at the start and end of each trajectory segment.

see if it lies in between its active corners, which are opposite vertices for symmetric polygons. Correspondingly, an *unsafe* obstacle must be below the path of one active corner and above the path of its opposing corner. We can express this generally as Equation (2), a Boolean test for being *unsafe*.

$$(y_O - f(x_O + \Delta x) - \Delta y)(y_O - f(x_O - \Delta x) + \Delta y) \leq 0 \tag{2}$$

Using Equation (2) is not sufficient; at points along the trajectory where the active corners change, we must account for the "notch" (like the bottom of the parabola in Figure 1). At those discrete instances, called *transition points*, we additionally check if an obstacle is unsafe by seeing if it lies within the polygon at that *transition point*.

3 A Mechanized Proof in PVS

In order to prove the correctness of our method, we must show soundness: that is, any point that the active corner method claims to be safe is indeed safe under the quantified definition (Equation (1)). We can show this by contraposition: we prove that all points $(x_{\text{int}}, y_{\text{int}})$ on the interior of a polygon centered at $(x_\mathcal{T}, y_\mathcal{T})$ for $y_\mathcal{T} = f(x_\mathcal{T})$ will return True (evaluate to *unsafe*) when plugged into Equation (2). A paper proof of the opposite direction of implication (completeness) is included in our prior work [12].

Our original proof of correctness reasons about segments of a trajectory function in which the active corners do not change; that is, segments where the derivative of the function

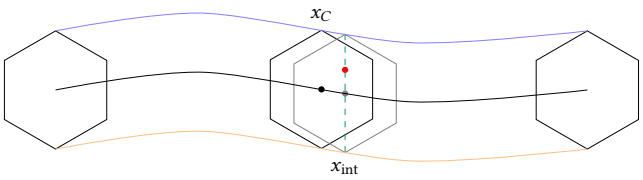

Figure 3. Shifted polygon illustration. The red point is in the interior of the center black polygon but also lies along the dashed green centerline of the grey polygon.

stays bounded within a certain interval. Our proof generalizes to the entirety of a trajectory by reasoning segment-by-segment. As illustrated in Figure 2, our proof considers interior points in three sections: the Left and Right Endpoint sections account for *transition points*, and the Middle Segment in Figure 2 considers interior points lying between the paths of the two active corners, as in Equation (2).

Our original proof from our past work proves completeness by showing that, in the "Middle Segment" of Figure 2 any interior point of a polygon lies along the center line of a different, shifted polygon (Figure 3). By bounding two quantities: 1) the coordinates of that interior point and 2) the center coordinates of that shifted polygon, we can show that the interior point lies above one active corner and below the opposite active corner, and so Equation (2) is True and any interior point will test *unsafe*, as desired.

The remainder of this section focuses on a novel proof approach that reasons algebraically rather than geometrically and is more straightforward to prove in PVS, as no geometry libraries or definitions are required so far. Our proof approach defines the polygon as a conjunction of linear half-plane constraints (inequalities corresponding to edges), and leverages the active corners by combining the inequalities corresponding to the edges adjacent to each corner. We do not yet have a general proof using this more algebraic approach but provide proofs for squares and diamonds moving on straight-line paths in this section. We discuss how this approach may generalize and discuss future work in Section 4.

3.1 Square on straight line

Our initial example for this new proof structure was a square moving in a straight line, as in Figure 4. We defined our trajectory as $f(x) = \alpha x$ for $\alpha \in \mathbb{R}$ and used a square with side length 1, though the algebra generalizes to positive side lengths.

We aim to show any point that the active corner method claims to be safe is indeed safe under the quantified, implicit definition, or $safe_{\text{ACM}} \implies safe_{\forall}$. We express this by contraposition: $unsafe_{\forall} \implies unsafe_{\text{ACM}}$; intuitively this means we want to show that any points inside any polygon on the path will test *unsafe* using the active corner method. This

