10-2019

# Exercises Integrating High School Mathematics with Robot Motion Planning

Ronald I. Greenberg
*Loyola University Chicago*, Rgreen@luc.edu

George K. Thiruvathukal
*Loyola University Chicago*, gthiruv@luc.edu

# Exercises Integrating High School Mathematics with Robot Motion Planning

Ronald I. Greenberg
Department of Computer Science
Loyola University Chicago
Chicago, Illinois 60611–2147
Email: rig@cs.luc.edu

George K. Thiruvathukal
Department of Computer Science
Loyola University Chicago
Chicago, Illinois 60611-2147
Email: gkt@cs.luc.edu

***Abstract*—This Innovative Practice Work in Progress presents progress in developing exercises for high school students incorporating level-appropriate mathematics into robotics activities. We assume mathematical foundations ranging from algebra to precalculus, whereas most prior work on integrating mathematics into robotics uses only very elementary mathematical reasoning or, at the other extreme, is comprised of technical papers or books using calculus and other advanced mathematics. The exercises suggested are relevant to any differerential-drive robot, which is an appropriate model for many different varieties of educational robots. They guide students towards comparing a variety of natural navigational strategies making use of typical movement primitives. The exercises align with Common Core State Standards for Mathematics.**

***Index Terms*—Robotics, Computer science education, Mathematics of Computing.**

## I. Introduction

Providing students with robotics experiences has become a popular and successful mechanism for broadening participation in computing and STEM more generally, retaining more students in these fields, and improving their learning (e.g., [1], [2], [3], [4], [5], [6]). Furthermore, one study of student responses to brief computing outreach visits found that the most popular component of such visits was viewing of robotics videos [7].

In terms of skill development in robotics programs, there is typically a focus on logical thinking, computer programming, and/or engineering design, while this paper takes a different focus. We provide avenues for students participating in robotics programs to be motivated towards substantial and creative mathematical exploration. Conversely, students who enjoy mathematics can find that robotics challenges, incorporating such activities as computer programming, provide a concrete application of their mathematical skills. A wide body of literature supports the effectiveness of motivating study of one subject by using another subject of interest to students or, more generally, tapping into students perceptions of the usefulness of the subject (e.g. [8], [9], [10], [11])

Many mentors are well aware of the potential for integration with mathematics, but there is limited availability of level-appropriate materials geared towards achieving such integration for high school students. Specifically, we assume that the mathematical foundations for these grades ranges from algebra to precalculus. In this paper, we focus on motion planning for robots of the type typically used in educational robotics and attempt to bridge the gap between very elementary treatments and advanced treatments to provide a valuable framework for intermediate-level students to integrate mathematical thinking into planning the movement of their robots.

## II. Technical Background

We will work with a robot drawn as in Fig. 1 that must navigate through a two-dimensional field. With two wheels that may be driven independently (forwards or backwards up to some maximum speed) and a third balance point such as a caster wheel or track ball, such a robot is generally referred to as a differential-drive robot [12, p. 11–12], [13], [14] or a two-driving wheel robot [15, p. 4]. (We do not consider Swedish wheels [12, p. 89] that can roll in all directions or even wheels that turn as in a typical modern automobile.) This model is a good fit for most educational robots, for example:

- the iRobot Create® robot based on the Roomba® vacuum cleaning platform that is used in the Botball® program [16],
- LEGO®-compatible Edison robots (the primary recommended platform for the robotics unit in the Exploring Computer Science (ECS) high school curriculum [17], which has spread to many locations in the United States [18], and is attracting international attention as well)
- Robots built from LEGO pieces and other provided parts in various programs such as the demonstration LEGO robot for Botball
- Finch™ robots that can be borrowed as easily as visiting a public library in some locales [19]

Some of these platforms have a fixed distance $w$ between the wheels, most often referred to as *track width*, but this important parameter can be a point of flexibility in many LEGO-based designs. Intuitively, a small $w$ makes the robot able to move more nimbly, assuming one avoids values that are so small as to lead to problems with rollover.