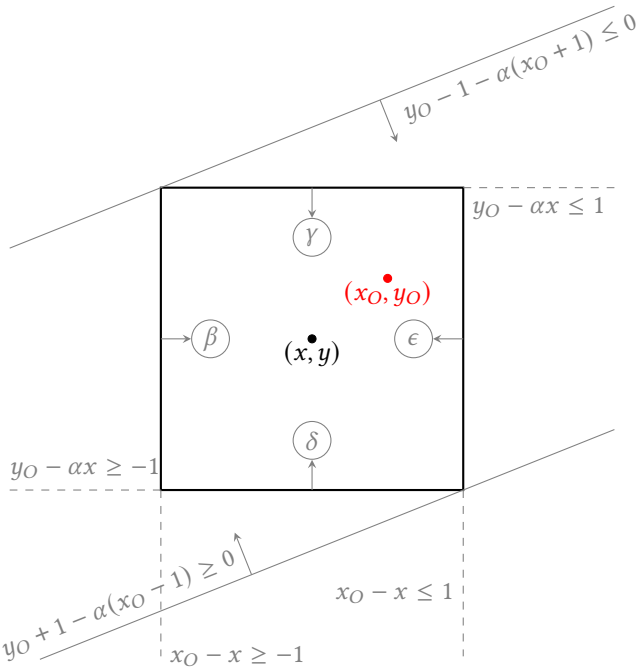

Figure 4. Square moving on line with positive slope α. An obstacle (x_O, y_O) is labeled, along with inequalities defining the sides of the square and the active-corner-trajectories.

leads to the following implication, which we seek to prove:

$$\begin{aligned}
&\exists x \textbf{ s.t. } \big(\underbrace{x_O - x \leq 1}_{\epsilon} \wedge \underbrace{x_O - x \geq -1}_{\beta} \wedge \\
&\underbrace{y_O - \alpha x \leq 1}_{\gamma} \wedge \underbrace{y_O - \alpha x \geq -1}_{\delta}\big) \\
&\implies \\
&((y_O - 1 - \alpha\,(x_O - 1))\,(y_O + 1 - \alpha\,(x_O + 1))) \leq 0 \vee \\
&((y_O - 1 - \alpha\,(x_O + 1))\,(y_O + 1 - \alpha\,(x_O - 1))) \leq 0
\end{aligned} \tag{3}$$

Consider two cases. First we consider $\alpha \geq 0$.

Then we introduce a known constraint γ based on a side of the polygon, as illustrated in Figure 4.

$$\begin{aligned}
y_O - \alpha x &\leq 1 && (\gamma) \\
y_O - 1 \leq \alpha x &\leq \alpha(x_O + 1) && (\text{bound with } \beta) \\
(y_O - 1) - \alpha(x_O + 1) &\leq 0 && (\text{rearrange})
\end{aligned} \tag{4}$$

We combine this by manipulating and substituting into inequality δ, which corresponds to the side opposite γ.

$$\begin{aligned}
y_O - \alpha x &\geq -1 && (\delta) \\
y_O + 1 \geq \alpha x &\geq \alpha(x_O - 1) && (\text{bound with } \epsilon) \\
(y_O + 1) - \alpha(x_O - 1) &\geq 0 && (\text{rearrange})
\end{aligned} \tag{5}$$

The choice of relation introduced in the third line of both Equations (4) and (5) is no coincidence. By pairing constraints

γ and β in Equation (4), we consider where these two constraints intersect; from Figure 4 we can see that is the top left corner of the square, which is an active corner for linear motion with positive slope.

We can use the results of Equations (4) and (5) to conclude that given the left-hand of Equation (3), the following holds:

$$((y_O - 1 - \alpha\,(x_O + 1))\,(y_O + 1 - \alpha\,(x_O - 1))) \leq 0 \quad (6)$$

This is exactly one branch of the disjunction in (3), and so we have proven (3) for $\alpha \geq 0$.

In the case of $\alpha \leq 0$, we consider γ and δ, but pair them with inequalities ϵ and β respectively (the opposite of the above section). The linkage of constraints represents the two active corners; clauses/edges γ and ϵ intersect form the top right active corner for descending motion, and δ and β form the bottom left active corner in this situation.

$$\begin{aligned} y_O - 1 \leq \alpha x \leq \alpha(x_O - 1) \quad & (\gamma, \text{ then } \epsilon) \\ y_O - 1 - \alpha(x_O - 1) \leq 0 & \\ y_O + 1 \geq \alpha x \geq \alpha(x_O + 1) \quad & (\delta, \text{ then } \beta) \\ y_O + 1 - \alpha(x_O + 1) \geq 0 & \end{aligned} \quad (7)$$

From Equation (7), we have that

$$((y_O - 1 - \alpha\,(x_O - 1))\,(y_O + 1 - \alpha\,(x_O + 1))) \leq 0 \quad (8)$$

Again, we have proven one half of the disjunction at the second part of the implication in Equation (3), and so the implication holds for $\alpha \leq 0$.

3.2 Diamond on straight line

Again, we use $f(x) = \alpha x$ as above, but consider instead a diamond moving along this line, illustrated in Figure 5.

Here, we seek to prove a different implication of the same form as Equation (3).