In the limit, with $w = 0$, we would be essentially dealing with moving a single point around our two-dimensional field, but here we consider practicalities of working with a robot that has the limitation of a positive track width. This alone

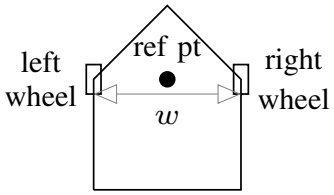


Fig. 1. We will sketch our robot as shown, with a pointed front and a reference point centered between the two independently-driven wheels; there is also an assumed third balance point, for example a caster wheel in the back.

is enough to put the motion of the robot into the category referred to as non-holonomic [12, p. 88–91].

On the other hand, a complication to which we will not give much attention is obstacle avoidance. While there is extensive literature on motion planning in the presence of obstacles, perhaps even with many obstacles and with moving obstacles, we restrict attention to a more elementary scenario that is generally practical in educational robotics. That is, while we discuss the use of navigational primitives that can allow for mild skirting of obstacles, we assume that students can manually preplan paths (missions) comprised of a modest number of trajectories of specific types.

Consistent with this approach, we focus on dead reckoning through a known terrain without substantial use of sensors. Dead reckoning may be defined as the process of calculating one's current position by using a previously determined position and advancing that position based upon known or estimated speeds over elapsed time and course. (In experiments to verify the feasibility of motion planning approaches suggested in this paper [20], we did use a robot with a built-in functionality of testing the amount of rotation of the motors controlling the robot wheels. This makes the results somewhat more robust in the face of possibly varying levels of battery charge, but it may also be possible to achieve passable results using only timing delays as long as the battery is frequently recharged.)

Experienced teams in robotics competitions do tend to make use of other sensors that are typically available, for example a range finder to judge closeness of approach to a landmark or a reflectance sensor to detect black/white boundaries on the driving surface. While such sensors can be extremely useful, students often can achieve passable results without advanced sensors. In addition to providing an opportunity to integrate mathematical and statistical or data-driven reasoning, this approach can be very valuable to both beginning and advanced teams. Beginning teams may not have yet mastered the use of sensors and the requisite programming, while advanced teams may wish to incorporate a fail-safe in case of failure of a sensor, its mount, or its electrical connection. In all cases, dead reckoning is likely to be helpful, even if only for initial rough positioning before using sensors, since this often can be done at higher speed and lower power consumption.

Within this dead reckoning context, there are very simple exercises suitable for students as young as elementary grades; see, for example, this simple guide to compute robot travel distances and rotation angles in terms of numbers of wheel rotations: [21]. (Also mentioned there is a popular exercise of having students drive in a path defined by a regular polygon, including calculation of the appropriate angle for each turn, which can also be found as an exercise in what are essentially simple motion simulation environments like Scratch, e.g., [22].) Many other works have also considered using robotics to motivate mathematical learning, but only at more elementary levels than we consider here and usually with at least as much focus on more general STEM, problem solving and team work skills (e.g. [23], [24], [25], [26], [27], [28], [29] and references therein, of which, the math level begins to approach ours only in [28], [29]).

At the other extreme, one may embark on a calculus-based study of inertial navigation (overviewed in [12, p. 77–80] for example) or use calculus and other heavy mathematics to achieve proofs regarding optimal path types (using various constraints and criteria) for arbitrary changes in position and orientation of a differential-drive robot, e.g. [13], [14], [30], [31] (and even more complex car-like robots, e.g. [32], [33], [34], [35], [36], [37]).

Here, we take an intermediate tack, analyzing and experimenting with some intuitive alternatives for basic navigation tasks, using only algebra and trigonometry. The recent book of Ben-Ari and Mondada [12] is a rare example of explaining robotics concepts with a primary focus on this intermediate level of sophistication; in this paper, we add depth to their elementary discussions of odometry while stopping short of calculus-based discussions. This paper follows on investigations reported in [20] with one particular robot platform, by constructing a general student-focused worksheet integrating trigonometry and algebra with robot motion planning. This aligns especially with the following Common Core State Standards for Mathematics [38]:

• 8.G.B.7: Apply the Pythagorean Theorem to determine unknown side lengths in right triangles in real-world and mathematical problems in two and three dimensions.