$\exists x$ **s.t.**

$$\begin{aligned} &\Big(\underbrace{y_O - \alpha x \leq (x_o - x) + 1}_{\epsilon} \wedge \underbrace{y_O - \alpha x \geq (x_o - x) - 1}_{\delta} \wedge \\ &\underbrace{y_O - \alpha x \leq -(x_o - x) + 1}_{\gamma} \wedge \underbrace{y_O - \alpha x \geq -(x_o - x) - 1}_{\beta}\Big) \\ &\implies \\ &\left(\underbrace{y_O \geq \alpha x_O - 1}_{\eta} \wedge \underbrace{y_O \leq \alpha x_O + 1}_{\xi}\right) \vee \\ &\left(\underbrace{y_O + \alpha \geq \alpha x_O}_{\theta} \wedge \underbrace{y_O - \alpha \leq \alpha x_O}_{\kappa}\right) \end{aligned} \quad (9)$$

Note that the clauses η, ξ, θ, and κ in the consequent are a bit different, but can be reformulated into the same form as Equation (3) by rearranging terms so each inequality is ≥ 0 or ≤ 0, multiplying together both sides of the $\wedge$ (one of which will be positive and the other negative), and setting that product ≤ 0.

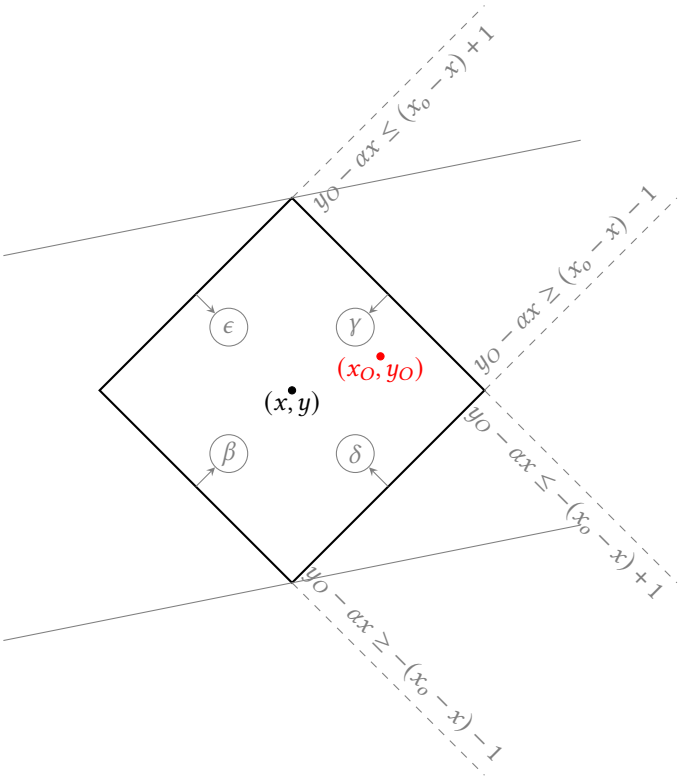

Figure 5. Diamond moving on line with slope α. An obstacle (x_O, y_O) is labeled, along with inequalities defining the sides of the square.

Based on the two pairs of potential active corners for this setting, we consider cases $\alpha \in (-1, 1)$ and $|\alpha| \geq 1$.

First we consider $\alpha \in (-1, 1)$. We must show the following:

$$(\epsilon \wedge \gamma \wedge |\alpha| \leq 1 \implies \xi) \wedge (\beta \wedge \delta \wedge |\alpha| \leq 1 \implies \eta) \quad (10)$$

A proof of the first half of the conjunction in Equation (10) follows; the second part follows similarly. We manipulate inequalities ϵ and γ:

$$\begin{aligned} y_O - \alpha x &\leq (x_O - x) + 1 \qquad (\epsilon) \\ y_O - x_O - 1 &\leq (\alpha - 1)x \\ x &\leq \frac{1}{\alpha - 1}(y_O - x_O - 1) \end{aligned} \quad (11)$$

$$\begin{aligned} y_O - \alpha x &\leq -(x_O - x) + 1 \qquad (\gamma) \\ y_O + x_O - 1 &\leq (\alpha + 1)x \\ x &\geq \frac{1}{\alpha + 1}(y_O + x_O - 1) \end{aligned} \quad (12)$$

Next, given the conclusions of Equations (11) and (12), we eliminate x and bound y_O.

$$\begin{aligned}
\frac{1}{\alpha-1}(y_O - x_O - 1) &\geq \frac{1}{\alpha+1}(y_O + x_O - 1) && \text{(eliminate } x\text{)}\\
(\alpha+1)(y_O - x_O - 1) &\leq (\alpha-1)(y_O + x_O - 1) && \text{(flip inequality)}\\
-\alpha x_O + y_O - 1 &\leq \alpha x_O - y_O + 1 &&\\
y_O &\leq \alpha x_O + 1 && (\xi\text{, shown})
\end{aligned} \tag{13}$$

Note that the inequality switches towards the end of Equation (13) because $\alpha < 1$, and we multiply by $(\alpha+1)(\alpha-1)$, a negative number.