• HSG.SRT.C.8: Use trigonometric ratios and the Pythagorean Theorem to solve right triangles in applied problems.

(Also see [20] for discussion of integrating linear regression with robot motion planning, and see [39] for discussion of using intermediate-level mathematics, programming, and data analysis to inform robot building.)

## III. Student Exercises

Here we present a series of exercises for students. Most of what we have discussed so far does not need to be presented to students, but we will begin by referring back to the basic robot model of Figure 1.

### A. Worksheet Introduction

This worksheet focuses on analyzing the time required for a robot to complete an example motion under different schemes of operation that vary in programming difficulty. Since the time varies and other considerations may also apply, the final choice may involve a tradeoff among various considerations.

But we need to establish a formal model and complete some mathematical analyses to have a basis for comparing robot navigation times.

We will work with a robot drawn as in Fig. 1 that must navigate through a two-dimensional field. With two wheels that may be driven independently (forwards or backwards up to some maximum speed) and a third balance point such as a caster wheel or track ball, such a robot is generally referred to as a differential-drive robot. (Unlike wheels in a modern automobile, these wheels do not turn from side to side.)

We will think about moving the reference point on the robot from a starting position at coordinates $(0, 0)$ to a target position $(x, y)$ but keeping in mind that the movement is constrained by the use of our two wheels at separation $w$. We also assume that the robot should point in the same direction at the end of the motion as at the beginning. The distance $w$ between the wheels is most often referred to as *track width*, and it constrains how tightly the robot can turn. Until the very end of the worksheet, we assume the following parameter values. The track width $w$ is 2cm. Each wheel can turn at a maximum speed of 1cm per second, and we assume that acceleration and deceleration is instantaneous. We will use $x = 3.586$ and $y = 4.414$. Compute your answers to questions to 3 decimal places.

Students working with educational robots typically find it convenient to program the robot to make $90°$ turns ($\pi/2$ radians) and to go straight for a specified distance. There are two styles in which turns are typically programmed, either rotations (about the reference point as one wheel moves forward and the other backward), or swings (with one wheel moving so that the robot pivots around the fixed wheel). A more advanced programming approach might allow for turns to various angles. We will begin by considering turns of just $\pi/2$ radians and then consider more general turns.

### B. Horizontal and Vertical Navigation

With all navigation along horizontal and vertical lines with turns of $\pi/2$ radians, the path of the robot will look as in Fig. 2(a) or 2(b), depending whether we use rotations or swings (assuming we choose a "middling" point to make our turns. (If there are obstacles that must be avoided, the point at which the first turn is taken can be adjusted without affecting the navigation time.)

**Exercise 1a** Compute the time under the rotation approach of Fig. 2(a). Note that since both wheels are always in motion, we can compute the time as being proportional to the distance traveled by either wheel. Remember the relationship that the length of a circular arc is the product of the radius and the subtended angle.

**Exercise 1b** Compute the time under the swing approach of Fig. 2(b). (We can mostly consider the distance traveled by either wheel, but when that wheel is stationary, we must account for the distance traveled by the other wheel.)

**Exercise 1c** When working with horizontal and vertical navigation, is the robot motion faster with rotations or swings?