In the case of $|\alpha| \geq 1$, we take a different approach, using the left and right-most corners of the diamond (which are active for motion with slopes above 45 deg and below 45 deg). Correspondingly, to show θ, we use clauses δ and γ. That is, we seek to prove

$$\delta \wedge \gamma \wedge |\alpha| \geq 1 \implies \theta \tag{14}$$

The proof follows.

$$\begin{aligned}
y_O - \alpha x &\geq (x_O - x) - 1 && (\delta)\\
y_O - \alpha x &\leq -(x_O - x) + 1 && (\gamma)\\
y_O - x_O + 1 &\geq (\alpha-1)x && (x(\alpha+1) \geq 0)\\
y_O + x_O - 1 &\leq (\alpha+1)x && (x(\alpha-1) \geq 0)\\
(\alpha+1)(y_O - x_O + 1) &\geq (\alpha-1)(y_O + x_O + 1) &&\\
\alpha(-x_O + 1) + y_O &\geq \alpha(x_O - 1) - y_O &&\\
\alpha(-x_O + 1) + y_O &\geq 0 &&\\
y_O &\geq \alpha x_O - \alpha && (\theta\text{, shown})
\end{aligned} \tag{15}$$

3.3 Implementation in PVS

We have proven the square case in PVS and are working on the diamond case proof. Our approach to proving the square case directly follows the algebraic manipulations in Equations (4) and (5).

Our theorem of soundness for the square case in PVS is below. It effectively expresses Equation (3) in PVS.

```
SoundnessAlpha(xo, yo, alpha: real) : bool =
  (EXISTS (x : real) :
  ((xo - x) <= 1 AND (xo - x) >= -1 AND
   (yo - alpha*x) <= 1 AND (yo - alpha*x) >= -1))
  IMPLIES
  ((yo - alpha*xo - alpha - 1) *
   (yo - alpha*xo + alpha + 1) <= 0) OR
  ((yo - alpha*xo - alpha + 1) *
   (yo - alpha*xo + alpha - 1) <= 0)
```

Our PVS proof for the square-linear case proceeds by splitting cases on the condition $\alpha \geq 0$ and bounding x in terms of x_O. By multiplying the constraints with x and x_O by α, we then apply those bounds to the $y_O - \alpha x$ inequalities and show one portion of the PVS consequent for each case of α (positive and negative).

Similarly, the PVS definition for the diamond case implements Equation (9) in PVS syntax.

```
DiamondSoundness(xo, yo, alpha : real) : bool =
  (EXISTS(x : real) :
   ((yo - alpha*x <=  xo - x + 1) AND
    (yo - alpha*x >=  xo - x - 1) AND
    (yo - alpha*x <= -xo + x + 1) AND
    (yo - alpha*x >= -xo + x - 1)))
  IMPLIES
   ((yo - alpha*xo + alpha) *
    (yo - alpha*xo - alpha) <= 0 OR
    (yo - alpha*xo + 1) *
    (yo - alpha*xo - 1) <= 0)
```

Our PVS proof for the diamond-linear case proceeds by splitting cases on the conditions $-1 < \alpha < 1$ and $|\alpha| >= 1$. We then eliminate x from inequalities representing adjacent sides of the diamond as in 3.2, which yields terms that can be rearranged to upper- or lower-bound y_O. These bounds themselves can be rearranged and multiplies to match one term of the PVS consequent for our two cases on α.

4 Future Work and Conclusion

We have presented work in progress on formalizing the active corner method [12]. Our goal is to generate machine-checkable proof certificates of correctness and provide these to users of our automated implementation, thus bridging the trust gap between our paper proof and our Python implementation. In this paper, we have reviewed the active corner method; discussed two proof approaches: one geometric from our past paper and a novel, more algebraic approach; and presented progress on formalizing this new proof in PVS.