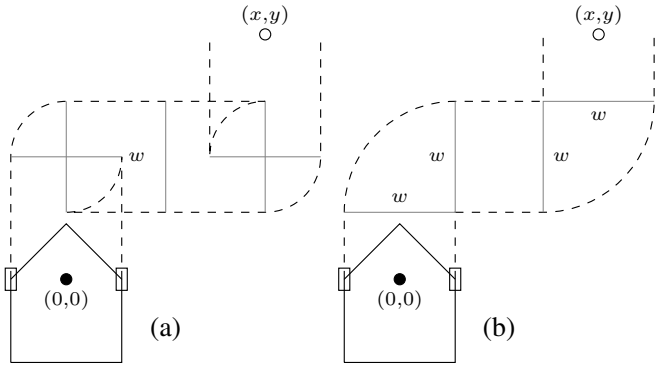


Fig. 2. The paths of the robot wheels with horizontal and vertical navigation, using rotations in (a) and swings in (b).

### C. Generalized Navigation

While the navigational approach of the prior section is simple, we would expect to be able to navigate more quickly by proceeding on a path closer to a straight line.

Figures 3(a) and 3(b) show the paths under the rotation and swing approaches assuming we use the rotations or swings just to line us up for straight-line navigation.

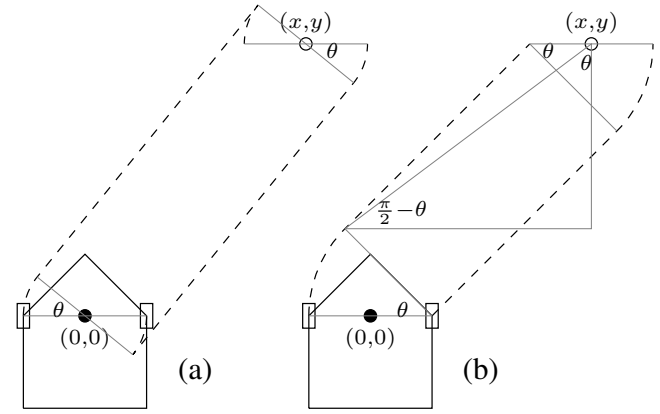


Fig. 3. The paths of the robot wheels under general navigation using rotations in (a) and swings in (b).

**Exercise 2a** Compute the time under the rotation approach of Fig. 3(a). (Since both wheels are always in motion, we can compute the time as being proportional to the distance traveled by either wheel.) You will need to compute the angle $\theta$.

**Exercise 2b** Compute the time under the swing approach of Fig. 3(b). (We can mostly consider the distance traveled by either wheel, but when that wheel is stationary, we must account for the distance traveled by the other wheel.) In this case, the angle $\theta$ is $\pi/4$ radians.

**Exercise 2c** When working with horizontal and vertical navigation, is the robot motion faster with rotations or swings?

### D. Variations on Generalized Navigation

A problem that may often occur in practice with the routing of Fig. 3(a) is that the robot might need to butt up against

a boundary wall at the start and/or end of the path; then there will be insufficient room to perform the rotations. This motivates the following exercises.

Consider a variation on the path of Fig. 3(a) in which the robot runs straight forward for a distance $d$ before rotating as illustrated in Fig. 4, and, similarly, rotates back at distance $d$ before the target.

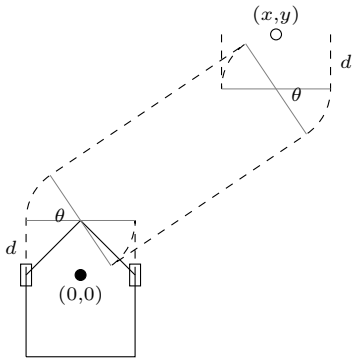


Fig. 4. Here we use rotations but with straight segments of distance $d$ at the beginning and end of the path so the rotations can occur without banging into horizontal boundaries abutting the robot's beginning and ending positions. The distance $d$ in this picture is exaggerated for visual effect.

**Exercise 3a** If the distance from the robot's reference point to the back of the robot (bottom in the picture) is 1cm and a boundary runs immediately behind the robot, how large must $d$ be to allow room for the rotation?

**Exercise 3b** Using the value of $d$ computed in Exercise 3a, compute the time to traverse the path in Fig. 4. Similarly to Exercise 2a, you will need to compute the new angle $\theta$.

A further consideration regarding the various paths we have analyzed is that we have been making a simplifying assumption that we can change the velocities of the wheels instantaneously as long as we impose an upper limit on velocity. In reality, the robot will neither stop nor start instantaneously, and it will probably be necessary in practice to insert brief delays within the path whenever a turn begins or ends to allow a settling of robot movement before changing the motor speeds.

**Exercise 4a** For each of the paths considered so far (Figures 2(a), 2(b), 3(a), 3(b), and 4), how many internal delays would we need based on the discussion in the paragraph above?

We could reduce the number of internal delays to just 1 by routing the reference point along a circular arc to $(x/2, y/2)$ and then mirroring that motion to reach $(x, y)$ as in Fig. 5.

**Exercise 4b** Compute the time (ignoring internal delays as before) to traverse the path in Fig. 5. Use the right triangle drawn for reference to compute the angle $\theta$ in this case and the radius $r$ of each of the two arcs traversed by the reference point.

### *E. General $x$ and $y$*

**Exercise 5** We have considered only one set of values for $w$, $x$, and $y$. For extra credit, return to the previous problems that involve computing the time for a robot motion, and find the result in terms of $w$, $x$, and $y$ instead of using specific numbers. (Warning: In the case of Figure 3(b), it is difficult to determine the angle $\theta$, and you should just express the result in terms of $\theta$ and determine a relationship that $\theta$ should obey.)

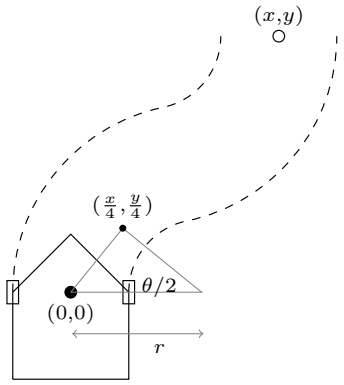


Fig. 5. A path in which the reference point traverses two identical circular arcs.

## IV. Further Work

A freestanding version of the worksheet developed in this paper will be available through the Loyola University Chicago eCommons (http://ecommons.luc.edu). In addition, source files used in conjunction with the LaTeX document preparation system [40] and with the TikZ interface to the PGF package [41] will be available, so that teachers can create modified versions starting from this base.

A preliminary version of this worksheet has already been distributed to some Chicago high school teachers of algebra and precalculus classes who expressed interest in bringing trigonometry applications into their classrooms. The worksheet will also be provided to additional teachers that are being identified through the Robotics in Education conference and the Global Conference on Educational Robotics that co-locate with the European and global finals of the Botball competition.

During the new school year, teachers will be reminded and encouraged to use this worksheet or a variation of it. Initially, we will be collecting feedback from teachers regarding what portions of the worksheet, and perhaps modified presentations, they find most suitable for their classes or robotics clubs. We will collect informal evaluations from teachers in math classes regarding how these exercise impact student interest in algebra and trigonometry, and from robotics coaches regarding how these exercises impact student thinking in connection with planning and completing robotics tasks. After that, we hope to do more formalized studies through student surveys.


## Acknowledgments

The authors are supported in part by National Science Foundation grants CNS-1738691, CNS-1543217, and CNS-1542971.

## REFERENCES


[1] A. Eguchi, "RoboCupJunior for promoting STEM education, 21st century skills, and technological advancement through robotics competition," *Robotics and Autonomous Systems*, vol. 75, Part B, pp. 692–699, 2016.

[2] B. Ericson and T. McKlin, "Effective and sustainable computing summer camps," in *SIGCSE '12*. Association for Computing Machinery, 2012, pp. 289–294.

[3] L. M. Grabowski and P. Brazier, "Robots, recruitment and retention: Broadening participation through CS0," in *Proceedings of 2011 Frontiers in Education Conference (FIE)*, 2011, pp. F4H1–5.

[4] S. H. Kim and J. W. Jeon, "Introduction for freshmen to embedded systems using LEGO Mindstorms," *IEEE Transactions on Education*, vol. 52, no. 1, pp. 99–108, 2009.