We envision several next steps to generalize our PVS proofs. To consider notches, we must add additional constraints that bound the domain for each proof case (intervals where the active corner does not change). So far, we have proven our square-linear case in PVS for a finite start but not a finite end. To generalize to all convex polygons, we must construct half-plane constraints as in (3) and (9) automatically from a polygon definition in PVS and then manipulate those constraints corresponding to the active corners. Lastly, we must reason about function derivatives rather than linear motion. Our assumption that the active corners do not change make this more straightforward: we can consider the maximum and minimum values of the derivative $f'(x)$ when manipulating inequalities, the way that we use α in our linear cases when bounding terms now.

Acknowledgments

This work was funded by a NASA Fellowship. Thank you to Aaron Dutle and Tanner Slagel for their assistance in developing these proof techniques and the PVS proofs.

References

[1] Erika Ábrahám-Mumm, Ulrich Hannemann, and Martin Steffen. 2001. Verification of hybrid systems: Formalization and proof rules in PVS. In *Proceedings Seventh IEEE International Conference on Engineering of Complex Computer Systems*. IEEE, 48–57.

[2] Matthias Althoff, Goran Frehse, and Antoine Girard. 2021. Set propagation techniques for reachability analysis. *Annual Review of Control, Robotics, and Autonomous Systems* 4 (2021), 369–395.

[3] Michaël Armand, Germain Faure, Benjamin Grégoire, Chantal Keller, Laurent Théry, and Benjamin Werner. 2011. A modular integration of SAT/SMT solvers to Coq through proof witnesses. In *International Conference on Certified Programs and Proofs*. Springer, 135–150.

[4] Haniel Barbosa, Clark W. Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Andrew Reynolds, Ying Sheng, Cesare Tinelli, and Yoni Zohar. 2022. cvc5: A Versatile and Industrial-Strength SMT Solver. In *Tools and Algorithms for the Construction and Analysis of Systems - 28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2-7, 2022, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 13243)*, Dana Fisman and Grigore Rosu (Eds.). Springer, 415–442. https://doi.org/10.1007/978-3-030-99524-9_24

[5] Heiko Becker, Nikita Zyuzin, Raphael Monat, Eva Darulova, Magnus O. Myreen, and Anthony Fox. 2018. A Verified Certificate Checker for Finite-Precision Error Bounds in Coq and HOL4. In *Formal Methods in Computer Aided Design (FMCAD)*. IEEE. https://doi.org/10.23919/FMCAD.2018.8603019

[6] George E. Collins. 1975. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In *Automata Theory and Formal Languages*, H. Brakhage (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 134–183.

[7] James H. Davenport and Joos Heintz. 1988. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation* 5, 1 (1988), 29–35. https://doi.org/10.1016/S0747-7171(88)80004-X

[8] Simon Foster, Jonathan Julián Huerta y Munive, Mario Gleirscher, and Georg Struth. 2021. Hybrid Systems Verification with Isabelle/HOL: Simpler Syntax, Better Models, Faster Proofs. In *International Symposium on Formal Methods*. Springer, 367–386.

[9] Frédéric Gilbert. 2017. Proof certificates in PVS. In *International Conference on Interactive Theorem Proving*. Springer, 262–268.

[10] Antoine Girard. 2005. Reachability of uncertain linear systems using zonotopes. In *International Workshop on Hybrid Systems: Computation and Control*. Springer, 291–305.

[11] Guy Katz, Clark Barrett, Cesare Tinelli, Andrew Reynolds, and Liana Hadarean. 2016. Lazy proofs for DPLL (T)-based SMT solvers. In *2016 Formal Methods in Computer-Aided Design (FMCAD)*. IEEE, 93–100.

[12] Nishant Kheterpal, Elanor Tang, and Jean-Baptiste Jeannin. 2022. Automating Geometric Proofs of Collision Avoidance with Active Corners. In *Proceedings of the 22nd Conference on Formal Methods in Computer-Aided Design – FMCAD 2022*, Alberto Griggio and Neha Rungta (Eds.), Vol. 3. TU Wien Academic Press, 359–368.

[13] Stefan Mitsch, Khalil Ghorbal, and André Platzer. 2013. On provably safe obstacle avoidance for autonomous robotic ground vehicles. In *Robotics: Science and Systems IX, Technische Universität Berlin, Berlin, Germany, June 24-June 28, 2013*.

[14] Sam Owre, John M Rushby, and Natarajan Shankar. 1992. PVS: A prototype verification system. In *International Conference on Automated Deduction*. Springer, 748–752.

[15] Sam Owre and Natarajan Shankar. 2008. A brief overview of PVS. In *International Conference on Theorem Proving in Higher Order Logics*. Springer, 22–27.

[16] J Tanner Slagel, Lauren White, and Aaron Dutle. 2021. Formal verification of semi-algebraic sets and real analytic functions. In *Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs*. 278–290.