[5] R. B. Osborne, A. J. Thomas, and J. R. N. Forbes, "Teaching with robots: A service learning approach to mentor training," in *SIGCSE '10*. Association for Computing Machinery, 2010, pp. 172–176.

[6] B. S. Barker, G. Nugent, N. Grandgenett, and V. I. Adamchuk, Eds., *Robots in K–12 Education: A New Technology for Learning*. IGI Global, 2012.

[7] S. McGee, R. I. Greenberg, D. F. Reed, and J. Duck, "Evaluation of the IMPACTS computer science presentations," *The Journal for Computing Teachers*, pp. 26–40, Summer 2013, international Society for Technology in Education, http://www.iste.org/resources/product?id=2853.

[8] C. Hulleman and J. M. Harackiewicz, "Promoting interest and performance in high school science classes," *Science*, vol. 326, no. 5958, pp. 1410–1412, 2009.

[9] H. Gaspard, A.-L. Dicke, B. Flunger, B. M. Brisson, I. Häfner, B. Nagengast, and U. ch Trautwein, "Fostering adolescents' value beliefs for mathematics with a relevance intervention in the classroom," *Developmental Psychology*, vol. 51, no. 1226–1240, 2015.

[10] J. Eccles, "Who am I and what am I going to do with my life? Personal and collective identities as motivators of action," *Educational Psychologist*, vol. 44, no. 2, pp. 78–89, 2009.

[11] J. S. Eccles, "Studying gender and ethnic differences in participation in math, physical science, and information technology," *New Directions for Child and Adolescent Development*, no. 110, pp. 7–14, Winter 2005.

[12] M. Ben-Ari and F. Mondada, *Elements of Robotics*. Springer-Verlag, 2018.

[13] H. Chitsaz, S. M. La Valle, D. J. Balkcom, and M. T. Mason, "Minimum wheel-rotation paths for differential-drive mobile robots," *The International Journal of Robotics Research*, vol. 28, no. 1, pp. 66–80, 2009, https://doi.org/10.1177/0278364908096750.

[14] D. J. Balkcom and M. T. Mason, "Time optimal trajectories for bounded velocity differential drive vehicles," *International Journal of Robotics Research*, vol. 21, no. 3, pp. 199–217, 2002.

[15] J. P. Laumond, S. Sekhavat, and F. Lamiraux, "Guidelines in nonholomic motion planning for mobile robots," in *Robot Motion Planning and Control*, ser. Lecture Notes in Control and Information Sciences 229, J.-P. Laumond, Ed. Springer-Verlag, 1998, ch. 1, pp. 1–53.

[16] KISS Institute for Practical Robotics, "Botball," http://www.botball.org, 2019, accessed Jan. 22, 2019.

[17] J. Goode and G. Chapman, "Exploring Computer Science (version 8)," Retrieved October 1, 2018 from http://www.exploringcs.org/curriculum, 2018.

[18] Exploring Computer Science, "Where is ECS?" http://www.exploringcs.org/for-teachers-districts/ecs-now, 2018, accessed April 4, 2018.

[19] Chicago Public Library, "Finch robots land at CPL," https://www.chipublib.org/news/finch-robots-land-at-cpl, May 2014, accessed Jan. 22, 2019.

[20] R. I. Greenberg, G. K. Thiruvathukal, and S. T. Greenberg, "Integrating mathematics and educational robotics: Simple motion planning," in *Proceedings of the 10th International Conference on Robotics in Education, RiE 2019*, ser. Advances in Intelligent Systems and Computing. To be published by Springer-Verlag.

[21] G. K. Thiruvathukal, R. I. Greenberg, and D. Garcia, "Understanding turning radius and driving in convex polygon paths in introductory robotics," http://ecommons.luc.edu/cs_facpubs/202, 2018.

[22] D. Rider, "Easy projects to get started with scratch: Drawing 2D shapes," Edutopia blog of the George Lucas Educational Foundation: https://www.edutopia.org/blog/scratch-programming-drawing-2d-shapes-dylan-ryder, Aug. 2014, accessed Jan. 21, 2019.

[23] W. Johal, S. Andersen, M. Chevallier, A. Ozgur, F. Mondada, and P. Dillenbourg, "Learning symmetry with tangible robots," in *the 10th International Conference on Robotics in Education, RiE 2019*, ser. Advances in Intelligent Systems and Computing. To be published by Springer-Verlag.

[24] J. Leoste and M. Heidmets, "Bringing an educational robot into a basic education math lesson," in *the 10th International Conference on Robotics in Education, RiE 2019*, ser. Advances in Intelligent Systems and Computing. To be published by Springer-Verlag.

[25] G. Nugent, B. Barker, N. Grandgenett, and V. I. Adamchuk, "Impact of robotics and geospatial technology interventions on youth STEM learning and attitudes," *Journal of Research on Technology in Education*, vol. 42, no. 4, pp. 391–408, 2010.

[26] B. S. Barker and J. Ansorge, "Robotics as means to increase achievement scores in an informal learning environment," *Journal of Research on Technology in Education*, vol. 39, pp. 229–243, 2007.

[27] S. Hussain, J. Lindh, and G. Shukur, "The effect of LEGO training on pupils' school performance in mathematics, problem solving ability and attitude: Swedish data," *Educational Technology & Society*, vol. 9, no. 3, pp. 182–194, 2006.

[28] V. Estivill-Castro, "Inviting teachers to use educational robotics to foster mathematical problem-solving," in *the 10th International Conference on Robotics in Education, RiE 2019*, ser. Advances in Intelligent Systems and Computing. To be published by Springer-Verlag.

[29] C. C. Chung, C. Cartwright, and M. Cole, "Assessing the impact of an autonomous robotics competition for stem education," *Journal of STEM Education*, vol. 15, no. 2, pp. 24–34, 2014.

[30] D. B. Reister and F. G. Pin, "Time-optimal trajectories for mobile robots with two independently driven wheels," *The International Journal of Robotics Research*, vol. 13, no. 1, pp. 38–54, Feb. 1994.

[31] M. Renaud and J.-Y. Fourquet, "Minimum time motion of a mobile robot with two independent, acceleration-driven wheels," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, 1997, pp. 2608–2613.

[32] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, Jul. 1957, https://www.jstor.org/stable/2372560.

[33] J. A. Reeds, III and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[34] H. J. Sussmann and G. Tang, "Shortest paths for the Reeds-Shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control," Department of Mathematics, Rutgers University, Report SYCON 91-10, sep 1991.

[35] P. Souères and J.-P. Laumond, "Shortest paths synthesis for a car-like robot," *IEEE Transactions on Automatic Control*, vol. 41, no. 5, pp. 672–688, May 1996.

[36] H. Wang, Y. Chen, and P. Souères, "An efficient geometric algorithm to compute time-optimal trajectories for a car-like robot," in *Proceedings of the 46th IEEE Conference on Decision and Control*, Dec. 2007, pp. 5383–5388.

[37] ——, "A geometric algorithm to compute time-optimal trajectories for a bidirectional steered robot," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 399–413, Apr. 2009.

[38] Common Core State Standards Initiative, "Common core state standards for mathematics," http://www.corestandards.org/wp-content/uploads/Math_Standards1.pdf, 2010.

[39] R. I. Greenberg, "Pythagorean approximations for LEGO: Merging educational robot construction with programming and data analysis," in *Proceedings of the 8th International Conference on Robotics in Education, RiE 2017*, ser. Advances in Intelligent Systems and Computing, vol. 630. Springer-Verlag, 2018, pp. 65–76.

[40] The LaTeX Project, "Latex - A document preparation system," http://www.latex-project.org accessed 6/19/19.

[41] T. Tantau, "CTAN: package pgf," https://ctan.org/pkg/pgf?lang=en accessed 6/19/19.